Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router, Release 6.1.x

First Published: 2016-11-01

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

PREFACE

CHAPTER 1

CHAPTER 2

CHAPTER 3

Preface

The *Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router* provides information and procedures related to router interface and hardware configuration.

The preface contains these sections:

Changes to This Document

This table lists the technical changes made to this document since it was first released.

Table 1: Changes to This Document

Date	Summary
May 2017	Added ERSPAN and LACP Fallback features.
February 2017	Added IP-in-IP De-capsulation feature for Release 6.1.3
November 2016	Initial release of this document.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

CHAPTER 1

New and Changed Interface and Hardware Component Features

This chapter lists all the features that have been added or modified in this guide. The table also contains references to these feature documentation sections.

-

New and Changed Information

Table 2: New and Changed Features

Feature	Description	Changed in Release	Where Documented
No new features in this release	NA	Release 6.1.2	NA

CHAPTER 2

Preconfiguring Physical Interfaces

This module describes the preconfiguration of physical interfaces on the .

Preconfiguration is supported for the following types of interfaces and controllers:

- ATM
- Gigabit Ethernet
- 10-Gigabit Ethernet
- Management Ethernet
- Packet-over-SONET/SDH (POS)
- Spatial Reuse Protocol (SRP)
- Serial
- SONET controllers and channelized SONET controllers

Preconfiguration allows you to configure modular services cards before they are inserted into the router. When the cards are inserted, they are instantly configured.

The preconfiguration information is created in a different system database tree (known as the *preconfiguration directory* on the route processor [RP]), rather than with the regularly configured interfaces.

There may be some preconfiguration data that cannot be verified unless the modular services card is present, because the verifiers themselves run only on the modular services card. Such preconfiguration data is verified when the modular services card is inserted and the verifiers are initiated. A configuration is rejected if errors are found when the configuration is copied from the preconfiguration area to the active area.

Note Only physical interfaces can be preconfigured.

Feature History for Preconfiguring Physical Interfaces

Release	Modification
Release 2.0	POS preconfiguration was introduced.
Release 3.0	Ethernet preconfiguration was introduced.

Release 3.3.0	The following interface preconfiguration support was introduced: • Management Ethernet interface • SRP interface
Release 3.7.0	ATM interface preconfiguration was introduced.

Preconfiguring Physical Interfaces

This module describes the preconfiguration of physical interfaces on the .

Preconfiguration is supported for the following types of interfaces and controllers:

- ATM
- Gigabit Ethernet
- 10-Gigabit Ethernet
- Management Ethernet
- Packet-over-SONET/SDH (POS)
- Spatial Reuse Protocol (SRP)
- Serial
- SONET controllers and channelized SONET controllers

Preconfiguration allows you to configure modular services cards before they are inserted into the router. When the cards are inserted, they are instantly configured.

The preconfiguration information is created in a different system database tree (known as the *preconfiguration directory* on the route processor [RP]), rather than with the regularly configured interfaces.

There may be some preconfiguration data that cannot be verified unless the modular services card is present, because the verifiers themselves run only on the modular services card. Such preconfiguration data is verified when the modular services card is inserted and the verifiers are initiated. A configuration is rejected if errors are found when the configuration is copied from the preconfiguration area to the active area.

Note Only physical interfaces can be preconfigured.

Feature History for Preconfiguring Physical Interfaces

Release	Modification
Release 2.0	POS preconfiguration was introduced.
Release 3.0	Ethernet preconfiguration was introduced.
Release 3.3.0	The following interface preconfiguration support was introduced: • Management Ethernet interface • SRP interface
Release 3.7.0	ATM interface preconfiguration was introduced.

Prerequisites for Preconfiguring Physical Interfaces

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 preconfiguring physical interfaces, be sure that the following condition is met:

- Preconfiguration drivers and files are installed. Although it may be possible to preconfigure physical interfaces without a preconfiguration driver installed, the preconfiguration files are required to set the interface definition file on the router that supplies the strings for valid interface names.

Information About Preconfiguring Physical Interfaces

To preconfigure interfaces, you must understand the following concepts:

Physical Interface Preconfiguration Overview

Preconfiguration is the process of configuring interfaces before they are present in the system. Preconfigured interfaces are not verified or applied until the actual interface with the matching location (rack/slot/module) is inserted into the router. When the anticipated modular services card is inserted and the interfaces are created, the precreated configuration information is verified and, if successful, immediately applied to the router's running configuration.

Note When you plug the anticipated modular services card in, make sure to verify any preconfiguration with the appropriate **show** commands.

Use the **show run** command to see interfaces that are in the preconfigured state.

Note We recommend filling out preconfiguration information in your site planning guide, so that you can compare that anticipated configuration with the actual preconfigured interfaces when that card is installed and the interfaces are up.

Tip Tip Use the **commit best-effort** command to save the preconfiguration to the running configuration file. The **commit best-effort** command merges the target configuration with the running configuration and commits only valid configuration (best effort). Some configuration might fail due to semantic errors, but the valid configuration still comes up.

Benefits of Interface Preconfiguration

Preconfigurations reduce downtime when you add new cards to the system. With preconfiguration, the new modular services card can be instantly configured and actively running during modular services card bootup.

Another advantage of performing a preconfiguration is that during a card replacement, when the modular services card is removed, you can still see the previous configuration and make modifications.

Use of the Interface Preconfigure Command

Interfaces that are not yet present in the system can be preconfigured with the **interface preconfigure** command in global configuration mode.

The **interface preconfigure** command places the router in interface configuration mode. Users should be able to add any possible interface commands. The verifiers registered for the preconfigured interfaces verify the configuration. The preconfiguration is complete when the user enters the **end** command, or any matching exit or global configuration mode command.

Note It is possible that some configurations cannot be verified until the modular services card is inserted.

Do not enter the **no shutdown** command for new preconfigured interfaces, because the no form of this command removes the existing configuration, and there is no existing configuration.

Users are expected to provide names during preconfiguration that will match the name of the interface that will be created. If the interface names do not match, the preconfiguration cannot be applied when the interface is created. The interface names must begin with the interface type that is supported by the router and for which drivers have been installed. However, the slot, port, subinterface number, and channel interface number information cannot be validated.

Note Specifying an interface name that already exists and is configured (or an abbreviated name like e0/3/0/0) is not permitted.

Active and Standby RPs and Virtual Interface Configuration

The standby RP is available and in a state in which it can take over the work from the active RP should that prove necessary. Conditions that necessitate the standby RP to become the active RP and assume the active RP's duties include:

- Failure detection by a watchdog
- Standby RP is administratively commanded to take over
- Removal of the active RP from the chassis

If a second RP is not present in the chassis while the first is in operation, a second RP may be inserted and will automatically become the standby RP. The standby RP may also be removed from the chassis with no effect on the system other than loss of RP redundancy.

After switchover, the virtual interfaces will all be present on the standby (now active) RP. Their state and configuration will be unchanged, and there will have been no loss of forwarding (in the case of tunnels) over the interfaces during the switchover. The Cisco CRS-1 Router uses nonstop forwarding (NSF) over tunnels through the switchover of the host RP.

Note The user does not need to configure anything to guarantee that the standby interface configurations are maintained.

How to Preconfigure Physical Interfaces

This task describes only the most basic preconfiguration of an interface.

SUMMARY STEPS

1. **configure**
2. **interface preconfigure** *type interface-path-id*
3. Use one of the following commands:
   - **ipv4 address** *ip-address subnet-mask*
   - **ipv4 address***ip-address / prefix*
4. Configure additional interface parameters, as described in this manual in the configuration chapter that applies to the type of interface that you are configuring.
5. **end** or **commit** best-effort
6. **show running-config**

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**interface preconfigure** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface preconfigure GigabitEthernet 0/1/0/0`	Enters interface preconfiguration mode for an interface, where *type* specifies the supported interface type that you want to configure and *interface-path-id* specifies the location where the interface will be located in *rack*/*slot*/*module*/*port* notation.
Step 3	Use one of the following commands: • **ipv4 address** *ip-address subnet-mask* • **ipv4 address***ip-address / prefix* **Example:** `RP/0/RP0/CPU0:router(config-if-pre)# ipv4 address 192.168.1.2/32`	Assigns an IP address and mask to the interface.
Step 4	Configure additional interface parameters, as described in this manual in the configuration chapter that applies to the type of interface that you are configuring.	

Step 5	**end** or **commit** best-effort
	Example:
	`RP/0/RP0/CPU0:router(config-if-pre)# end`
	or
	`RP/0/RP0/CPU0:router(config-if-pre)# 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)?`
		• 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 best-effort** command to save the configuration changes to the running configuration file and remain within the configuration session. The **commit best-effort** command merges the target configuration with the running configuration and commits only valid changes (best effort). Some configuration changes might fail due to semantic errors.

Step 6	**show running-config** **Example:**	(Optional) Displays the configuration information currently running on the router.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show running-config`	

Configuration Examples for Preconfiguring Physical Interfaces

This section contains the following example:

Preconfiguring an Interface: Example

The following example shows how to preconfigure a basic Ethernet interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface preconfigure GigabitEthernet 0/1/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.2/32
RP/0/RP0/CPU0:router(config-if)# commit
```

CHAPTER 3

Configuring ATM Interfaces

This module describes how to configure ATM.

ATM is a cell-switching and multiplexing technology that is widely used in Wide Area Networks (WANs). ATM protocol standards enable point-to-point, point-to-multipoint, and broadcast services connections using various slow- and high-speed network media. Connectivity between two ATM permanent virtual circuits (PVCs) is established using ATM signaling mechanisms. Various ATM signaling standards are defined by these ATM forum standards:

- UNI Version 3.0, Version 3.1, and Version 4.0
- ITU
- IETF

Feature History for Configuring ATM Interfaces

Release	Modification
Release 3.7.0	ATM Layer 2 VPN (Port Mode) and QoS was introduced on the following SPAs: • 3-Port Clear Channel OC-3 ATM SPA • 1-Port Clear Channel OC-12 ATM SPA
Release 3.8.0	Support for ATM over MPLS was added.

Release 3.9.2	Support for ATM UNI (Layer 3 VPN) was added for the following SPAs: • 1-Port Clear Channel OC-3 ATM SPA • 3-Port Clear Channel OC-3 ATM SPA • 1-Port Clear Channel OC-12 ATM SPA The following ATM UNI features are supported in this release: • ATM UNI L3 VC termination (UNI 3.0/3.1) • ILMI • Per VC QoS • ATM COS • MPLS L3VPN per VC sub-interface • Support for both L2VPN and L3VPN under the same physical interface • ATM F4/F5 OAM • MR-APS • IGP routing with VRF/VPN support: OSPF, BGP, EIGRP, RIP and static • 150 VC/VP connections per port, maximum VC/VP numbering up to 1024

Configuring ATM Interfaces

This module describes how to configure ATM.

ATM is a cell-switching and multiplexing technology that is widely used in Wide Area Networks (WANs). ATM protocol standards enable point-to-point, point-to-multipoint, and broadcast services connections using various slow- and high-speed network media. Connectivity between two ATM permanent virtual circuits (PVCs) is established using ATM signaling mechanisms. Various ATM signaling standards are defined by these ATM forum standards:

- UNI Version 3.0, Version 3.1, and Version 4.0
- ITU
- IETF

Feature History for Configuring ATM Interfaces

Release	Modification
Release 3.7.0	ATM Layer 2 VPN (Port Mode) and QoS was introduced on the following SPAs: • 3-Port Clear Channel OC-3 ATM SPA • 1-Port Clear Channel OC-12 ATM SPA
Release 3.8.0	Support for ATM over MPLS was added.
Release 3.9.2	Support for ATM UNI (Layer 3 VPN) was added for the following SPAs: • 1-Port Clear Channel OC-3 ATM SPA • 3-Port Clear Channel OC-3 ATM SPA • 1-Port Clear Channel OC-12 ATM SPA The following ATM UNI features are supported in this release: • ATM UNI L3 VC termination (UNI 3.0/3.1) • ILMI • Per VC QoS • ATM COS • MPLS L3VPN per VC sub-interface • Support for both L2VPN and L3VPN under the same physical interface • ATM F4/F5 OAM • MR-APS • IGP routing with VRF/VPN support: OSPF, BGP, EIGRP, RIP and static • 150 VC/VP connections per port, maximum VC/VP numbering up to 1024

Prerequisites for Implementing ATM

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.

Information About ATM

This section provides overviews of these features:

Network nodes use ATM connections to transfer bits of data organized as 53-byte ATM cells. User information (such as voice, video, and data) is segmented into ATM cells on one end of the connection, and then reassembled on the other end of the connection. ATM Adaptation Layer (AAL) defines the conversion of user information into ATM cells. AAL1 and AAL2 handle isochronous traffic (such as voice and video), and are relevant to the ATM node only when it is equipped with either a CES (Circuit Emulation Service) ATM interface card, or when it has voice over AAL2 capabilities. AAL3/4 and AAL5 support data communications; that is, they segment and reassemble data packets.

The two types of devices in an ATM network are switches and routers. Typically, ATM switches do packet switching at Layer 2, while ATM routers do packet switching using Layer 3 addresses, such as IPv4 network addresses, IPv6 network addresses, and MPLS labels.

ATM is supported on the following line cards:

- Cisco 1-port Clear Channel OC-3 SPA
- Cisco 3-port Clear Channel OC-3 SPA
- Cisco 1-port Clear Channel OC-12 SPA

Cisco IOS XR software ATM interfaces can operate in these modes:

- Point-to-point
- Layer 2 port mode

Note A single ATM interface can simultaneously support point-to-point and L2VPN subinterfaces.

In Cisco IOS XR software, ATM interface configuration is hierarchical and comprises the following elements:

1. The ATM main interface, which is the physical interface. ATM main interfaces can be configured with point-to-point subinterfaces, vp-tunnels, ILMI interfaces, or as Layer 2 port mode attachment circuits (ACs) or Layer 2 subinterface ACs.
2. ATM subinterfaces, which are configured under the ATM main interface. An ATM subinterface does not actively carry traffic until you configure a PVC or PVP under the ATM subinterface.
3. PVCs, which are configured under an ATM subinterface. A single PVC is allowed per subinterface. PVCs are supported under point-to-point and Layer 2 subinterfaces.
4. Permanent virtual paths (PVPs), which are configured under a Layer 2 ATM subinterface. A single PVP is allowed per subinterface.

VC-Class Mapping

A virtual circuit (VC) class enables the configuration of VC parameters that are then mapped to a main interface, subinterface, or PVC. Without vc-classes, you must perform considerable manual configuration on each ATM main interface, subinterface, and PVC and on the router. This configuration can be time consuming and error prone. After you have created vc-class, you can apply that vc-class to as many ATM interfaces, subinterfaces, or PVCs as you want.

Vc-classes include the following types of configuration data:

- ATM encapsulation for the VC
- OAM management
- traffic shaping

The order of configuration precedence is hierarchical, as demonstrated in the following list, where configuration on the PVC takes the highest precedence, and configuration on a vc-class that is attached to the ATM main interface takes the lowest precedence:

1. Configuration on the PVC
2. Configuration on a vc-class that is attached to the PVC
3. Configuration on the subinterface
4. Configuration on a vc-class that is attached to the subinterface
5. Configuration on the ATM main interface
6. Configuration on a vc-class that is attached to the ATM main interface

For example, if the a PVC has unspecified bit rate (UBR) traffic shaping configured, but it is attached to a class map that is configure with CBR traffic shaping, the PVC maintains the UBR traffic shaping.

Note Vc-classes are not applicable to Layer 2 port mode ACs and Layer 2 PVPs. For Layer 2 VPN configurations, Vc-classes are applicable to the PVC only.

F5 OAM on ATM Interfaces

The F5 Operation, Administration, and Maintenance (OAM) feature performs fault-management and performance-management functions on PVCs. If the F5 OAM feature is not enabled on a PVC, then that PVC remains up on the end device in the event of a service disruption where network connectivity is lost. The result is that routing entries that point to the connection remain in the routing table and, therefore, packets are lost. The F5 OAM feature detects such failures and brings the PVC down if there is a disruption along its path.

Use the **oam-pvc manage** command to enable the F5OAM feature on a PVC. After OAM is enabled on a PVC, the PVC can generate F5 loopback cells and you can configure continuity check (CC) management for the PVC. Use the **oam ais-rdi** and **oam retry** commands to configure continuity checking on a PVC.

To drop all current and future OAM cells received on an ATM interface, use the **atm oam flush** command in interface configuration mode.

Note The **oam ais-rdi** and **oam retry** commands take effect only after OAM management is enabled on a PVC with the **oam-pvc manage** command.

ILMI on ATM Interfaces

The ILMI protocol is defined by the ATM Forum for setting and capturing physical layer, ATM layer, virtual path, and virtual circuit parameters on ATM interfaces. When two ATM interfaces run the ILMI protocol, they exchange ILMI packets across the physical connection. These packets consist of SNMP messages as large as 484 octets. ATM interfaces encapsulate these messages in an ATM adaptation layer 5 (AAL5) trailer, segment the packet into cells, and schedule the cells for transmission.

You must enable ILMI on ATM interfaces that communicate with end devices that are configured for ILMI. To enable ILMI, create a PVC with ILMI encapsulation directly under the main ATM interface by using the **pvc** *vpi*/*vci* **ilmi** command in interface configuration mode.

PVCs use ILMI encapsulation to carry ILMI messages. Use the **pvc** *vpi*/*vci* **ilmi** command in interface configuration mode to create an ILMI PVC on an ATM main interface.

Note You must use the same VPI and VCI values on both ends of the PVC that connects the end device and the router. The ILMI configuration commands are available only after an ILMI PVC is created under the ATM main interface. The ILMI configuration takes effect on the ATM main interface. ILMI configuration is not supported on Layer 2 port mode ACs.

Layer 2 VPN on ATM Interfaces

The Layer 2 VPN (L2VPN) feature enables the connection between different types of Layer 2 attachment circuits and pseudowires, allowing users to implement different types of end-to-end services.

Cisco IOS XR software supports a point-to-point, end-to-end service, where two ATM ACs are connected together.

Switching can take place in two ways:

- AC-to-PW—Traffic reaching the PE is tunneled over a pseudowire (and conversely, traffic arriving over the PW is sent out over the AC). This is the most common scenario.
- Local switching—Traffic arriving on one AC is immediately sent out another AC without passing through a pseudowire.

Keep the following in mind when configuring L2VPN on an ATM interface:

- Cisco IOS XR software supports up to 2000 ACs per line card.
- ATM-over-MPLS supports two types of cell encapsulation:
  - AAL5 CPCS mode—Unsegmented ATM cells are transported across an MPLS backbone.
  - ATM cell (AAL0) mode—Cells are segmented and then reassembled, or packed. AAL0 is supported on ATM main ports, PVCs, and PVPs. The benefits of using AAL0 mode is that groups of ATM cells share a label that maximizes bandwidth efficiencies.

Note AAL5 mode is supported on PVCs only.

Use the following commands to display AC and pseudowire information:

- **show interfaces**
- **show l2vpn xconnect**
- **show atm pvp**
- **show atm pvc**

Note For detailed information about configuring an L2VPN network, see the *Implementing MPLS Layer 2 VPNs* module of *Cisco IOS XR Multiprotocol Label Switching Configuration Guide.*

Cell Packing on L2VPN ACs with AAL0 Mode Encapsulation

Cell packing is supported on L2VPN ATM interfaces that are configured with AAL0 mode encapsulation. Cell packing relates to the delay variations that are defined in the ATM standards. Users can specify the number of cells that can be processed by the pseudowire, and configure the maximum cell packing timeout (MCPT) timers to use in conjunction with cell packing.

The **cell-packing** command allows the user to perform the following tasks:

- Configure the maximum number of cells that can be transmitted in a single packet.
- Attach one of the three MCPT timers to an individual Layer 2 port mode AC, PVC, or PVP.

The three MCPT timers are defined under the main ATM interface with the **atm mcpt-timer** command, which lets the user specify the maximum number of microseconds to wait to complete cell packing on a single packet before that packet is transmitted. If the associated MCPT timer expires before the maximum number of cells that can be packed is reached, then the packet is transmitted with the number of cells that have been packed thus far.

We recommend configuring a low, medium, and high value for the three MCPT timers to accommodate the different ATM traffic classes. Low- latency constant bit rate (CBR) traffic typically uses a low MCPT timer value, while high-latency Unspecified bit rate (UBR) traffic typically requires a high MCPT timer value. Variable bit rate real-time (VBR-rt) and variable bit rate non-real-time (VBR-nrt) traffic typically use a median MCPT timer value.

ATM Layer 2 QoS

QoS is configured on ATM interfaces primarily in the same way that it is configured on other interfaces. No new CLIs are added in this release.

For complete information on configuring QoS and QoS commands, refer to these documents:

- *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco CRS Router*
- *Cisco IOS XR Modular Quality of Service Command Reference for the Cisco CRS Router*

This section describes the features and restrictions that apply to ATM Layer 2 QoS.

Features

These QoS features are supported:

- Layer 2 Ingress QoS – classification on ATM CLP, marking with MPLS EXP imposition are supported.
- Layer 2 Egress Main Interface QoS – shaping, policing, and queueing are supported. Marking is not supported. This feature works on both Layer 2 and Layer 3 PVCs independent of any subinterface QoS policies.
- The Modular QoS CLI (MQC) actions are supported for ATM traffic in the ingress direction only.
  - match atm clp
  - match atm oam
  - set atm clp
  - set mpls exp imp
  - set prec tunnel (L2TPv3 only)
  - set dscp tunnel (L2TPv3 only)
- Traffic is classified based on Cell Loss Priority–CLP1, CLP0, or OAM.
- OAM traffic can be excluded from policing by using the match-oam classification in a hierarchical policy map
- The following set actions are supported:
  - set mpls exp imp
  - set prec tunnel
  - set dscp tunnel
  - set qos-group
  - set disc-class
  - set atm-clp (exceed action only)
- Policy map counters are supported.

Matching

The following match criteria is supported on Layer 2 ATM interfaces in the ingress direction only:

- match atm clp0
- match atm clp1
- match atm oam

The following match criteria is supported on Layer 2 ATM interfaces in the egress direction only:

- match mpls exp topmost (egress only)
- match qos-group (egress only)

Note The **match-all** command does not support the above match criteria.

Marking

The following marking actions are supported on Layer 2 ATM interfaces:

- set mpls exp imposition (AToM only)
- set qos-group (AToM and local switching)
- set discard-class (AToM and local switching)
- set mpls exp imposition and set atm-clp (AToM only)
- set tunnel-dscp (L2TPv3 only)
- set tunnel-prec (L2TPv3 only)

Note Packets can be matched and remarked for CLP0, CLP1, and OAM.

Policing

Policing is supported on Layer 2 ATM interfaces in the ingress direction only.

Policing is performed during segmentation and reassembly (SAR) for the following ATM traffic classes:

- CBR.1
- VBR.1
- VBR.2
- VBR.3
- UBR.1
- UBR.2

Policing is supported for VC and VP modes, but not for Port mode L2 ATM interfaces.

OAM cells are policed along with the user cells unless the QOS policy is explicitly configured to exclude OAM cells from being policed. This can be achieved using different match criteria in the policy map with class-default matching all the traffic including OAM cells.

Policing is supported for ATM AAL5SNAP, AAL5MUX and AAL5NLPID encapsulated packets.

Policing is done on AAL0 packets with the same conditions as AAL5 packets as follows:

- AAL5 packet is conforming if all the cells in the packet conform to PCR and SCR buckets.
- AAL5 packet is exceeding if at least one cell does not conform to the SCR bucket.
- AAL5 packet is violating if at least one cell does not conform to the PCR bucket.

Note The Martini Control Word C bit is set for all exceeding AAL5 packets. All violating AAL5 packets are dropped.

The following policing options are supported for ATM TM4.0 GCRA policing:

- Rate in cellsps and percent
- Peak rate in cellsps and percent
- Delay tolerance in us
- Maximum burst size in cells

The following conform and exceed actions are supported for Layer 2 ATM interfaces in the ingress direction:

- transmit
- drop
- set mpls exp imposition (AToM only)
- set qos-group (AToM and Local switching)
- set discard-class (AToM and Local switching)
- set atm-clp (exceed action only, AToM and Local switching)
- set tunnel-prec (L2TPv3 only)
- set tunnel-dscp (L2TPv3 only)

Note The only violate action that is supported is the drop action.

The following combination of multiple policing actions is supported:

- set mpls exp imposition and set atm-clp (exceed action only, AToM only)

Hierarchical Policy Maps

For VBR.2 and VBR.3 traffic classes, 2-level hierarchical policy maps are supported in the ingress direction only. Attempts to attach hierarchical policy maps in the egress direction are denied.

The parent policy contains the policing configuration for the PCR bucket and matches on all traffic. The parent policy may exclude OAM traffic.

The child policy contains the policing configuration for the SCR bucket and typically matches on CLP0 cells.

Marking actions are supported only in child policy maps. All other policing actions are allowed in parent policy maps.

Only two policing buckets per Layer 2 circuit are allowed; one in the parent policy that defines the peak rate, and one in the child policy that defines the SCR.

Typically CLP0 cells are sent to the SCR bucket, but it is possible to send both CLP0 and CLP1 cells to the SCR bucket, using the classification criteria in the child policy.

Note For ATM Layer 2 QoS, in policy maps, the **set atm-clp** command is supported only as a police exceed action. It is not supported as a standalone set action.

Configuring ATM Interfaces

The ATM interface configuration tasks are described in these procedures:

Bringing Up an ATM Interface

This task describes the commands used to bring up an ATM interface.

Before you begin

Restrictions

The configuration on both ends of the ATM connection must match for the interface to be active.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id*
3. **no shutdown**
4. **end** or **commit**
5. **exit**
6. **exit**
7. **show interfaces atm** *interface-path-id* **brief**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1`	Enters ATM interface configuration mode.
Step 3	**no shutdown**	Removes the shutdown configuration.

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router (config-if)# no shutdown`	**Note** • Removal of the shutdown configuration eliminates the forced administrative down on the interface, enabling it to move to an up or down state.

Step 4	**end** or **commit**
	
	Example:
	
	`RP/0/RP0/CPU0:router (config-if)# end`
	
	or
	
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**exit** **Example:** `RP/0/RP0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode. Repeat Step 1 through Step 6 to bring up the interface at the other end of the connection. Brings up the connection. **Note** • The configuration on both ends of the ATM connection must match.
Step 7	**show interfaces atm** *interface-path-id* **brief** **Example:** `RP/0/RP0/CPU0:router# show interfaces atm 0/6/0/1 brief`	(Optional) Verifies that the interface is active and properly configured. If you have brought up an ATM interface properly, the “Intf State” field for that interface in the **show interfaces atm** command output shows “up.”

What to do next

To modify the default configuration of the ATM interface you just brought up, see the "Configuring Optional ATM Interface Parameters" section on page 26.

To configure a point-to-point subinterface on the ATM interface you just brought up, see the "How to Create and Configure a Point-to-Point ATM Subinterface with a PVC" section on page 28.

To create a vp-tunnel on the ATM interface you just brought up, see the "How to Create and Configure a VP-Tunnel" section on page 33.

To use the interface as a Layer 2 post mode AC, see the "How to Configure a Layer 2 Attachment Circuit" section on page 40.

To attach a Vc-class to the ATM interface you just brought up, see the "How to Create and Configure a VC-Class" section on page 52.

To enable ILMI on the ATM interface you just brought up, see the "How to Configure ILMI on ATM Interfaces" section on page 59.

Configuring Optional ATM Interface Parameters

This task describes the commands you can use to modify the default configuration on an ATM interface.

Before you begin

Before you modify the default ATM interface configuration, we recommend that you bring up the ATM interface and remove the shutdown configuration, as described in the Bringing Up an ATM Interface.

Restrictions

The configuration on both ends of the ATM connection must match for the interface to be active.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id*
3. **atm maxvpi-bits 12**
4. **atm oam flush**
5. **atm mcpt-timers** *timer-1 timer-2 timer-3*
6. **end** or **commit**
7. **exit**
8. **exit**
9. **show atm interface atm** [*interface-path-id*]
10. **show interfaces atm** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1`	Enters ATM interface configuration mode.
Step 3	**atm maxvpi-bits 12** **Example:** `RP/0/RP0/CPU0:router (config-if)# atm maxvpi-bits 12`	(Optional) Enables support for the 12-bit VPI NNI cell format.
Step 4	**atm oam flush** **Example:** `RP/0/RP0/CPU0:router (config-if)# atm oam flush`	(Optional) Drops all current and future OAM cells received on an ATM interface.
Step 5	**atm mcpt-timers** *timer-1 timer-2 timer-3* **Example:** `RP/0/RP0/CPU0:router (config-if)# atm mcpt-timers 50 100 200`	(Optional) Specifies the maximum cell packing timeout values for each of the three per-interface MCPT timers, in microseconds. **Note** • The default value for each timer is 50 microseconds. • The **atm mcpt-timers** command is applicable to Layer 2 ATM ACs only.

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

	Command or Action	Purpose
Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode.
Step 9	**show atm interface atm** [*interface-path-id*] **Example:** `RP/0/RP0/CPU0:router# show atm interface atm 0/6/0/1`	(Optional) Displays ATM-specific data for the specified ATM interface.
Step 10	**show interfaces atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces atm 0/6/0/1`	(Optional) Displays general information for the specified ATM interface.

What to do next

To configure a point-to-point subinterface on the ATM interface you just brought up, see the "How to Create and Configure a Point-to-Point ATM Subinterface with a PVC" section on page 28.

To create a vp-tunnel on the ATM interface you just brought up, see the "How to Create and Configure a VP-Tunnel" section on page 33.

To use the interface as a Layer 2 ATM AC, see the "How to Configure a Layer 2 Attachment Circuit" section on page 40.

To attach a Vc-class to the ATM interface you just brought up, see the "How to Create and Configure a VC-Class" section on page 52.

To enable ILMI on the ATM interface you just brought up, see the "How to Configure ILMI on ATM Interfaces" section on page 59.

How to Create and Configure a Point-to-Point ATM Subinterface with a PVC

The configuration tasks for creating and configuring a point-to-point ATM subinterface with a PVC are described in the following procedures:

Creating a Point-to-Point ATM Subinterface with a PVC

The procedure in this section creates a point-to-point ATM subinterface and configures a permanent virtual circuit (PVC) on that ATM subinterface.

Before you begin

Before you can create an ATM subinterface on an ATM interface, you must bring up an ATM interface, as described in the Bringing Up an ATM Interface.

Restrictions

Only one PVC can be configured for each point-to-point ATM subinterface.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **point-to-point**
3. **ipv4 address** *ipv4_address/prefix*
4. **pvc** *vpi/vci*
5. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id.subinterface* **point-to-point** **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1.10`	Enters ATM subinterface configuration mode.
Step 3	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0/RP0/CPU0:router (config-subif)#`**`ipv4 address 10.46.8.6/24`**	Assigns an IP address and subnet mask to the subinterface.
Step 4	**pvc** *vpi/vci* **Example:** `RP/0/RP0/CPU0:router (config-subif)# pvc 5/10`	(Optional) Creates an ATM permanent virtual circuit (PVC) and enters ATM PVC configuration submode. **Note** • Only one PVC is allowed per subinterface.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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]:`

	Command or Action	Purpose
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. Repeat Step 1 through Step 5 to bring up the ATM subinterface and any associated PVC at the other end of the connection. Brings up the ATM connection. **Note** • The configuration on both ends of the subinterface connection must match.

What to do next

To configure optional PVC parameters, see the "Configuring Optional Point-to-Point ATM PVC Parameters" section on page 30.

To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or quality of service (QoS), to the PVC under the PVC submode, refer to the appropriate Cisco IOS XR software configuration guide.

To configure a vc-class and apply it to an ATM subinterface or PVC, see the "Creating and Configuring a VC-Class" section.

Configuring Optional Point-to-Point ATM PVC Parameters

This task describes the commands you can use to modify the default configuration on an ATM PVC.

Before you can modify the default PVC configuration, you must create the PVC on an ATM subinterface, as described in the Creating a Point-to-Point ATM Subinterface with a PVC.

Restrictions

The configuration on both ends of the PVC must match for the connection to be active.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **point-to-point**
3. **pvc** *vpi*/*vci*

4. **encapsulation** {**aal5mux ipv4** | **aal5nlpid** | **aal5snap**}
5. **oam-pvc manage** [*frequency*] [**disable**] [**keep-vc-up** [**seg-aisrdi-failure**]
6. **oam ais-rdi** [*down-count* [*up-count*]]
7. **oam retry** [*up-count* [*down-count* [*retry-frequency*]]]
8. **shape** [**cbr** *peak_output_rate* | **ubr** *peak_output_rate*| **vbr-nrt** *peak_output_rate sustained_output_rate burst_size*| **vbr-rt** *peak_output_rate sustained_output_rate burst_size*]
9. **service-policy** [**input** | **output**] *policy_name*
10. **end** or **commit**
11. Repeat Step 1 through Step 10 to configure the PVC at the other end of the connection.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id.subinterface* **point-to-point** **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1.10 point-to-point`	Enters ATM subinterface configuration mode.
Step 3	**pvc** *vpi*/*vci* **Example:** `RP/0/RP0/CPU0:router (config-subif)# pvc 5/10`	Enters subinterface configuration mode for the PVC.
Step 4	**encapsulation** {**aal5mux ipv4** | **aal5nlpid** | **aal5snap**} **Example:** `RP/0/RP0/CPU0:router (config-atm-vc)# encapsulation aal5snap`	Configures the ATM adaptation layer (AAL) and encapsulation type for a PVC. **Note** The default encapsulation type for a vc-class is AAL5/SNAP
Step 5	**oam-pvc manage** [*frequency*] [**disable**] [**keep-vc-up** [**seg-aisrdi-failure**] **Example:** `RP/0/RP0/CPU0:router (config-atm-vc)# oam-pvc manage 200 keep-vc-up`	Enable ATM OAM F5 loopback cell generation and configures continuity check (CC) management for the ATM permanent virtual circuit (PVC). • Include the **disable** keyword to disable OAM management on the specified PVC. • Include the **keep-vc-up** keyword specify that PVC remains in the UP state when CC cells detect connectivity failure. • Include the **seg-aisrdi-failure** keyword to specify that, if segment AIS/RDI cells are received, the VC

	Command or Action	Purpose
		will not be brought down because of end CC failure or loopback failure.
Step 6	**oam ais-rdi** [*down-count* [*up-count*]] **Example:** `RP/0/RP0/CPU0:router (config-atm-vc)# oam ais-rdi 25 5`	Configures the PVC so that it is brought down after a specified number of OAM alarm indication signal/remote defect indication (AIS/RDI) cells are received on the associated PVC.
Step 7	**oam retry** [*up-count* [*down-count* [*retry-frequency*]]] **Example:** `RP/0/RP0/CPU0:router (config-atm-vc)# oam retry 5 10 5`	Configures parameters related to OAM management for the PVC. If no OAM AIS/RDI cells are received within the specified interval, the PVC is brought up.
Step 8	**shape** [**cbr** *peak_output_rate* | **ubr** *peak_output_rate*| **vbr-nrt** *peak_output_rate sustained_output_rate burst_size*| **vbr-rt** *peak_output_rate sustained_output_rate burst_size*] **Example:** `RP/0/RP0/CPU0:router (config-atm-vc)# shape vbr-nrt 100000 100000 8000`	Configures ATM traffic shaping for the PVC. You must estimate how much bandwidth is required before you configure ATM traffic shaping. • *peak_output_rate*—Configures the maximum cell rate that is always available for the traffic. • *Sustained_output_rate*—Sustained output rate for the bit rate. • *burst size*—Burst cell size for the bit rate. Range is from 1 through 8192.
Step 9	**service-policy** [**input** | **output**] *policy_name* **Example:** `RP/0/RP0/CPU0:router (config-atm-vc)# service-policy input policyA`	Attaches a QoS policy to an input or output PVC. Replace *policy_name* with the name of the service policy you want to attach to the PVC. **Note** For information on creating and configuring service policies, see the *Cisco IOS XR Modular Quality of Service Configuration Guide.*

Step 10	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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.

	Command or Action	Purpose
		Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 11	Repeat Step 1 through Step 10 to configure the PVC at the other end of the connection.	Brings up the connection. **Note** The configuration on both ends of the connection must match.

What to do next

- To attach Layer 3 service policies, such as MPLS or QoS, to the PVC under the PVC submode, refer to the appropriate Cisco IOS XR software configuration guide.
- To configure a vc-class and apply it to an ATM subinterface or PVC, see the “Creating and Configuring a VC-Class” section.

Creating and Configuring a VP-Tunnel on an ATM Interface

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id*
3. **vp-tunnel** *vpi*
4. **f4oam disable**
5. **shape** [**cbr** *peak_output_rate* | **vbr-nrt** *peak_output_rate sustained_output_rate burst_size* | **vbr-rt** *peak_output_rate sustained_output_rate burst_size*]
6. **end** or **commit**
7. **exit**
8. **exit**
9. **show atm vp-tunnel interface atm** [*interface-path-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id* **Example:**	Enters ATM interface configuration mode.

	Command or Action	Purpose
	`RP/0//CPU0:router (config)# interface atm 0/6/0/1`	
Step 3	**vp-tunnel** *vpi* **Example:** `RP/0//CPU0:router (config)# vp-tunnel 10`	Configures a vp-tunnel on an ATM interface.
Step 4	**f4oam disable** **Example:** `RP/0//CPU0:router(config-atm-vp-tunnel)# f4oam disable`	(Optional) Disables the passing of OAM packets.
Step 5	**shape** [**cbr** *peak_output_rate* | **vbr-nrt** *peak_output_rate sustained_output_rate burst_size* | **vbr-rt** *peak_output_rate sustained_output_rate burst_size*] **Example:** `RP/0//CPU0:router (config-if)# shape`	Configures ATM traffic shaping for the PVC. You must estimate how much bandwidth is required before you configure ATM traffic shaping. • *peak_output_rate*—Configures the maximum cell rate that is always available for the traffic. • *Sustained_output_rate*—Sustained output rate for the bit rate. • *burst size*—Burst cell size for the bit rate. Range is from 1 through 8192. **Note** • After you configure traffic shaping on the vp-tunnel, you cannot configure traffic shaping directly on the PVCs configured under that vp-tunnel. If you attempt to use the **shape** command on a PVC that is configured under a tunnel, the command is rejected.

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

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**exit** **Example:** `RP/0//CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:** `RP/0//CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode. Repeat Step 1 through Step 8 to bring up the vp-tunnel at the other end of the connection. Brings up the vp-tunnel.
Step 9	**show atm vp-tunnel interface atm** [*interface-path-id*] **Example:** `RP/0//CPU0:router (config)# show atm vp-tunnel interface atm 0/6/0/1`	Displays vp-tunnel information for the entire router or a specific ATM interface.

To attach Layer 3 service policies, such as MPLS or QoS, to the vp-tunnel or its PVCs, refer to the appropriate Cisco IOS XR Software configuration guide.

Creating and Configuring Subinterfaces with PVCs on a VP-tunnel

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **point-to-point**
3. **ipv4 address** *ipv4_address/prefix*
4. **pvc** *vpi*/*vci*
5. **end** or **commit**
6. **ping atm interface atm** *interface-path-id.subinterface vpi*/*vci* RP/0//CPU0:router # ping atm interface atm 0/2/0/0.10 10/100
7. **show atm vp-tunnel** [**interface atm** *interface-path-id*]

DETAILED STEPS

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

	Command or Action	Purpose
	`/CPU0:router# configure`	
Step 2	**interface atm** *interface-path-id.subinterface* **point-to-point** **Example:** `RP/0//CPU0:router (config)# interface atm 0/6/0/1.10 point-to-point`	Creates a new subinterface and enters ATM subinterface configuration mode for that subinterface.
Step 3	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0//CPU0:router (config-subif)#ipv4 address 10.46.8.6/24`	Assigns an IP address and subnet mask to the subinterface.
Step 4	**pvc** *vpi*/*vci* **Example:** `RP/0//CPU0:router (config-subif)# pvc 5/10`	Creates an ATM permanent virtual circuit (PVC) on the subinterface and attaches it to the vp-tunnel you created in the Creating and Configuring a VP-Tunnel on an ATM Interface. Replace *vpi* with the VPI of the vp-tunnel on which you are creating the PVC. **Note** • The PVC VPI and vp-tunnel VCI must match or the connection will not be active. • A vp-tunnel is not usable until you create PVCs under it.

Step 5	**end** or **commit**
	Example:
	`RP/0//CPU0:router (config-subif)# end`
	or
	`RP/0//CPU0:router(config-subif)# 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.

	Command or Action	Purpose
		Repeat Step 1 through Step5 to bring up the subinterface and PVC at the other end of the vp-tunnel. Brings up the subinterface and PVC.
Step 6	**ping atm interface atm** *interface-path-id.subinterface vpi/vci* RP/0//CPU0:router # ping atm interface atm 0/2/0/0.10 10/100	Verifies connectivity between two ATM connection endpoints through the vp-tunnel you configured in Step 1 through Step 6. • Replace *interface-path-id.subinterface* with the ATM subinterface that is configured on the vp-tunnel whose connectivity you want to verify. This is the same *interface-path-id.subinterface* you configured in Step 2. • Replace *vci* with the VCI of the PVC configured on the vp-tunnel whose connectivity you want to verify. This is the same *vci* you configured in Step 4. • Replace *vpi* with the VPI of the PVC that is configured on the vp-tunnel whose connectivity you want to verify. This is the same *vpi* you configured in Step 4.
Step 7	**show atm vp-tunnel** [**interface atm** *interface-path-id*] **Example:** `RP/0//CPU0:router (config)# show atm vp-tunnel interface atm 0/6/0/1`	Displays vp-tunnel information for the entire router or a specific ATM interface.

What to do next

To create and configure ATM subinterfaces and PVCs on a vp-tunnel, see the “Creating and Configuring Subinterfaces with PVCs on a VP-tunnel” section on page 35

To configure a vc-class and apply it to an ATM interface, see the “Creating and Configuring a VC-Class” section on page 50.

How to Configure a Layer 2 Attachment Circuit

The Layer 2 AC configuration tasks are described in these procedures:

Note After you configure an interface for Layer 2 switching, no routing commands such as **ipv4 address** are permissible. If any routing commands are configured on the interface, then the **l2transport** command is rejected.

Creating a Layer 2 Port Mode AC

The procedure in this section creates a Layer 2 port mode AC.

Before you begin

Before you can create a Layer 2 port mode AC, you must bring up an ATM main interface, as described in the Bringing Up an ATM Interface.

Restrictions

ILMI configuration is not supported on Layer 2 port mode ACs. Before you can configure an Layer 2 port mode AC, you must ensure that no configuration, such as subinterfaces, already exists on that port. If any preconfiguration does exist, you must remove it.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id*
3. **l2transport**
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1`	Enters interface configuration mode for an ATM interface.
Step 3	**l2transport** **Example:** `RP/0/RP0/CPU0:router (config-if)# l2transport`	Enters ATM Layer 2 transport configuration mode, and enables Layer 2 port mode on this ATM interface.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-if-l2)# end`
	or
	`RP/0/RP0/CPU0:router(config-if-l2)# 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.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. Repeat Step 1 through Step 4 to bring up the Layer 2 port mode AC at the other end of the connection. Brings up the Layer 2 port mode AC. **Note** • The configuration on both ends of the connection must match.

What to do next

To configure a point-to-point pseudowire XConnect on the Layer 2 port mode AC you just created, see the Implementing MPLS Layer 2 VPNs module of Cisco IOS XR Multiprotocol Label Switching Configuration Guide.

To configure optional Layer 2 VPN parameters for the ATM AC, see the "Configuring Optional Parameters on a Layer 2 Port Mode AC" section on page 39.

Configuring Optional Parameters on a Layer 2 Port Mode AC

The procedure in this section configures optional Layer 2 VPN transport parameters on a Layer 2 port mode AC.

Before you begin

Before you can configure Layer 2 VPN parameters on a Layer 2 port mode AC, you must create a Layer 2 port mode AC, as described in the Creating a Layer 2 Port Mode AC.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id*
3. **atm mcpt-timers** *timer-1 timer-2 timer-3*
4. **l2transport**
5. **cell-packing** *cells timer* RP/0/RP0/CPU0:router (config-if-l2)# cell-packing 6 1
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1`	Enters interface configuration mode for an ATM interface.
Step 3	**atm mcpt-timers** *timer-1 timer-2 timer-3* **Example:** `RP/0/RP0/CPU0:router (config-if)# ` **`atm mcpt-timers 50 100 200`**	Specifies the maximum cell packing timeout values for each of the three per-interface MCPT timers, in microseconds. **Note** • The default value for each timer is 50 microseconds.
Step 4	**l2transport** **Example:** `RP/0/RP0/CPU0:router (config-if)# l2transport`	Enters ATM Layer 2 transport configuration mode.
Step 5	**cell-packing** *cells timer* RP/0/RP0/CPU0:router (config-if-l2)# cell-packing 6 1	Sets the maximum number of cells allowed per packet, and specifies a maximum cell packing timeout (MCPT) timer to use for cell packing. • Replace *cells* with the maximum number of cells to use per packet. Range is from 2 through 86. • Replace *timer* with the number that indicates the appropriate MCPT timer to use for cell packing. Can be **1**, **2**, or **3**. You can configure up to three different MCPT values for a single main interface.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-if-l2)# end`
	or
	`RP/0/RP0/CPU0:router(config-if-l2)# 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.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. Repeat Step 1 through Step 6 to configure the AC at the other end of the connection. Brings up the Layer 2 port mode AC. **Note** • The configuration on both ends of the connection must match.

Creating an ATM Layer 2 Subinterface with a PVC

The procedure in this section creates a Layer 2 subinterface with a PVC.

Before you begin

Before you can create a subinterface on an ATM interface, you must bring up an ATM interface, as described in the Bringing Up an ATM Interface.

Restrictions

Only one PVC can be configured for each ATM subinterface.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **l2transport**
3. **pvc** *vpi*/*vci*
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface atm** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config)# interface atm 0/6/0/1.10 l2transport`	Creates a subinterface and enters ATM subinterface configuration mode for that subinterface.
Step 3	**pvc** *vpi*/*vci* **Example:** `RP/0/RP0/CPU0:router(config-if)# pvc 5/20`	Creates an ATM permanent virtual circuit (PVC) and enters ATM Layer 2 transport PVC configuration mode. **Note** • Only one PVC is allowed per subinterface.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# end`
	or
	`RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# 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.
		Repeat Step 1 through Step 4 to bring up the ATM subinterface and any associated PVC at the other end of the AC.
		Brings up the AC.
		Note • The configuration on both ends of the AC must match.

What to do next

To configure optional PVC parameters, see the “Configuring Optional ATM Layer 2 PVC Parameters” section on page 43.

To configure a vc-class and apply it to the PVC, see the“Attaching a VC-Class to a PVC on an ATM Subinterface” section on page 55.

To configure a point-to-point pseudowire XConnect on the AC you just created, see the Implementing MPLS Layer 2 VPNs module of Cisco IOS XR Multiprotocol Label Switching Configuration Guide.

Configuring Optional ATM Layer 2 PVC Parameters

This task describes the commands you can use to modify the default configuration on an ATM Layer 2 PVC.

Before you begin

Before you can modify the default PVC configuration, you must create the PVC on a Layer 2 ATM subinterface, as described in the Creating an ATM Layer 2 Subinterface with a PVC.

Restrictions

The configuration on both ends of the PVC must match for the connection to be active.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **l2transport**
3. **pvc** *vpi/vci*
4. **encapsulation** {**aal0** | **aal5**}
5. **cell-packing** *cells timer*
6. **shape** [**cbr** *peak_output_rate* | **ubr** *peak_output_rate*| **vbr-nrt** *peak_output_rate sustained_output_rate burst_size*| **vbr-rt** *peak_output_rate sustained_output_rate burst_size*]
7. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config-if)# interface atm 0/6/0/1.10 l2transport`	Enters ATM subinterface configuration mode for a Layer 2 ATM subinterface.
Step 3	**pvc** *vpi/vci* **Example:** `RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# pvc 5/20`	Enters ATM Layer 2 transport PVC configuration mode for the specified PVC.
Step 4	**encapsulation** {**aal0** | **aal5**} **Example:**	Configures the ATM adaptation layer (AAL) and encapsulation type for a PVC.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td><code>RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# encapsulation aal5</code></td><td>Note • The default encapsulation type for a PVC is AAL5.</td></tr>
<tr><td>Step 5</td><td>cell-packing cells timer
Example:
<code>RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# cell-packing 5 2</code></td><td>Sets the maximum number of cells allowed per packet, and specifies a maximum cell packing timeout (MCPT) timer to use for cell packing.
• Replace cells with the maximum number of cells to use per packet. Range is from 2 through 86.
• Replace timer with the number that indicates the appropriate MCPT timer to use for cell packing. Can be 1, 2, or 3. You can configure up to three different MCPT values for a single main interface.</td></tr>
<tr><td>Step 6</td><td>shape [cbr peak_output_rate | ubr peak_output_rate| vbr-nrt peak_output_rate sustained_output_rate burst_size| vbr-rt peak_output_rate sustained_output_rate burst_size]
Example:
<code>RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# shape vbr-nrt 100000 100000 8000</code></td><td>Configures ATM traffic shaping for the PVC.
You must estimate how much bandwidth is required before you configure ATM traffic shaping.
• peak_output_rate—Configures the maximum cell rate that is always available for the traffic.
• Sustained_output_rate—Sustained output rate for the bit rate.
• burst size—Burst cell size for the bit rate. Range is from 1 through 8192.</td></tr>
<tr><td>Step 7</td><td>end or commit
Example:
<code>RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# end</code>
or
<code>RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# commit</code></td><td>Saves configuration changes.
• When you issue the end command, the system prompts you to commit changes:
<code>Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:</code>
• 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.</td></tr>
</table>

	Command or Action	Purpose
		Repeat Step 1 through Step 7 to configure the PVC at the other end of the AC. Brings up the AC. **Note** • The configuration on both ends of the connection must match.

What to do next

To configure a pseudo-wire XConnect on the AC you just created, see the Implementing MPLS Layer 2 VPNs module of Cisco IOS XR Multiprotocol Label Switching Configuration Guide.

To configure a vc-class and apply it to the PVC, see the"Attaching a VC-Class to a PVC on an ATM Subinterface" section on page 54.

Creating an ATM Layer 2 Subinterface with a PVP

The procedure in this section creates an ATM Layer 2 subinterface with a permanent virtual path (PVP) on that ATM subinterface.

Before you begin

Before you can create a subinterface with a PVP on an ATM interface, you must bring up an ATM interface, as described in the Bringing Up an ATM Interface.

Restrictions

- Only one PVP can be configured for each L2VPN ATM AC.
- F4 OAM emulation is not supported on Layer 2 PVPs.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **l2transport**
3. **pvp** *vpi*
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id.subinterface* **l2transport** **Example:**	Creates an ATM subinterface and enters ATM layer2 transport configuration mode for that subinterface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# interface atm 0/6/0/1.10 l2transport`	
Step 3	**pvp** *vpi* **Example:** `RP/0/RP0/CPU0:router(config-if)# pvp 100`	(Optional) Creates an ATM PVP and enters ATM PVP configuration submode. **Note** • Only one PVP is allowed per subinterface.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# end`
	or
	`RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# 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.
		Repeat Step 1 through Step 4 to bring up the ATM subinterface and any associated PVP at the other end of the AC.
		Brings up the ATM AC.
		Note • The configuration on both ends of the AC connection must match.

What to do next

To configure optional PVP parameters, see the "Configuring Optional ATM Layer 2 PVP Parameters" section on page 46.

To configure a point-to-point pseudowire XConnect on the AC you just created, see the Implementing MPLS Layer 2 VPNs module of Cisco IOS XR Multiprotocol Label Switching Configuration Guide.

Configuring Optional ATM Layer 2 PVP Parameters

This task describes the commands you can use to modify the default configuration on an ATM Layer 2 PVP.

Before you begin

Before you can modify the default PVP configuration, you must create the PVP on an ATM subinterface, as described in the Creating an ATM Layer 2 Subinterface with a PVP.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **l2transport**
3. **pvp** *vpi*
4. **cell-packing** *cells timer*
5. **shape** [**cbr** *peak_output_rate* | **ubr** *peak_output_rate*| **vbr-nrt** *peak_output_rate sustained_output_rate burst_size*| **vbr-rt** *peak_output_rate sustained_output_rate burst_size*]
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config)# interface atm 0/6/0/1.10 l2transport`	Enters ATM subinterface configuration mode.
Step 3	**pvp** *vpi* **Example:** `RP/0/RP0/CPU0:router(config-if)# pvp 10`	Enters subinterface configuration mode for the PVP.
Step 4	**cell-packing** *cells timer* **Example:** `RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# cell-packing 5 2`	Sets the maximum number of cells allowed per packet, and specifies a maximum cell packing timeout (MCPT) timer to use for cell packing. • Replace *cells* with the maximum number of cells to use per packet. Range is from 2 through 86. • Replace *timer* with the number that indicates the appropriate MCPT timer to use for cell packing. Can be **1**, **2**, or **3**. You can configure up to three different MCPT values for a single main interface.

	Command or Action	Purpose
Step 5	**shape** [**cbr** *peak_output_rate* | **ubr** *peak_output_rate*| **vbr-nrt** *peak_output_rate sustained_output_rate burst_size*| **vbr-rt** *peak_output_rate sustained_output_rate burst_size*] **Example:** `RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# shape vbr-nrt 100000 100000 8000`	Configures ATM traffic shaping for the PVC. You must estimate how much bandwidth is required before you configure ATM traffic shaping. • *peak_output_rate*—Configures the maximum cell rate that is always available for the traffic. • *Sustained_output_rate*—Sustained output rate for the bit rate. • *burst size*—Burst cell size for the bit rate. Range is from 1 through 8192.

Step 6	**end** or **commit**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# end`
	
	or
	
	`RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# 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.
		
		Repeat Step 1 through Step 6 to configure the PVP at the other end of the AC.
		
		Brings up the AC.
		
		Note • The configuration on both ends of the AC connection must match.

To configure a point-to-point pseudowire XConnect on the AC you just created, see the Implementing MPLS Layer 2 VPNs module of Cisco IOS XR Multiprotocol Label Switching Configuration Guide.

How to Create and Configure a VC-Class

The configuration tasks for creating and configuring an ATM VC-Class are described in these procedures:

Creating and Configuring a VC-Class

This section describes the tasks and commands required to create a virtual circuit (VC) class and attach it to an ATM main interface, subinterface, or permanent virtual circuit (PVC).

Restrictions

For Layer 2 VPN AC configurations, VC-classes can be applied to PVCs only. VC-classes are not supported on Layer 2 port mode interfaces or PVPs.

SUMMARY STEPS

1. **configure**
2. **vc-class atm name**
3. **encapsulation** {**aal5mux ipv4** | **aal5nlpid** | **aal5snap**}
4. **oam ais-rdi** [*down-count* [*up-count*]]
5. **oam retry** [*up-count* [*down-count* [*retry-frequency*]]]
6. **oam-pvc manage** *seconds*
7. **shape** [**cbr** *peak_output_rate* | **ubr** *peak_output_rate* | **vbr-nrt** *peak_output_rate sustained_output_rate burst_size*| **vbr-rt** *peak_output_rate sustained_output_rate burst_size*]
8. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**vc-class atm name** **Example:** `RP/0/RP0/CPU0:router (config)# vc-class atm class1`	Creates a vc-class for an ATM interface and enters vc-class configuration mode.
Step 3	**encapsulation** {**aal5mux ipv4** | **aal5nlpid** | **aal5snap**} **Example:** `RP/0/RP0/CPU0:router (config-vc-class-atm)# encapsulation aal5snap`	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM vc-class. **Note** • The default encapsulation type for a vc-class is AAL5/SNAP • In Vc-classes, the **encapsulation** command applies to Layer 3 Point-to-point configurations only.

	Command or Action	Purpose
Step 4	**oam ais-rdi** [*down-count* [*up-count*]] **Example:** `RP/0/RP0/CPU0:router (config-vc-class-atm)# oam ais-rdi 25 5`	Configures the vc-class so that it is brought down after a specified number of OAM alarm indication signal/remote defect indication (AIS/RDI) cells are received on the associated PVC. **Note** • In vc-classes, the **oam ais-rdi** command applies to Layer 3 Point-to-point configurations only.
Step 5	**oam retry** [*up-count* [*down-count* [*retry-frequency*]]] **Example:** `RP/0/RP0/CPU0:router (config-vc-class-atm)# oam retry 5 10 5`	Configures parameters related to OAM management. **Note** • In vc-classes, the **oam retry** command applies to Layer 3 Point-to-point configurations only.
Step 6	**oam-pvc manage** *seconds* **Example:** `RP/0/RP0/CPU0:router (config-vc-class-atm)# oam-pvc manage 300`	Configures the ATM OAM F5 loopback frequency. **Note** • In vc-classes, the **oam-pvc manage** command applies to Layer 3 Point-to-point configurations only.
Step 7	**shape** [**cbr** *peak_output_rate* | **ubr** *peak_output_rate* | **vbr-nrt** *peak_output_rate sustained_output_rate burst_size*| **vbr-rt** *peak_output_rate sustained_output_rate burst_size*] **Example:** `RP/0/RP0/CPU0:router (config-vc-class-atm)# shape vbr-nrt 100000 100000 8000`	Configures ATM traffic shaping for the PVC. You must estimate how much bandwidth is required before you configure ATM traffic shaping. • *peak_output_rate*—Configures the maximum cell rate that is always available for the traffic. • *Sustained_output_rate*—Sustained output rate for the bit rate. • *burst size*—Burst cell size for the bit rate. Range is from 1 through 8192.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

What to do next

Attach the vc-class to an ATM main interface, subinterface, or PVC.

To attach a vc-class to an ATM main interface, see the "Attaching a VC-Class to a Point-to-Point ATM Main Interface" section on page -50.

To attach a vc-class to an ATM subinterface, see the "Attaching a VC-Class to a Point-to-Point ATM Subinterface" section on page -52.

To attach a vc-class to an ATM PVC, see the "Attaching a VC-Class to a PVC on an ATM Subinterface" section on page -53.

Attaching a VC-Class to a Point-to-Point ATM Main Interface

This section describes the tasks and commands required to attach a vc-class to a point-to-point ATM main interface.

Restrictions

VC-classes are not applicable to Layer 2 port mode ACs. For Layer 2 VPN configurations, Vc-classes are applicable to the PVC only.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id* **point-to-point**
3. **class-int** *vc-class-name*
4. **end** or **commit**

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1 point-to-point`	
Step 3	**class-int** *vc-class-name* **Example:** `RP/0/RP0/CPU0:router (config-if)# class-int classA`	Attaches the vc-class to an ATM main interface. Replace the *vc-class-name* argument with the name of the vc-class you configured in the Creating and Configuring a VC-Class.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Attaching a VC-Class to a Point-to-Point ATM Subinterface

This section describes the tasks and commands required to attach a vc-class to an ATM subinterface.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id.subinterface* **point-to-point**
3. **class-int** *vc-class-name*
4. **end** or **commit**

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**interface atm** *interface-path-id.subinterface* **point-to-point** **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1.10 point-to-point`	Enters ATM subinterface configuration mode.
Step 3	**class-int** *vc-class-name* **Example:** `RP/0/RP0/CPU0:router (config-subif)# class-int classA`	Assigns the vc-class to an ATM subinterface. Replace the *vc-class-name* argument with the name of the vc-class you configured in the Creating and Configuring a VC-Class.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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.

Attaching a VC-Class to a PVC on an ATM Subinterface

This section describes the tasks and commands required to attach a vc-class to a PVC on an ATM subinterface.

Note VC-classes are supported on point-to-point and Layer 2 PVCs.

SUMMARY STEPS

1. **configure**

2. **interface atm** *interface-path-id.subinterface* [**point-to-point** | **l2transport**]
3. **pvc** *vpi/vci*
4. **class-vc** *vc-class-name*
5. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id.subinterface* [**point-to-point** | **l2transport**] **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1.10`	Enters subinterface configuration mode and creates the ATM subinterface if it does not already exist. Use the **point-to-point** keyword if you are attaching the vc-class to a point-to-point subinterface. Use the **l2transport** keyword if you are attaching the vc-class to a Layer 2 transport subinterface. **Note** • For more information on creating and configuring ATM subinterfaces, see the Creating a Point-to-Point ATM Subinterface with a PVC.
Step 3	**pvc** *vpi/vci* **Example:** `RP/0/RP0/CPU0:router (config-if)# pvc 5/50`	Enters ATM PVC configuration mode and creates the PVC if it does not already exist. **Note** • For more information on creating and configuring PVCs on ATM subinterfaces, see the Creating a Point-to-Point ATM Subinterface with a PVC.
Step 4	**class-vc** *vc-class-name* **Example:** `RP/0/RP0/CPU0:router (config-atm-vc)# class-vc classA`	Assigns a vc-class to an ATM PVC. Replace the *vc-class-name* argument with the name of the vc-class you want to attach to the PVC.

Step 5	**end** or **commit**
	
	Example:
	
	`RP/0/RP0/CPU0:router (config-if)# end`
	
	or
	
	`RP/0/RP0/CPU0:router(config-if)# 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]:`

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

How to Configure ILMI on ATM Interfaces

The configuration tasks for managing ILMI on ATM interfaces are described in the following procedures:

Enabling ILMI on an ATM Interface

This task describes the commands you can use to configure an ATM interface for ILMI.

Note For ILMI, a PVC is configured directly on the ATM main interface. Subinterface configuration is not necessary for ATM interfaces that are used for ILMI.

Before you begin

Bring up the ATM interface and remove the shutdown configuration, as described in the Bringing Up an ATM Interface.

Restrictions

- The configuration on both ends of the ATM ILMI connection must match for the interface to be active.
- ILMI configuration is not supported on Layer 2 port mode ACs.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id*
3. **atm address-registration**
4. **atm ilmi-keepalive** [**act-poll-freq** *frequency*] [**retries** *count*] [**inact-poll-freq** *frequency*]
5. **pvc** *vpi*/*vci* **ilmi**
6. **end**
7. **commit** RP/0/RP0/CPU0:router (config-if)# end

8. **exit**
9. **exit**
10. **show atm ilmi-status** [**atm** *interface-path-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1`	Enters ATM interface configuration mode.
Step 3	**atm address-registration** **Example:** `RP/0/RP0/CPU0:router (config-if)# atm address-registration`	(Optional) Enables the router to engage in address registration and callback functions with the Interim Local Management Interface (ILMI).
Step 4	**atm ilmi-keepalive** [**act-poll-freq** *frequency*] [**retries** *count*] [**inact-poll-freq** *frequency*] **Example:** `RP/0/RP0/CPU0:router (config-if)# atm ilmi-keepalive`	(Optional) Enables ILMI keepalives on an ATM interface.
Step 5	**pvc** *vpi*/*vci* **ilmi** **Example:** `RP/0/RP0/CPU0:router (config-if)# pvc 5/30 ilmi`	Creates an ATM permanent virtual circuit (PVC) with ILMI encapsulation.
Step 6	**end**	or
Step 7	**commit** RP/0/RP0/CPU0:router (config-if)# end **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	or 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.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**exit** **Example:** `RP/0/RP0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode.
Step 10	**show atm ilmi-status** [**atm** *interface-path-id*] **Example:** `RP/0/RP0/CPU0:router (config)# show atm ilmi-status atm 0/6/0/1`	(Optional) Verifies the ILMI configuration for the specified interface.

Disabling ILMI on an ATM Interface

This task describes the commands you can use to disable ILMI on an ATM interface.

SUMMARY STEPS

1. **configure**
2. **interface atm** *interface-path-id*
3. **atm ilmi-config disable**
4. **end** or **commit**
5. **exit**
6. **exit**
7. **show atm ilmi-status** [**atm** *interface-path-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1`	Enters ATM interface configuration mode.
Step 3	**atm ilmi-config disable** **Example:** `RP/0/RP0/CPU0:router (config-if)# atm ilmi-config disable`	(Optional) Disables ILMI on the ATM interface. To re-enable ILMI on an ATM interface, use the **no atm ilmi-config disable** form of this command.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**exit** **Example:**	Exits global configuration mode and enters EXEC mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router (config)# exit`	
Step 7	**show atm ilmi-status** [**atm** *interface-path-id*] **Example:** `RP/0/RP0/CPU0:router (config)# show atm ilmi-status atm 0/6/0/1`	(Optional) Verifies the ILMI configuration for the specified interface.

How to Configure Channelized ATM

SUMMARY STEPS

1. **configure**
2. **hw-module subslot** *subslot-id* **cardtype** {**t3** \| **e3**}
3. **controller t3** *interface-path-id*
4. **mode** *mode*
5. **controller t1** *interface-path-id*
6. **mode** *mode*
7. **interface atm** *interface-path-id*
8. **interface atm** *interface-path-id.subinterface* **point-to-point**
9. **pvc** *vpi/vci*
10. **ipv4 address** *ipv4_address/prefix*
11. **end** or **commit**
12. **hw-module subslot** *subslot-id* **cardtype** {**t3** \| **e3**}
13. **controller t3** *interface-path-id*
14. **mode** *mode*
15. **interface atm** *interface-path-id*
16. **vp-tunnel** *vpi*
17. **interface atm** *interface-path-id.subinterface* **point-to-point**
18. **pvc** *vpi/vci*
19. **ipv4 address** *ipv4_address/prefix*
20. **interface atm** *interface-path-id.subinterface* **point-to-point**
21. **pvc** *vpi/vci*
22. **ipv4 address** *ipv4_address/prefix*
23. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**hw-module subslot** *subslot-id* **cardtype** {**t3** | **e3**} **Example:** `RP/0/RP0/CPU0:router(config)# hw-module subslot 0/1/0 cardtype t3`	Sets the card type for the SPA. • t3—Specifies T3 connectivity of 44,210 kbps through the network, using B3ZS coding. This is the default setting. • e3—Specifies a wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 34,010 kbps.
Step 3	**controller t3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0`	Creates a T3 controller and enters the T3 controller configuration mode. Specifies the T3 controller interface-path-id identifier with the *rack/slot/module/port* notation.
Step 4	**mode** *mode* **Example:** `RP/0/RP0/CPU0:router(config-t3)# mode t1`	Sets the mode of interface. The possible modes are: • atm—clear channel carrying atm • e1—channelize into 21 E1s • serial—clear channel carrying hdlc like payload • t1—channelized into 28 T1s
Step 5	**controller t1** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-t3)# controller t1 0/1/0/0`	Creates a T1 controller and enters the T1 controller configuration submode. Specifies the T1 controller *interface-path-id* with the *rack/slot/module/port* notation.
Step 6	**mode** *mode* **Example:** `RP/0/RP0/CPU0:router(config-t1)# mode atm`	Sets the mode of interface. The possible modes are: • atm—clear channel carrying atm • e1—channelize into 21 E1s • serial—clear channel carrying hdlc like payload • t1—channelized into 28 T1s
Step 7	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-t1)# interface atm 0/1/0/0`	Creates an ATM interface and enters ATM interface configuration mode. Specifies the ATM interface with the *rack/slot/module/port* notation.
Step 8	**interface atm** *interface-path-id.subinterface* **point-to-point** **Example:**	Creates an ATM subinterface as one endpoint of a point-to-point link and enters ATM subinterface configuration mode. Specifies the ATM interface with the *rack/slot/module/port.subinterface* notation.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# interface atm 0/1/0/1.1 point-to-point`	
Step 9	**pvc** *vpi*/*vci* **Example:** `RP/0/RP0/CPU0:router(config-subif)# pvc 10/100`	Creates an ATM permanent virtual circuit (PVC) and enters ATM PVC configuration submode. **Note** • Only one PVC is allowed per subinterface.
Step 10	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-atm-vc)#ipv4 address 10.212.4.22 255.255.255.0`	Assigns an IP address and subnet mask to the subinterface.

Step 11	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-sonet)# end`
	or
	`RP/0/RP0/CPU0:router(config-sonet)# commit`
	Example:
	`RP0/CPU0:router# configure`
		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.
		configure
		Enters global configuration mode.

Step 12	**hw-module subslot** *subslot-id* **cardtype** {**t3** | **e3**} **Example:** `RP/0/RP0/CPU0:router(config)# hw-module subslot 0/1/0 cardtype t3`	Sets the card type for the SPA. • t3—Specifies T3 connectivity of 44,210 kbps through the network, using B3ZS coding. This is the default setting. • e3—Specifies a wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 34,010 kbps.

	Command or Action	Purpose
Step 13	**controller t3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0`	Creates a T3 controller and enters the T3 controller configuration mode. Specifies the T3 controller *interface-path-id* with the *rack/slot/module/port* notation.
Step 14	**mode** *mode* **Example:** `RP/0/RP0/CPU0:router(config-t3)# mode t1`	Sets the mode of interface. The possible modes are: • atm—clear channel carrying atm • e1—channelize into 21 E1s • serial—clear channel carrying hdlc like payload • t1—channelized into 28 T1s
Step 15	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-t1)# interface atm 0/1/0/0`	Creates an ATM interface and enters ATM interface configuration mode. Specifies the ATM interface with the *rack/slot/module/port* notation.
Step 16	**vp-tunnel** *vpi* **Example:** `RP/0/RP0/CPU0:router (config)# vp-tunnel 10`	Configures a vp-tunnel on an ATM interface.
Step 17	**interface atm** *interface-path-id.subinterface* **point-to-point** **Example:** `RP/0/RP0/CPU0:router(config-if)# interface atm 0/1/0/1.1 point-to-point`	Creates an ATM subinterface as one endpoint of a point-to-point link and enters ATM subinterface configuration mode. Specifies the ATM interface with the *rack/slot/module/port.subinterface* notation.
Step 18	**pvc** *vpi*/*vci* **Example:** `RP/0/RP0/CPU0:router(config-subif)# pvc 10/100`	Creates an ATM permanent virtual circuit (PVC) and enters ATM PVC configuration submode. **Note** • Only one PVC is allowed per subinterface.
Step 19	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-atm-vc)#ipv4 address 10.212.8.22 255.255.255.0`	Assigns an IP address and subnet mask to the subinterface.
Step 20	**interface atm** *interface-path-id.subinterface* **point-to-point** **Example:**	Creates an ATM subinterface as one endpoint of a point-to-point link and enters ATM subinterface configuration mode. Specifies the ATM interface with the *rack/slot/module/port.subinterface* notation.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# interface atm 0/1/0/1.2 point-to-point`	
Step 21	**pvc** *vpi*/*vci* **Example:** `RP/0/RP0/CPU0:router(config-subif)# pvc 10/200`	Creates an ATM permanent virtual circuit (PVC) and enters ATM PVC configuration submode. **Note** • Only one PVC is allowed per subinterface.
Step 22	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-atm-vc)#ipv4 address 10.212.12.22 255.255.255.0`	Assigns an IP address and subnet mask to the subinterface.

Step 23	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-sonet)# end`
	or
	`RP/0/RP0/CPU0:router(config-sonet)# 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.

Attaching a Service-Policy to an Attachment Circuit

The QoS **service-policy** command can be configured for an attachment circuit in the following modes:

- PVC mode
- PVP mode
- Port mode
- Main Interface (non-port mode)

In PVC mode, PVP mode, and Port mode, the service policy is attached in the l2transport sub-interface mode. In non-port mode, the service policy is attached to the main interface.

Use the following procedures to attach a service-policy to an attachment circuit.

PVC Mode

SUMMARY STEPS

1. **config**
2. **interface atm** *interface-path-id.subinterface* **l2transport**
3. **interface atm** *interface-path-id*
4. **service-policy input | output** *policy_name*
5. **commit**
6. **config**
7. **interface atm** *interface-path-id.subinterface* **l2transport**
8. **pvp** *vpi*
9. **service-policy input | output** *policy_name*
10. **commit**
11. **config**
12. **interface atm** *interface-path-id*
13. **l2transport**
14. **service-policy input | output** *policy_name*
15. **commit**
16. **config**
17. **interface atm** *interface-path-id*
18. **service-policy input | output** *policy_name*
19. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config** **Example:** `RP/0/RP0/CPU0:router# config terminal`	Enters global configuration mode.
Step 2	**interface atm** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config)# interface atm 0/1/0/0.2 l2transport`	Creates a subinterface and enters ATM subinterface configuration mode for that subinterface.
Step 3	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/1/0/1`	Enters interface configuration mode for an ATM interface.

	Command or Action	Purpose
Step 4	**service-policy input | output** *policy_name* **Example:** `RP/0/RP0/CPU0(config-atm-l2transport-pvc)#service-policy input | output atm_policy_1`	Attaches the specified service policy to the ATM PVC subinterface.
Step 5	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes. PVP Mode
Step 6	**config** **Example:** `RP/0/RP0/CPU0:router# config terminal`	Enters global configuration mode.
Step 7	**interface atm** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config)# interface atm 0/1/0/0.2 l2transport`	Creates a subinterface and enters ATM subinterface configuration mode for that subinterface.
Step 8	**pvp** *vpi* **Example:** `RP/0/RP0/CPU0:router(config-subif)# pvp 30`	(Optional) Creates an ATM PVP and enters ATM PVP configuration submode. **Note** • Only one PVP is allowed per subinterface.
Step 9	**service-policy input | output** *policy_name* **Example:** `RP/0/RP0/CPU0(config-atm-l2transport-pvp)#service-policy input | output atm_policy_2`	Attaches the specified service policy to the ATM PVP subinterface.
Step 10	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes. Port Mode
Step 11	**config** **Example:** `RP/0/RP0/CPU0:router# config terminal`	Enters global configuration mode.
Step 12	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/1/0/1`	Enters interface configuration mode for an ATM interface.

	Command or Action	Purpose
Step 13	**l2transport** **Example:** `RP/0/RP0/CPU0:router (config-if)# l2transport`	Enters ATM Layer 2 transport configuration mode, and enables Layer 2 port mode on this ATM interface.
Step 14	**service-policy input | output** *policy_name* **Example:** `RP/0/RP0/CPU0(config-if-l2)#service-policy input | output atm_policy_3`	Attaches the specified service policy to the ATM Layer 2 subinterface.
Step 15	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes. Main Interface (non-port mode)
Step 16	**config** **Example:** `RP/0/RP0/CPU0:router# config terminal`	Enters global configuration mode.
Step 17	**interface atm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router (config)# interface atm 0/1/0/1`	Enters interface configuration mode for an ATM interface.
Step 18	**service-policy input | output** *policy_name* **Example:** `RP/0/RP0/CPU0(config-if)#service-policy input | output atm_policy_4`	Attaches the specified service policy to the main ATM interface.
Step 19	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes.

ATM Configuration: Examples

This section provides the following configuration examples:

ATM Interface Bring Up and Configuration: Example

The following example shows how to bring up and configure an ATM interface:

```
RP/0/RP0/CPU0:router #configure
RP/0/RP0/CPU0:router(config)# interface atm 0/6/0/0
RP/0/RP0/CPU0:router(config-if)# atm address-registration
RP/0/RP0/CPU0:router(config-if)# no shutdown

RP/0/RP0/CPU0:router(config-if)# commit
```

Point-To-Point ATM Subinterface Configuration: Example

The following example shows how to configure a point-to-point ATM subinterface on an ATM main interface:

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router (config)# interface atm 0/2/0/2.1 point-to-point
RP/0/RP0/CPU0:router (config-if)# ipv4 address 10.46.8.6/24
RP/0/RP0/CPU0:router (config-if)# pvc 0/200
RP/0/RP0/CPU0:router (config-atm-vc)# commit
RP/0/RP0/CPU0:router (config-atm-vc)# exit
RP/0/RP0/CPU0:router (config-if)# exit
RP/0/RP0/CPU0:router (config)# exit

RP/0/RP0/CPU0:router # show interfaces atm 0/2/0/2.1

ATM0/2/0/2.1 is up, line protocol is up
  Hardware is ATM network sub-interface(s)
  Description: Connect to P4_C12810 ATM 1/2.1
  Internet address is 10.46.8.6/24
  MTU 4470 bytes, BW 155000 Kbit
     reliability Unknown, txload Unknown, rxload Unknown
  Encapsulation AAL5/SNAP, controller loopback not set,
  Last clearing of "show interface" counters Unknown
  Datarate information unavailable.
  Interface counters unavailable.

RP/0/RP0/CPU0:router # show atm interface atm 0/2/0/3

Interface                             : ATM0/2/0/3
AAL Enabled                           : AAL5
Max-VP                                : 254
Max-VC                                : 2046
Configured L2 PVPs                    : 0
Configured L2 PVCs                    : 0
Configured L3 VP-Tunnels              : 0
Configured L3 PVCs                    : 1
L2 PVPs in Down State                 : 0
L2 PVCs in Down State                 : 0
L3 VP-Tunnels in Down State           : 0
L3 PVCs in Down State                 : 0
Cell packing count                    : 0

Received Side Statistics:
    Received Cells                    : 0
    Received Bytes                    : 0
    Received AAL Packets              : 0

Receive Side Cells Dropped:
    Unrecognized VPI/VCI              : 0

Receive Side AAL5 Packets Dropped:
    Unavailable SAR Buffer            : 0
    Non-Resource Exhaustion           : 0
```

```
    Reassembly Timeout               : 0
    Zero Length                      : 0
    Unavailable Host Buffer          : 0
    Packet size exceeds MPS          : 0
    AAL5 Trailer Length Errors       : 0

Transmit Side Statistics:
    Transmitted Cells                : 1899716067
    Transmitted Bytes                : 0
    Transmitted AAL Packets          : 0

Transmit Side Cells Dropped:
    Unrecognized VPI/VCI             : 0

Transmit Side AAL5 Packets Dropped:
    Unavailable SAR Buffer           : 0
    Non-Resource Exhaustion          : 0
    WRED Threshold                   : 0
    WRED Random                      : 0

RP/0/RP0/CPU0:router # show atm pvc 10/100

Detailed display of VC(s) with VPI/VCI = 10/100

ATM0/2/0/3.100: VPI: 10 VCI: 100
UBR, PeakRate: 622000 Kbps
AAL5-LLC/SNAP
OAM frequency: 10 second(s), OAM retry frequency: 1 second(s),
OAM up retry count: 3, OAM down retry count: 5,
OAM Keep-vc-up: False, OAM AIS-RDI failure: None,
OAM AIS-RDI down count: 1, OAM AIS-RDI up time: 3 second(s),
OAM Loopback status: No loopback enabled,
OAM VC state: Loopback Not verified,
VC is not managed by OAM,

OAM cells received: 0,
F5 InEndLoop: 0, F5 InSegLoop: 0,
F5 InEndAIS: 0,  F5 InSegAIS: 0,
F5 InEndRDI: 0,  F5 InSegRDI: 0,
OAM cells sent: 0,
F5 OutEndLoop: 0, F5 OutSegLoop: 0,
F5 OutEndAIS: 0,  F5 OutSegAIS: 0,
F5 OutEndRDI: 0,  F5 OutSegRDI: 0,
OAM cells drops: 0

InPkts: 0                    OutPkts: 0
InBytes: 0                   OutBytes: 0
WRED pkt drop: 0
Non WRED pkt drop: 0

Internal state: READY
Status: UP
```

Layer 2 AC Creation and Configuration: Example

This example shows how to create and configure one endpoint of a Layer 2 port mode AC:

```
RP/0/RP0/CPU0:router# configure
```

```
RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1
```

```
RP/0/RP0/CPU0:router (config-if)# l2transport
RP/0/RP0/CPU0:router (config-if-l2)# cell-packing 6 1

RP/0/RP0/CPU0:router(config-if-l2)# commit
```

The following example shows how to create and configure an AC on a Layer 2 subinterface with a PVC:

```
RP/0/RP0/CPU0:router# configure

RP/0/RP0/CPU0:router(config)# interface atm 0/1/0/0.230 l2transport

RP/0/RP0/CPU0:router(config-if)# pvc 15/230

RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# encapsulation aal0

RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# cell-packing 5 2

RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# shape cbr 622000

RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# commit

RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)#

RP/0/RP0/CPU0:router(config-if)# exit

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

RP/0/RP0/CPU0:router# show atm pvc

                                              Peak  Avg/Min  Burst
Interface          VPI  VCI Type  Encaps  SC   Kbps   Kbps    Cells  Sts
ATM0/1/0/0.230      15  230 PVC   AAL0   UBR  622000    N/A      N/A   UP
ATM0/1/0/3.19       17   19 PVC   SNAP   UBR  622000    N/A      N/A   UP
```

The following example shows how to create and configure an AC on an ATM subinterface with a PVP:

```
RP/0/RP0/CPU0:router# configure

RP/0/RP0/CPU0:router(config)# interface atm 0/6/0/1.10 l2transport

RP/0/RP0/CPU0:router(config-if)# pvp 100

RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# cell-packing 5 2

RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# shape ubr 155000

RP/0/RP0/CPU0:router(config-atm-l2transport-pvp)# commit
```

```
RP/0/RP0/CPU0:router# show atm pvp interface atm 0/6/0/1

                                   Peak  Avg/Min   Burst
Interface          VPI  SC         Kbps     Kbps   Cells      Sts
ATM0/6/0/1.10      100  UBR       155000    N/A      N/A        UP
```

VC-Class Creation and Configuration: Example

The following example shows how to configure a vc-class:

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# vc-class atm atm-class-1
RP/0/RP0/CPU0:router(config-vc-class-atm)# encapsulation aal5snap
RP/0/RP0/CPU0:router(config-vc-class-atm)# oam ais-rdi 25 5
RP/0/RP0/CPU0:router(config-vc-class-atm)# oam retry 5 10 5
RP/0/RP0/CPU0:router(config-vc-class-atm)# oam-pvc manage 300
RP/0/RP0/CPU0:router(config-vc-class-atm)# shape cbr 100000

RP/0/RP0/CPU0:router(config-vc-class-atm)# commit
```

The following example shows how to attach a vc-class to an ATM main interface:

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface ATM0/2/0/0.1 point-to-point
RP/0/RP0/CPU0:router (config-if)# class-int atm-class-1
RP/0/RP0/CPU0:router (config-if)# commit
```

The following example shows how to attach a vc-class to an ATM subinterface:

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface ATM0/2/0/0.1 point-to-point
RP/0/RP0/CPU0:router(config-if)# pvc 10/100
RP/0/RP0/CPU0:router (config-atm-vc)# class-vc atm-class-1
RP/0/RP0/CPU0:router (config-atm-vc)# commit
```

The following example shows how to display information about a specific ATM vc-class:

```
RP/0/RP0/CPU0:router # show atm vc-class atm-class-1
ATM vc-class atm-class-1

  encapsulation          -    aal5snap
  shape                  -    cbr 100000
  oam ais-rdi            -    not configured
  oam retry              -    not configured
  oam-pvc                -    manage 300
```

The following example shows how to display configuration information for the parameters on a virtual circuit (VC) class that is associated with a particular PVC:

```
RP/0/RP0/CPU0:router # show atm class-link 10/100

Detailed display of VC(s) with VPI/VCI = 10/100
```

```
Class link for VC 10/100
ATM0/2/0/0.1: VPI: 10 VCI: 100
shape : cbr 100000 (VC-class configured on VC)
encapsulation : aal5snap (VC-class configured on VC)
oam-pvc : manage 300   (VC-class configured on VC)
oam retry : 3 5 1 (Default value)
oam ais-rdi : 1 3 (Default value)
```

ATM Layer 2 QoS Configuration: Examples

The following examples show how to configure QoS for ATM. For complete information on configuring QoS and QoS commands, refer to the Cisco XR 12000 Series Router Modular Quality of Service Configuration Guide and the Cisco XR 12000 Series Router Modular Quality of Service Command Reference.

Attaching a Service-Policy to an Attachment Circuit Configuration: Example

PVC Mode

```
config
interface ATM 0/1/0/0.2 l2transport
pvc 10/2
service-policy input | output atm_policy_o
```

PVP Mode

```
config
interface ATM 0/1/0/0.3 l2transport
pvp 30
service-policy input atm_policy_i
```

Port Mode

```
config
   interface ATM 0/1/0/0
      l2transport
         service-policy input atm_policy_i
```

Main Interface (non-port mode)

```
config
  interface ATM 0/1/0/0
      service-policy input | output atm_policy_o
```

Policy Map Configuration for CBR/UBR: Example

For CBR.1 (real-time traffic) and UBR (best effort, non-real time traffic) you must specify the PCR and delay tolerance parameters for policing. The main difference between the configurations for UBR.1 and UBR.2 traffic is that for UBR.2 traffic, the exceed action includes the **set-clp-transmit** option to tag non-conforming cells. The police rate can also be expressed as a percentage.

The following example shows how to configure a QoS policy map for CBR/UBR:

```
policy-map CBR1
```

```
    class class-default
         police rate pcr cellsps delay-tolerance cdvt us
               conform-action action
              exceed-action action
```

Policy Map Configuration for VBR.1: Example

For VBR.1 real-time and non-real time traffic you must specify the PCR, SCR, and delay tolerance parameters for for policing. The **atm-mbs** parameter can be specified to define the burst allowed on the SCR bucket. The police rates can also be expressed as percentages. Class atm_clp1 is allowed with police actions.

The following example shows how to configure a QoS policy map for VBR.1:

```
policy-map VBR1
    class class-default
          police rate scr cellsps atm-mbs mbs cells peak-rate pcr cellsps delay-tolerance
cdvt us
               conform-action action
               exceed-action action
```

Policy Map Configuration for VBR.2 and VBR.3: Example

For VBR.2 and VBR.3 real-time and non-real time traffic you must specify the PCR, SCR, and delay tolerance parameters for policing. The **atm-mbs** parameter can be specified to define the burst allowed on the SCR bucket. The main difference between VBR.1 and VBR.2/VBR.3 is that the SCR bucket is for CLP0 cells only. The police rates can be expressed as percentages. The child policy can have other set actions and can match on ATM CLP1.

The following example shows how to configure a hierarchical policy for VBR.2:

```
policy-map child
    class atm_clp0
          police rate scr cellsps atm-mbs mbs cells
              conform-action action
             exceed-action action

policy-map VBR2
    class class-default
          police rate pcr cellsps delay-tolerance cdvt us
               conform-action action
               exceed-action action
      service-policy child
```

Policy Map Configuration to Exclude OAM Cells: Example

OAM cells can be excluded from being policed by configuring the classification criteria. Since **match not** is not supported, the different classes must be explicitly configured:

The following example shows how to configure a policy map to exclude OAM cells:

```
class-map clp-0-1
 match clp 0
 match clp 1

policy-map child
    class atm-oam
        set
    class class-default
```

```
          police rate scr cellsps atm-mbs mbs cells
               conform-action action
               exceed-action action

policy-map VBR2
    class clp-0-1
          police rate pcr cellsps delay-tolerance cdvt us
               conform-action action
               exceed-action action
      service-policy child
```

Policy Map Configuration for Dual Queue Limit: Example

Dual Queue limit configuration is supported on egress L2 ATM interfaces to differentiate between CLP0 and CLP1 cells.

Note For dual queue, only output service policies are supported. Input service policies are not supported.

The following example shows how to configure a policy map for Dual Queue Limit:

```
policy-map q-limit
    class class-default
        queue-limit atm-clp Threshold {[ms|us|cells]} Tail-drop-threshold {[ms|us|cells]}
```

Verifying ATM Layer 2 QoS Configuration: Examples

The following examples show how to display policing results for an ATM interface policy map:

```
show policy-map interface ATM 0/3/0/0.12 input

ATM 0/3/0/0.12 input: pvc1

Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                 0/0               0
    Transmitted         :                 0/0               0
    Total Dropped       :                 0/0               0

show policy-map interface ATM 0/3/0/0.12 output

ATM 0/3/0/0.12 output: pvc1

Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                 0/0               0
    Transmitted         :                 0/0               0
    Total Dropped       :                 0/0               0
```

The following examples show how to display the configured QoS properties for an ATM interface policy map:

```
show qos interface atm 0/3/0/0.12 input

Interface ATM0_3_0_0.12  --   Direction: input
Policy                  :   pvc1
Total number of classes:   1
```

```
Cell Packing Criteria = CELL_PACK_TIMER_MTU
---------------------------------------------------
LEVEL1 class: classid    =   0x1
class name               =   class-default
new exp                  =   6
```

show qos interface atm 0/3/0/0.12 output

```
Interface ATM0_3_0_0.12  --   Direction: output
Policy                 :   pvc1
Total number of classes:   1
Cell Packing Criteria = CELL_PACK_TIMER_MTU
---------------------------------------------------
LEVEL1 class: classid    =   0x1
class name               =   class-default
new exp                  =   6
```

CHAPTER 4

Advanced Configuration and Modification of the Management Ethernet Interface

This module describes the configuration of Management Ethernet interfaces.

Before you can use Telnet to access the router through the LAN IP address, you must set up a Management Ethernet interface and enable Telnet servers, as described in the *Configuring General Router Features* module of the . This module describes how to modify the default configuration of the Management Ethernet interface after it has been configured, as described in *Getting Started Guide.*

Note Forwarding between physical layer interface modules (PLIM) ports and Management Ethernet interface ports is disabled by default. To enable forwarding between PLIM ports and Management Ethernet interface ports, use the **rp mgmtethernet forwarding** command.

Note Although the Management Ethernet interfaces on the system are present by default, the user must configure these interfaces to use them for accessing the router, using protocols and applications such as Simple Network Management Protocol (SNMP), Common Object Request Broker Architecture (CORBA), HTTP, extensible markup language (XML), TFTP, Telnet, and command-line interface (CLI).

Feature History for Configuring Management Ethernet Interfaces

Release	Modification
Release 2.0	This feature was introduced on theCisco CRS-1 Router.
Release 3.2	This feature was first supported on the.Cisco XR 12000 Series Router
Release 3.3.0	Manual configuration of the Management Ethernet interface is the only option. The initial prompts that originally walked the user through Management Ethernet interface configuration upon software installation were removed.

Advanced Configuration and Modification of the Management Ethernet Interface

This module describes the configuration of Management Ethernet interfaces.

Before you can use Telnet to access the router through the LAN IP address, you must set up a Management Ethernet interface and enable Telnet servers, as described in the *Configuring General Router Features* module of the . This module describes how to modify the default configuration of the Management Ethernet interface after it has been configured, as described in *Getting Started Guide.*

Note Forwarding between physical layer interface modules (PLIM) ports and Management Ethernet interface ports is disabled by default. To enable forwarding between PLIM ports and Management Ethernet interface ports, use the **rp mgmtethernet forwarding** command.

Note Although the Management Ethernet interfaces on the system are present by default, the user must configure these interfaces to use them for accessing the router, using protocols and applications such as Simple Network Management Protocol (SNMP), Common Object Request Broker Architecture (CORBA), HTTP, extensible markup language (XML), TFTP, Telnet, and command-line interface (CLI).

Feature History for Configuring Management Ethernet Interfaces

Release	Modification
Release 2.0	This feature was introduced on theCisco CRS-1 Router.
Release 3.2	This feature was first supported on the.Cisco XR 12000 Series Router
Release 3.3.0	Manual configuration of the Management Ethernet interface is the only option. The initial prompts that originally walked the user through Management Ethernet interface configuration upon software installation were removed.

Prerequisites for Configuring Management Ethernet Interfaces

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 performing the Management Ethernet interface configuration procedures that are described in this module, be sure that the following tasks and conditions are met:

- You have performed the initial configuration of the Management Ethernet interface, as described in the *Configuring General Router Features* module of *Cisco IOS XR Getting Started Guide*.
- To use the **show running-config** command, you must be in a user group associated with a task group that includes the proper task IDs for configuration management commands. The Task ID for the **show running-config** command is listed in Cisco IOS XR System Management Command Reference.
- You know how to apply the generalized interface name specification *rack/slot/module/port*.

  For further information on interface naming conventions, refer to *Cisco IOS XR Getting Started Guide*.

Note Note that, for transparent switchover, both active and standby Management Ethernet interfaces are expected to be physically connected to the same LAN or switch.

Information About Configuring Management Ethernet Interfaces

To configure Management Ethernet interfaces, you must understand the following concept:

Default Interface Settings

This table describes the default Management Ethernet interface settings that can be changed by manual configuration. Default settings are not displayed in the **show running-config** command output.

Table 3: Management Ethernet Interface Default Settings

Parameter	Default Value	Configuration File Entry
Speed in Mbps	Speed is autonegotiated.	**speed** [**10** | **100** | **1000**] To return the system to autonegotiate speed, use the **no speed** [**10** | **100** | **1000**] command.
Duplex mode	Duplex mode is autonegotiated.	**duplex** {**full** | **half**} To return the system to autonegotiated duplex operation, use the **no duplex {full | half}** command, as appropriate.

Parameter	Default Value	Configuration File Entry
MAC address	MAC address is read from the hardware burned-in address (BIA).	**mac-address** address To return the device to its default MAC address, use the **no mac-address** address command.

How to Perform Advanced Management Ethernet Interface Configuration

This section contains the following procedures:

Configuring a Management Ethernet Interface

Perform this task to configure a Management Ethernet interface. This procedure provides the minimal configuration required for the Management Ethernet interface.

Note You do not need to perform this task if you have already set up the Management Ethernet interface to enable telnet servers, as described in the module of the *Getting Started Guide*.

SUMMARY STEPS

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **ipv4 address** *ip-address mask*
4. **mtu** *bytes*
5. **no shutdown**
6. **end** or **commit**
7. **show interfaces MgmtEth** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 2	**interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack*/*slot*/*module*/*port*. The example indicates port 0 on the RP card that is installed in slot 0.

	Command or Action	Purpose
Step 3	**ipv4 address** *ip-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224`	Assigns an IP address and subnet mask to the interface. • Replace *ip-address* with the primary IPv4 address for the interface. • Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways: • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.
Step 4	**mtu** *bytes* **Example:** `RP/0/RP0/CPU0:router(config-if# mtu 1448`	(Optional) Sets the maximum transmission unit (MTU) byte value for the interface. The default is 1514. • The default is 1514 bytes. • The range for the Management Ethernet interface Interface **mtu** values is 64 to 1514 bytes.
Step 5	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration, which removes the forced administrative down on the interface, enabling it to move to an up or down state.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**show interfaces MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/RP0/CPU0/0`	(Optional) Displays statistics for interfaces on the router.

Configuring the Duplex Mode for a Management Ethernet Interface

Perform this task to configure the duplex mode of the Management Ethernet interfaces for the RPs.

SUMMARY STEPS

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **duplex** [**full** | **half**]
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0`	Enters interface configuration mode and specifies the Management Ethernet interface name and instance.
Step 3	**duplex** [**full** | **half**] **Example:** `RP/0/RP0/CPU0:router(config-if)# duplex full`	Configures the interface duplex mode. Valid options are **full** or **half**. **Note** • To return the system to autonegotiated duplex operation, use the **no duplex** command.
Step 4	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose
	or `RP/0/RP0/CPU0:router(config-if)# commit`	`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 Speed for a Management Ethernet Interface

Perform this task to configure the speed of the Management Ethernet interfaces for the RPs.

SUMMARY STEPS

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **speed** {**10** | **100** | **1000**}
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0`	Enters interface configuration mode and specifies the Management Ethernet interface name and instance.
Step 3	**speed** {**10** | **100** | **1000**} **Example:**	Configures the interface speed parameter. Valid **speed** options are 10, **100** or 1000 Mbps.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# speed 100`	**Note** • The default Management Ethernet interface speed is autonegotiated. • To return the system to the default autonegotiated speed, use the **no speed** command.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Modifying the MAC Address for a Management Ethernet Interface

Perform this task to configure the MAC layer address of the Management Ethernet interfaces for the RPs.

SUMMARY STEPS

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **mac-address** *address*
4. **end** or **commit**

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0`	Enters interface configuration mode and specifies the Management Ethernet interface name and instance.
Step 3	**mac-address** *address* **Example:** `RP/0/RP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD`	Configures the MAC layer address of the Management Ethernet interface. **Note** • To return the device to its default MAC address, use the **no mac-address** address command.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying Management Ethernet Interface Configuration

Perform this task to verify configuration modifications on the Management Ethernet interfaces for the RPs.

SUMMARY STEPS

1. **show interfaces MgmtEth** *interface-path-id*
2. **show running-config interface MgmtEth** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show interfaces MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/RP0/CPU0/0`	Displays the Management Ethernet interface configuration.
Step 2	**show running-config interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show running-config interface MgmtEth 0/RP0/CPU0/0`	Displays the running configuration.

Configuration Examples for Management Ethernet Interfaces

This section provides the following configuration examples:

Configuring a Management Ethernet Interface: Example

This example displays advanced configuration and verification of the Management Ethernet interface on the RP:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0//CPU0/0
RP/0/RP0/CPU0:router(config)# ipv4 address 172.29.52.70 255.255.255.0
RP/0/RP0/CPU0:router(config-if)# speed 100
RP/0/RP0/CPU0:router(config-if)# duplex full
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:Mar 26 01:09:28.685 :ifmgr[190]:%LINK-3-UPDOWN :Interface MgmtEth0/RP0/CPU0/0,
 changed state to Up
RP/0/RP0/CPU0:router(config-if)# end

RP/0/RP0/CPU0:router# show interfaces MgmtEth 0//CPU0/0

MMgmtEth0/RP0/CPU0/0 is up, line protocol is up
  Hardware is Management Ethernet, address is 0011.93ef.e8ea (bia 0011.93ef.e8ea
)
  Description: Connected to Lab LAN
  Internet address is 172.29.52.70/24
  MTU 1514 bytes, BW 100000 Kbit
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA,  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 3000 bits/sec, 7 packets/sec
  5 minute output rate 0 bits/sec, 1 packets/sec
     30445 packets input, 1839328 bytes, 64 total input drops
     0 drops for unrecognized upper-level protocol
     Received 23564 broadcast packets, 0 multicast packets
```

```
         0 runts, 0 giants, 0 throttles, 0 parity
     57 input errors, 40 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     171672 packets output, 8029024 bytes, 0 total output drops
     Output 16 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions

RP/0/RP0/CPU0:router# show running-config interface MgmtEth 0//CPU0/0

interface MgmtEth0/RP0/CPU0/0
 description Connected to Lab LAN
 ipv4 address 172.29.52.70 255.255.255.0
!
```

CHAPTER 5

Configuring Ethernet Interfaces

This module describes the configuration of Ethernet interfaces.

The distributed Gigabit Ethernet, 10-Gigabit, 40-Gigabit, 100-Gigabit Ethernet, and Fast Ethernet architecture and features deliver network scalability and performance, while enabling service providers to offer high-density, high-bandwidth networking solutions designed to interconnect the router with other systems in POPs, including core and edge routers and Layer 2 switches.

Feature History for Configuring Ethernet Interfaces

Release	Modification
Release 3.0	This feature was introduced on the Cisco CRS-1 Router.
Release 3.2	Support was added for the Cisco CRS-1 Router for the SIP-800. Support for the 8-Port Gigabit Ethernet SPA was introduced on the Cisco CRS-1 Router.
Release 3.3.0	Support was added for egress MAC accounting on the 8-port 10-Gigabit Ethernet PLIM.
Release 3.4.0	The Layer 2 Virtual Private Network (L2VPN) feature was first supported on Ethernet interfaces. Support was added for the 8-Port 1-Gigabit Ethernet SPA.
Release 3.5.0	Support was added on the Cisco CRS-1 Router for the 1-port 10-Gigabit Ethernet WAN SPA.

Release 4.0.0	Support for the following physical layer interface modules (PLIMs) was added: • 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (14X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (20X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)
Release 4.0.1	Support for the following PLIMs was added: • 1-Port 100-Gigabit Ethernet PLIM (1X100GBE) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP) • 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)
Release 4.2.3	Support for Link Layer Discovery Protocol (LLDP) was added.

Configuring Ethernet Interfaces

This module describes the configuration of Ethernet interfaces.

The distributed Gigabit Ethernet, 10-Gigabit, 40-Gigabit, 100-Gigabit Ethernet, and Fast Ethernet architecture and features deliver network scalability and performance, while enabling service providers to offer high-density, high-bandwidth networking solutions designed to interconnect the router with other systems in POPs, including core and edge routers and Layer 2 switches.

Feature History for Configuring Ethernet Interfaces

Release	Modification
Release 3.0	This feature was introduced on the Cisco CRS-1 Router.

Release 3.2	Support was added for the Cisco CRS-1 Router for the SIP-800. Support for the 8-Port Gigabit Ethernet SPA was introduced on the Cisco CRS-1 Router.
Release 3.3.0	Support was added for egress MAC accounting on the 8-port 10-Gigabit Ethernet PLIM.
Release 3.4.0	The Layer 2 Virtual Private Network (L2VPN) feature was first supported on Ethernet interfaces. Support was added for the 8-Port 1-Gigabit Ethernet SPA.
Release 3.5.0	Support was added on the Cisco CRS-1 Router for the 1-port 10-Gigabit Ethernet WAN SPA.
Release 4.0.0	Support for the following physical layer interface modules (PLIMs) was added: • 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (14X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (20X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)
Release 4.0.1	Support for the following PLIMs was added: • 1-Port 100-Gigabit Ethernet PLIM (1X100GBE) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP) • 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)
Release 4.2.3	Support for Link Layer Discovery Protocol (LLDP) was added.

Prerequisites for Configuring Ethernet Interfaces

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 interfaces, be sure that these tasks and conditions are met:

- Confirm that at least one of these line cards supported on the router is installed:
  - 8-Port Fast Ethernet SPA
  - 4-Port 1-Gigabit Ethernet physical layer interface module (PLIM)
  - 2-Port Gigabit Ethernet SPA
  - 5-Port Gigabit Ethernet SPA
  - 8-Port Gigabit Ethernet SPA
  - 10-Port Gigabit Ethernet SPA
  - 1-Port 10-Gigabit Ethernet SPA
  - 1-Port 10-Gigabit Ethernet WAN SPA
  - 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP)
  - 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)
  - 8-Port 10-Gigabit Ethernet PLIM
  - 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (14X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)
  - 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (20X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)
  - 1-Port 100-Gigabit Ethernet PLIM (1X100GBE) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)
- Know the interface IP address.
- You know how to apply the specify the generalized interface name with the generalized notation *rack/slot/module/port*.
- If you are configuring a 10-Gigabit Ethernet interface on a 10-GE DWDM PLIM, you must have configured the DWDM controller.

Information About Configuring Ethernet

Ethernet is defined by the IEEE 802.3 international standard. It enables the connection of up to 1024 nodes over coaxial, twisted-pair, or fiber-optic cable.

The Cisco CRS-1 Router supports Gigabit Ethernet (1000 Mbps), 10-Gigabit Ethernet (10 Gbps), and 100-Gigabit Ethernet (100 Gbps) interfaces.

This section provides the following information sections:

Default Configuration Values for Gigabit Ethernet and 10-Gigabit Ethernet

This table describes the default interface configuration parameters that are present when an interface is enabled on a Gigabit Ethernet or 10-Gigabit Ethernet modular services card and its associated PLIM.

Note You must use the **shutdown** command to bring an interface administratively down. The interface default is **no shutdown**. When a modular services card is first inserted into the router, if there is no established preconfiguration for it, the configuration manager adds a shutdown item to its configuration. This shutdown can be removed only be entering the **no shutdown** command.

Table 4: Gigabit Ethernet and 10-Gigabit Ethernet Modular Services Card Default Configuration Values

Parameter	Configuration File Entry	Default Value
MAC accounting	**mac-accounting**	**off**
Flow control	**flow-control**	egress on ingress off
MTU	**mtu**	• 1514 bytes for normal frames • 1518 bytes for 802.1Q tagged frames. • 1522 bytes for Q-in-Q frames.
MAC address	**mac address**	Hardware burned-in address (BIA)

Default Configuration Values for Fast Ethernet

This table describes the default interface configuration parameters that are present when an interface is enabled on the Fast Ethernet SPA card and its associated PLIM.

Note You must specifically configure the **shutdown** command to bring an interface administratively down. The interface default is **no shutdown**. When a modular services card is first inserted into the router, if there is no established preconfiguration for it, the configuration manager adds a shutdown item to its configuration. This shutdown can be removed only be entering the **no shutdown** command.

Table 5: Fast Ethernet Default Configuration Values

Parameter	Configuration File Entry	Default Value
MAC accounting	**mac-accounting**	**off**
Duplex operation	**duplex full** **duplex half**	Auto-negotiates duplex operation
MTU	**mtu**	1500 bytes
Interface speed	**speed**	100 Mbps
Auto-negotiation	**negotiation auto**	**disable**

Layer 2 VPN on Ethernet Interfaces

Layer 2 Virtual Private Network (L2VPN) connections emulate the behavior of a LAN across an L2 switched, IP or MPLS-enabled IP network, allowing Ethernet devices to communicate with each other as if they were connected to a common LAN segment.

The L2VPN feature enables service providers (SPs) to provide Layer 2 services to geographically disparate customer sites. Typically, an SP uses an access network to connect the customer to the core network. This access network is may use a mixture of Layer 2 technologies, such as Ethernet, ATM and Frame Relay. The connection between the customer site and the nearby SP edge router is known as an Attachment Circuit (AC).

Traffic from the customer travels over this link to the edge of the SP core network. The traffic then tunnels through an L2VPN over the SP core network to another edge router. The edge router sends the traffic down another attachment circuit (AC) to the customer's remote site.

The L2VPN feature enables users to implement different types of end-to-end services.

Cisco IOS XR Software supports a point-to-point end-to-end service, where two Ethernet circuits are connected together. An L2VPN Ethernet port can operate in one of two modes:

- Port Mode—In this mode, all packets reaching the port are sent over the PW (pseudowire), regardless of any VLAN tags that are present on the packets. In VLAN mode, the configuration is performed under the l2transport configuration mode.
- VLAN Mode—Each VLAN on a CE (customer edge) or access network to PE (provider edge) link can be configured as a separate L2VPN connection (using either VC type 4 or VC type 5). In VLAN mode, the configuration is performed under the individual subinterface.

Note The system sets a limit of 24K single vlan tags per NP and a 64K LC limit on the following line cards:

- A9K-MOD400-SE
- A9K-MOD400-CM
- A9K-MOD200-SE/CM
- Cisco ASR 9000 Series 24-port and 48-port dual-rate 10GE/1GE SE/CM
- A9K-8x100 SE/CM
- A99-8x100 SE/CM

Switching can take place in three ways:

- AC-to-PW—Traffic reaching the PE is tunneled over a PW (and conversely, traffic arriving over the PW is sent out over the AC). This is the most common scenario.
- Local switching—Traffic arriving on one AC is immediately sent out of another AC without passing through a pseudowire.
- PW stitching—Traffic arriving on a PW is not sent to an AC, but is sent back into the core over another PW.

Keep the following in mind when configuring L2VPN on an Ethernet interface:

- L2VPN links support QoS (Quality of Service) and MTU (maximum transmission unit) configuration.
- If your network requires that packets are transported transparently, you may need to modify the packet's destination MAC (Media Access Control) address at the edge of the Service Provider (SP) network. This prevents the packet from being consumed by the devices in the ST network.
- Cisco IOS XR software supports up to 4,000 ACs per line card. Note that not all line cards can support as many as 4,000 ACs. Refer to the specifications of the individual line card for details on the maximum number of ACs supported.

Use the **show interfaces** command to display AC and PW information.

To configure a point-to-point pseudowire xconnect on an AC, see the *Implementing MPLS Layer 2 VPNs* module of the *Cisco IOS XR Multiprotocol Label Switching Configuration Guide.*

To attach Layer 2 service policies, such as QoS, to the Ethernet interface, refer to the appropriate Cisco IOS XR software configuration guide.

Gigabit Ethernet Protocol Standards Overview

The Gigabit Ethernet interfaces support the following protocol standards:

These standards are further described in the sections that follow.

IEEE 802.3 Physical Ethernet Infrastructure

The IEEE 802.3 protocol standards define the physical layer and MAC sublayer of the data link layer of wired Ethernet. IEEE 802.3 uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access at a variety of speeds over a variety of physical media. The IEEE 802.3 standard covers 10 Mbps Ethernet. Extensions to the IEEE 802.3 standard specify implementations for Gigabit Ethernet, 10-Gigabit Ethernet, and Fast Ethernet.

IEEE 802.3ab 1000BASE-T Gigabit Ethernet

The IEEE 802.3ab protocol standards, or Gigabit Ethernet over copper (also known as 1000BaseT) is an extension of the existing Fast Ethernet standard. It specifies Gigabit Ethernet operation over the Category 5e/6 cabling systems already installed, making it a highly cost-effective solution. As a result, most copper-based environments that run Fast Ethernet can also run Gigabit Ethernet over the existing network infrastructure to dramatically boost network performance for demanding applications.

IEEE 802.3z 1000 Mbps Gigabit Ethernet

Gigabit Ethernet builds on top of the Ethernet protocol, but increases speed tenfold over Fast Ethernet to 1000 Mbps, or 1 Gbps. Gigabit Ethernet allows Ethernet to scale from 10 or 100 Mbps at the desktop to 100 Mbps up to 1000 Mbps in the data center. Gigabit Ethernet conforms to the IEEE 802.3z protocol standard.

By leveraging the current Ethernet standard and the installed base of Ethernet and Fast Ethernet switches and routers, network managers do not need to retrain and relearn a new technology in order to provide support for Gigabit Ethernet.

IEEE 802.3ae 10 Gbps Ethernet

Under the International Standards Organization's Open Systems Interconnection (OSI) model, Ethernet is fundamentally a Layer 2 protocol. 10-Gigabit Ethernet uses the IEEE 802.3 Ethernet MAC protocol, the IEEE

802.3 Ethernet frame format, and the minimum and maximum IEEE 802.3 frame size. 10 Gbps Ethernet conforms to the IEEE 802.3ae protocol standards.

Just as 1000BASE-X and 1000BASE-T (Gigabit Ethernet) remained true to the Ethernet model, 10-Gigabit Ethernet continues the natural evolution of Ethernet in speed and distance. Because it is a full-duplex only and fiber-only technology, it does not need the carrier-sensing multiple-access with the CSMA/CD protocol that defines slower, half-duplex Ethernet technologies. In every other respect, 10-Gigabit Ethernet remains true to the original Ethernet model.

IEEE 802.3ba 100 Gbps Ethernet

IEEE 802.3ba is supported on the Cisco 1-Port 100-Gigabit Ethernet PLIM beginning in Cisco IOS XR 4.0.1.

MAC Address

A MAC address is a unique 6-byte address that identifies the interface at Layer 2.

MAC Accounting

The MAC address accounting feature provides accounting information for IP traffic based on the source and destination MAC addresses on LAN interfaces. This feature calculates the total packet and byte counts for a LAN interface that receives or sends IP packets to or from a unique MAC address. It also records a time stamp for the last packet received or sent.

These statistics are used for traffic monitoring, debugging and billing. For example, with this feature you can determine the volume of traffic that is being sent to and/or received from various peers at NAPS/peering points. This feature is currently supported on Ethernet, FastEthernet, and bundle interfaces and supports Cisco Express Forwarding (CEF), distributed CEF (dCEF), flow, and optimum switching.

Note A maximum of 512 MAC addresses per trunk interface are supported for MAC address accounting.

Ethernet MTU

The Ethernet maximum transmission unit (MTU) is the size of the largest frame, minus the 4-byte frame check sequence (FCS), that can be transmitted on the Ethernet network. Every physical network along the destination of a packet can have a different MTU.

Cisco IOS XR software supports two types of frame forwarding processes:

- Fragmentation for IPV4 packets–In this process, IPv4 packets are fragmented as necessary to fit within the MTU of the next-hop physical network.

  Note IPv6 does not support fragmentation.

- MTU discovery process determines largest packet size–This process is available for all IPV6 devices, and for originating IPv4 devices. In this process, the originating IP device determines the size of the largest IPv6 or IPV4 packet that can be sent without being fragmented. The largest packet is equal to the smallest MTU of any network between the IP source and the IP destination devices. If a packet is larger than the smallest MTU of all the networks in its path, that packet will be fragmented as necessary. This process ensures that the originating device does not send an IP packet that is too large.

Jumbo frame support is automatically enable for frames that exceed the standard frame size. The default value is 1514 for standard frames and 1518 for 802.1Q tagged frames. These numbers exclude the 4-byte frame check sequence (FCS).

Flow Control on Ethernet Interfaces

The flow control used on 10-Gigabit Ethernet interfaces consists of periodically sending flow control pause frames. It is fundamentally different from the usual full- and half-duplex flow control used on standard management interfaces. Flow control can be activated or deactivated for ingress traffic only. It is automatically implemented for egress traffic.

802.1Q VLAN

A VLAN is a group of devices on one or more LANs that are configured so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, it is very flexible for user and host management, bandwidth allocation, and resource optimization.

The IEEE's 802.1Q protocol standard addresses the problem of breaking large networks into smaller parts so broadcast and multicast traffic does not consume more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.

VRRP

The Virtual Router Redundancy Protocol (VRRP) eliminates the single point of failure inherent in the static default routed environment. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VPN concentrators on a LAN. The VRRP VPN concentrator controlling the IP addresses associated with a virtual router is called the master, and forwards packets sent to those IP addresses. When the master becomes unavailable, a backup VPN concentrator takes the place of the master.

For more information on VRRP, see the *Implementing VRRP on Cisco IOS XR Software* module of .

HSRP

Hot Standby Routing Protocol (HSRP) is a proprietary protocol from Cisco. HSRP is a routing protocol that provides backup to a router in the event of failure. Several routers are connected to the same segment of an Ethernet, FDDI, or token-ring network and work together to present the appearance of a single virtual router on the LAN. The routers share the same IP and MAC addresses and therefore, in the event of failure of one router, the hosts on the LAN are able to continue forwarding packets to a consistent IP and MAC address. The transfer of routing responsibilities from one device to another is transparent to the user.

HSRP is designed to support non disruptive switchover of IP traffic in certain circumstances and to allow hosts to appear to use a single router and to maintain connectivity even if the actual first hop router they are using fails. In other words, HSRP protects against the failure of the first hop router when the source host cannot learn the IP address of the first hop router dynamically. Multiple routers participate in HSRP and in concert create the illusion of a single virtual router. HSRP ensures that one and only one of the routers is forwarding packets on behalf of the virtual router. End hosts forward their packets to the virtual router.

The router forwarding packets is known as the *active router*. A standby router is selected to replace the active router should it fail. HSRP provides a mechanism for determining active and standby routers, using the IP addresses on the participating routers. If an active router fails a standby router can take over without a major interruption in the host's connectivity.

HSRP runs on top of User Datagram Protocol (UDP), and uses port number 1985. Routers use their actual IP address as the source address for protocol packets, not the virtual IP address, so that the HSRP routers can identify each other.

For more information on HSRP, see the *Implementing HSRP on Cisco IOS XR Software* module of *IP Addresses and Services Configuration Guide.*

Duplex Mode on Fast Ethernet Interfaces

Fast Ethernet ports support the duplex transmission type. Full-duplex mode enables the simultaneous data transmission between a sending station and a receiving station, while half-duplex mode enables data transmission in only one direction at a time.

When configuring duplex mode on a Fast Ethernet interface, keep the following in mind:

- If auto-negotiation is enabled on the interface, the default is duplex negotiated.
- If auto-negotiation is disabled on the interface, the default is full-duplex.

Note You can configure duplex mode on Fast Ethernet interfaces only. Gigabit Ethernet and 10-Gigabit Ethernet interfaces always run in full-duplex mode.

Fast Ethernet Interface Speed

You can configure the interface speed on Fast Ethernet interfaces. Keep the following in mind when configuring the speed for a Fast Ethernet interface:

- If auto-negotiation is enabled on an interface, the default is speed negotiated.
- If auto-negotiation is disabled on an interface, the default speed is the maximum speed allowed on the interface.

Note Both ends of a link must have the same interface speed. A manually configured interface speed overrides any auto-negotiated speed, which can prevent a link from coming up if the configured interface speed at one end of a link is different from the interface speed on the other end.

Link Autonegotiation on Ethernet Interfaces

Link autonegotiation ensures that devices that share a link segment are automatically configured with the highest performance mode of interoperation. Use the **negotiation auto** command in interface configuration mode to enable link autonegotiation on an Ethernet interface. On line card Ethernet interfaces, link autonegotiation is disabled by default.

Note The **negotiation auto** command is available on Gigabit Ethernet and Fast Ethernet interfaces only.

This Tbale describes the performance of the system for different combinations of the duplex and speed modes. The specified **duplex** command configured with the specified **speed** command produces the resulting system action, provided that you have configured autonegotiation on the interface.

.

Table 6: Relationship Between duplex and speed Commands

duplex Command	speed Command	
no duplex	**no speed**	Auto-negotiates both speed and duplex modes.
no duplex	**speed 1000**	Auto-negotiates for duplex mode and forces 1000 Mbps.
no duplex	**speed 100**	Auto-negotiates for duplex mode and forces 100 Mbps.
no duplex	**speed 10**	Auto-negotiates for duplex mode and forces 10 Mbps.
full-duplex	**no speed**	Forces full duplex and auto-negotiates for speed.
full-duplex	**speed 1000**	Forces full duplex and 1000 Mbps.
full-duplex	**speed 100**	Forces full duplex and 100 Mbps.
full-duplex	**speed 10**	Forces full duplex and 10 Mbps.
half-duplex	**no speed**	Forces half duplex and auto-negotiates for speed.
half-duplex	**speed 1000**	Forces half duplex and 1000 Mbps.
half-duplex	**speed 100**	Forces half duplex and 100 Mbps.
half-duplex	**speed 10**	Forces half duplex and 10 Mbps.

LLDP

> **Note** LLDP is not supported on the FP-X line cards.

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the Data Link layer) on all Cisco-manufactured devices (routers, bridges, access servers, and switches). CDP allows network management applications to automatically discover and learn about other Cisco devices connected to the network.

To support non-Cisco devices and to allow for interoperability between other devices, the Cisco CRS Router also supports the IEEE 802.1AB LLDP. LLDP is also a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the Data Link Layer, which allows two systems running different network layer protocols to learn about each other.

LLDP supports a set of attributes that it uses to learn information about neighbor devices. These attributes have a defined format known as a Type-Length-Value (TLV). LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

In addition to the mandatory TLVs (Chassis ID, Port ID, and Time-to-Live), the router also supports the following basic management TLVs, which are optional:

- Port Description
- System Name
- System Description
- System Capabilities
- Management Address

These optional TLVs are automatically sent when LLDP is active, but you can disable them as needed using the **lldp tlv-select disable** command.

LLDP Frame Format

LLDP frames use the IEEE 802.3 format, which consists of the following fields:

- Destination address (6 bytes)—Uses a multicast address of 01-80-C2-00-00-0E.
- Source address (6 bytes)—MAC address of the sending device or port.
- LLDP Ethertype (2 bytes)—Uses 88-CC.
- LLDP PDU (1500 bytes)—LLDP payload consisting of TLVs.
- FCS (4 bytes)—Cyclic Redundancy Check (CRC) for error checking.

LLDP TLV Format

LLDP TLVs carry the information about neighboring devices within the LLDP PDU using the following basic format:

- TLV Header (16 bits), which includes the following fields:
  - TLV Type (7 bits)
  - TLV Information String Length (9 bits)
- TLV Information String (0 to 511 bytes)

LLDP Operation

LLDP is a one-way protocol. The basic operation of LLDP consists of a device enabled for transmit of LLDP information sending periodic advertisements of information in LLDP frames to a receiving device.

Devices are identified using a combination of the Chassis ID and Port ID TLVs to create an MSAP (MAC Service Access Point). The receiving device saves the information about a neighbor for a certain amount time specified in the TTL TLV, before aging and removing the information.

LLDP supports the following additional operational characteristics:

- LLDP can operate independently in transmit or receive modes.
- LLDP operates as a slow protocol using only untagged frames, with transmission speeds of less than 5 frames per second.
- LLDP packets are sent when the following occurs:
  - The packet update frequency specified by the **lldp timer** command is reached. The default is 30 seconds.
  - When a change in the values of the managed objects occurs from the local system's LLDP MIB.
  - When LLDP is activated on an interface (3 frames are sent upon activation similar to CDP).
- When an LLDP frame is received, the LLDP remote services and PTOPO MIBs are updated with the information in the TLVs.
- LLDP supports the following actions on these TLV characteristics:
  - Interprets a TTL value of 0 as a request to automatically purge the information of the transmitting device. These shutdown LLDPDUs are typically sent prior to a port becoming inoperable.
  - An LLDP frame with a malformed mandatory TLV is dropped.
  - A TLV with an invalid value is ignored.
  - A copy of an unknown organizationally-specific TLV is maintained if the TTL is non-zero, for later access through network management.

Supported LLDP Functions

The Cisco CRS Router supports the following LLDP functions:

- IPv4 and IPv6 management addresses—In general, both IPv4 and IPv6 addresses will be advertised if they are available, and preference is given to the address that is configured on the transmitting interface.

If the transmitting interface does not have a configured address, then the TLV will be populated with an address from another interface. The advertised LLDP IP address is implemented according to the following priority order of IP addresses for interfaces on the Cisco CRS Router:

- Locally configured address
- MgmtEth0/RP0/CPU0/0
- MgmtEth0/RP0/CPU0/1
- MgmtEth0/RP1/CPU0/0
- MgmtEth0/RP1/CPU0/1
- Loopback interfaces

  There are some differences between IPv4 and IPv6 address management in LLDP:

- For IPv4, as long as the IPv4 address is configured on an interface, it can be used as an LLDP management address.
- For IPv6, after the IPv6 address is configured on an interface, the interface status must be Up and pass the DAD (Duplicate Address Detection) process before it is can be used as an LLDP management address.
- LLDP is supported for the nearest physically attached, non-tunneled neighbors.
- Port ID TLVs are supported for Ethernet interfaces, subinterfaces, bundle interfaces, and bundle subinterfaces.

Unsupported LLDP Functions

The following LLDP functions are not supported on the Cisco CRS Router:

- LLDP-MED organizationally unique extension—However, interoperability still exists between other devices that do support this extension.
- Tunneled neighbors, or neighbors more than one hop away.
- LLDP TLVs cannot be disabled on a per-interface basis; However, certain optional TLVs can be disabled globally.
- LLDP SNMP trap lldpRemTablesChange.

Enabling LLDP Per Interface

When you enable LLDP globally, all interfaces that support LLDP are automatically enabled for both transmit and receive operations. However, if you want to enable LLDP per interface, perform the following configuration steps:

1. `RP/0/RSP0/CPU0:ios(config)# int gigabitEthernet 0/2/0/0`
2. `RP/0/RSP0/CPU0:ios(config-if)# no sh`
3. `RP/0/RSP0/CPU0:ios(config-if)#commit`
4. `RP/0/RSP0/CPU0:ios(config-if)#lldp ?`

5. `RP/0/RSP0/CPU0:ios(config-if)#lldp enable`

6. `RP/0/RSP0/CPU0:ios(config-if)#commit`

Running configuration

```
RP/0/RSP0/CPU0:ios#sh running-config
Wed Jun 27 12:40:21.274 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Wed Jun 27 00:59:29 2018 by UNKNOWN
!
interface GigabitEthernet0/1/0/0
 shutdown
!
interface GigabitEthernet0/1/0/1
 shutdown
!
interface GigabitEthernet0/1/0/2
 shutdown
!
interface GigabitEthernet0/2/0/0
 Shutdown
!
interface GigabitEthernet0/2/0/1
 shutdown
!
interface GigabitEthernet0/2/0/2
 shutdown
!
end
```

Verification

```
Verifying the config
==================
RP/0/RSP0/CPU0:ios#sh lldp interface <===== LLDP enabled only on GigEth0/2/0/0
Wed Jun 27 12:43:26.252 IST


GigabitEthernet0/2/0/0:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
RP/0/RSP0/CPU0:ios#

RP/0/RSP0/CPU0:ios# show lldp neighbors
Wed Jun 27 12:44:38.977 IST
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf          Hold-time  Capability     Port ID
ios            Gi0/2/0/0           120        R              Gi0/2/0/0       <====== LLDP
 enabled only on GigEth0/2/0/0 and neighborship seen for the same.

Total entries displayed: 1

RP/0/RSP0/CPU0:ios#
```

Carrier Delay on Ethernet Interfaces

When enabled on an Ethernet interface, the Carrier Delay feature slows the response of the system to line-up or line-down events. You can configure both Carrier Delay up and Carrier Delay down on an interface at the same time.

Carrier Delay up suppresses short line flaps where the line is down, then comes up, then goes down again. A line that was previously down must be up longer than the duration specified for the **carrier-delay up** command before the system is informed that the interface has come up. All flaps that are shorter than the duration specified for the **carrier-delay up** command are suppressed.

Configuring Carrier Delay up helps to ensure that a line is reasonably stable before the system is informed that the interface is up and ready to forward traffic.

Carrier Delay down suppresses short line flaps where the line is up, then goes down, then comes up again. A line that was previously up must be down longer than the duration specified for the **carrier-delay down** command before the system is informed that the interface has gone down. All flaps that are shorter than the value specified for the **carrier-delay down** command are suppressed.

Configuring Carrier Delay down can be beneficial in suppressing very short link flaps, thereby preventing interface flaps. Alternatively, configuring this feature can be beneficial in allowing other line protection equipment to have enough time to intervene.

How to Configure Ethernet

This section provides the following configuration procedures:

Configuring Ethernet Interfaces

This section provides the following configuration procedures:

Configuring Gigabit Ethernet Interfaces

Use the following procedure to create a basic Gigabit Ethernet, 10-Gigabit Ethernet, or 100-Gigabit Ethernet interface configuration.

SUMMARY STEPS

1. **show version**
2. **show interfaces** [**GigabitEthernet** | **TenGigE** | **HundredGigE**] *interface-path-id*
3. **configure**
4. **interface** [**GigabitEthernet | TenGigE | HundredGigE**] *interface-path-id*
5. **ipv4 address** *ip-address mask*
6. **flow-control** {**bidirectional**| **egress** | **ingress**}
7. **mtu** *bytes*
8. **mac-address** *value1.value2.value3*
9. **negotiation auto**
10. **no shutdown**
11. **end** or **commit**
12. **show interfaces** [**GigabitEthernet | TenGigE**] *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show version** **Example:** `RP/0/RP0/CPU0:router# show version`	(Optional) Displays the current software version, and can also be used to confirm that the router recognizes the modular services card.
Step 2	**show interfaces** [**GigabitEthernet** | **TenGigE** | **HundredGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces TenGigE 0/1/0/0`	(Optional) Displays the configured interface and checks the status of each interface port. Possible interface types for this procedure are: • GigabitEthernet • HundredGigE • TenGigE
Step 3	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 4	**interface** [**GigabitEthernet** | **TenGigE** | **HundredGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. Possible interface types for this procedure are: • GigabitEthernet • HundredGigE • TenGigE **Note** • The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
Step 5	**ipv4 address** *ip-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224`	Assigns an IP address and subnet mask to the interface. • Replace *ip-address* with the primary IPv4 address for the interface. • Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways: • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits

	Command or Action	Purpose
		of the mask are ones, and the corresponding bits of the address are network address.
Step 6	**flow-control** {**bidirectional**| **egress** | **ingress**} **Example:** `RP/0/RP0/CPU0:router(config-if)# flow control ingress`	(Optional) Enables the sending and processing of flow control pause frames. • **egress**—Enables the sending of flow control pause frames in egress. • **ingress**—Enables the processing of received pause frames on ingress. • **bidirectional**—Enables the sending of flow control pause frames in egress and the processing of received pause frames on ingress.
Step 7	**mtu** *bytes* **Example:** `RP/0/RP0/CPU0:router(config-if)# mtu 1448`	(Optional) Sets the MTU value for the interface. • The default is 1514 bytes for normal frames and 1518 bytes for 802.1Q tagged frames. • The range for Gigabit Ethernet and 10-Gigabit Ethernet mtu values is 64 bytes to 65535 bytes.
Step 8	**mac-address** *value1.value2.value3* **Example:** `RP/0/RP0/CPU0:router(config-if)# mac address 0001.2468.ABCD`	(Optional) Sets the MAC layer address of the Management Ethernet interface. • The values are the high, middle, and low 2 bytes, respectively, of the MAC address in hexadecimal. The range of each 2-byte value is 0 to ffff.
Step 9	**negotiation auto** **Example:** `RP/0/0/CPU0:router(config-if)# negotiation auto`	(Optional) Enables autonegotiation on a Gigabit Ethernet interface. • Autonegotiation must be explicitly enabled on both ends of the connection, or speed and duplex settings must be configured manually on both ends of the connection. • If autonegotiation is enabled, any speed or duplex settings that you configure manually take precedence. **Note** • The **negotiation auto** command is available on Gigabit Ethernet and Fast Ethernet interfaces only.
Step 10	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration, which forces an interface administratively down.

	Command or Action	Purpose

Step 11	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 12	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces TenGigE 0/3/0/0`	(Optional) Displays statistics for interfaces on the router.

What to do next

To configure MAC Accounting on the Ethernet interface, see the “Configuring MAC Accounting on an Ethernet Interface” section later in this module.

To configure an AC on the Ethernet port for Layer 2 VPN implementation, see the “Configuring a L2VPN Ethernet Port” section later in this module.

To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or Quality of Service (QoS), to the Ethernet interface, refer to the appropriate Cisco IOS XR software configuration guide.

Configuring a Fast Ethernet Interface

What to do next

- To configure an AC on the Fast Ethernet port for Layer 2 VPN implementation, see the “Configuring a L2VPN Ethernet Port” section later in this module.

- To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or Quality of Service (QoS), to the Fast Ethernet interface, refer to the appropriate Cisco ASR 9000 Series Router or Cisco IOS XR software configuration guide.

Configuring MAC Accounting on an Ethernet Interface

This task explains how to configure MAC accounting on an Ethernet interface. MAC accounting has special show commands, which are illustrated in this procedure. Otherwise, the configuration is the same as configuring a basic Ethernet interface, and the steps can be combined in one configuration session. See "Configuring Gigabit Ethernet Interfaces" in this module for information about configuring the other common parameters for Ethernet interfaces.

SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE** | **fastethernet**] *interface-path-id*
3. **ipv4 address** *ip-address mask*
4. **mac-accounting** {**egress** | **ingress**}
5. **end** or **commit**
6. **show mac-accounting** *type* **location** *instance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** [**GigabitEthernet** | **TenGigE** | **fastethernet**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	Physical interface or virtual interface. **Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (**?**) online help function.
Step 3	**ipv4 address** *ip-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224`	Assigns an IP address and subnet mask to the interface. • Replace *ip-address* with the primary IPv4 address for the interface. • Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways: • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.

	Command or Action	Purpose
		• The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.
Step 4	**mac-accounting {egress | ingress}** **Example:** `RP/0/RP0/CPU0:router(config-if)# mac-accounting egress`	Generates accounting information for IP traffic based on the source and destination MAC addresses on LAN interfaces. • To disable MAC accounting, use the **no** form of this command.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 6	**show mac-accounting** *type* **location** *instance* **Example:** `RP/0/RP0/CPU0:router# show mac-accounting TenGigE location 0/2/0/4`	Displays MAC accounting statistics for an interface.

Configuring a L2VPN Ethernet Port

Use the following procedure to configure an L2VPN Ethernet port.

Note The steps in this procedure configure the L2VPN Ethernet port to operate in port mode.

To configure a point-to-point pseudowire xconnect on an AC, see the Implementing MPLS Layer 2 VPNs module of the Cisco IOS XR L2VPN and Ethernet Services Configuration Guide for the Cisco CRS Router.

To attach Layer 2 service policies, such as quality of service (QoS), to the Ethernet interface, refer to the appropriate Cisco IOS XR software configuration guide.

SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **l2transport**
4. **l2protocol cpsv** {**tunnel** | **reverse-tunnel**}
5. **l2protocol** {**cdp** | **pvst** | **stp** | **vtp**}{[**forward** | **tunnel**][**experimental** *bits*]|**drop**}
6. **end** or **commit**
7. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. Possible interface types for this procedure are: • GigabitEthernet • TenGigE
Step 3	**l2transport** **Example:** `RP/0/RP0/CPU0:router(config-if)# l2transport`	Enables Layer 2 transport mode on a port and enter Layer 2 transport configuration mode.
Step 4	**l2protocol cpsv** {**tunnel** | **reverse-tunnel**} **Example:** `RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol cpsv tunnel`	Configures Layer 2 protocol tunneling and protocol data unit (PDU) filtering on an Ethernet interface. • **tunnel**—Specifies L2PT encapsulation on frames as they enter the interface, and de-encapsulation on frames as they exit they interface. • **reverse-tunnel**—Specifies L2PT encapsulation on frames as they exit the interface, and de-encapsulation on frames as they enter the interface.
Step 5	**l2protocol** {**cdp** | **pvst** | **stp** | **vtp**}{[**forward** | **tunnel**][**experimental** *bits*]|**drop**} **Example:**	Configures Layer 2 protocol tunneling and protocol data unit (PDU) filtering on an Ethernet interface for one of the following protocols: CDP, PVST+, STP, VTP, where:

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if-l2)# l2protocol stp tunnel`	Possible protocols and options are: • **cdp**—Cisco Discovery Protocol (CDP) tunneling and data unit parameters. • **pvst**—Configures VLAN spanning tree protocol tunneling and data unit parameters. • **stp**—spanning tree protocol tunneling and data unit parameters. • **vtp**—VLAN trunk protocol tunneling and data unit parameters. • **tunnel**—(Optional) Tunnels the packets associated with the specified protocol. • **experimental** *bits*—(Optional) Modifies the MPLS experimental bits for the specified protocol. • **drop**—(Optional) Drop packets associated with the specified protocol.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if-l2)# end`
	or
	`RP/0/RP0/CPU0:router(config-if-l2)# 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.

Step 7	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces TenGigE 0/3/0/0`	(Optional) Displays statistics for interfaces on the router.

To configure a point-to-point pseudowire xconnect on an AC, refer to these documents:

- Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide
- Cisco ASR 9000 Series Aggregation Services Router VPN and Ethernet Services Command Reference

Configuring LLDP

Note LLDP is not supported on the FP-X line cards.

This section includes the following configuration topics for LLDP:

LLDP Default Configuration

This table shows the values of the LLDP default configuration on the Cisco CRS Router. To change the default settings, use the LLDP global configuration and LLDP interface configuration commands.

LLDP Function	Default
LLDP global state	Disabled
LLDP holdtime (before discarding)	120 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP reinitialization delay	2 seconds
LLDP TLV selection	All TLVs are enabled for sending and receiving.
LLDP interface state	Enabled for both transmit and receive operation when LLDP is globally enabled.

Enabling LLDP Globally

To run LLDP on the router, you must enable it globally. When you enable LLDP globally, all interfaces that support LLDP are automatically enabled for both transmit and receive operations.

You can override this default operation at the interface to disable receive or transmit operations. For more information about how to selectively disable LLDP receive or transmit operations for an interface, see the Disabling LLDP Receive and Transmit Operation for an Interface.

To enable LLDP globally, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **lldp**
3. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**lldp** **Example:** `RP/0/RP0/CPU0:router(config)#lldp`	Enables LLDP globally for both transmit and receive operation on the system.

Step 3	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# 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 Global LLDP Operational Characteristics

The LLDP Default Configuration describes the default operational characteristics for LLDP. When you enable LLDP globally on the router using the **lldp** command, these defaults are used for the protocol.

To modify the global LLDP operational characteristics such as the LLDP neighbor information holdtime, initialization delay, or packet rate, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **lldp holdtime** *seconds*
3. **lldp reinit** *seconds*
4. **lldp timer** *seconds*

5. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**lldp holdtime** *seconds* **Example:** `RP/0/RP0/CPU0:router(config)#lldp holdtime 60`	(Optional) Specifies the length of time that information from an LLDP packet should be held by the receiving device before aging and removing it.
Step 3	**lldp reinit** *seconds* **Example:** `RP/0/RP0/CPU0:router(config)# lldp reinit 4`	(Optional) Specifies the length of time to delay initialization of LLDP on an interface.
Step 4	**lldp timer** *seconds* **Example:** `RP/0/RP0/CPU0:router(config)#lldp reinit 60`	(Optional) Specifies the LLDP packet rate.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# 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.

Disabling Transmission of Optional LLDP TLVs

Certain TLVs are classified as mandatory in LLDP packets, such as the Chassis ID, Port ID, and Time to Live (TTL) TLVs. These TLVs must be present in every LLDP packet. You can suppress transmission of certain other optional TLVs in LLDP packets.

To disable transmission of optional LLDP TLVs, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **lldp tlv-select** *tlv-name* **disable**
3. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**lldp tlv-select** *tlv-name* **disable** **Example:** `RP/0/RP0/CPU0:router(config)# lldp tlv-select system-capabilities disable`	(Optional) Specifies that transmission of the selected TLV in LLDP packets is disabled. The *tlv-name* can be one of the following LLDP TLV types: • **management-address** • **port-description** • **system-capabilities** • **system-description** • **system-name**

Step 3	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# 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.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Disabling LLDP Receive and Transmit Operation for an Interface

When you enable LLDP globally on the router, all supported interfaces are automatically enabled for LLDP receive and transmit operation. You can override this default by disabling these operations for a particular interface.

To disable LLDP receive and transmit operations for an interface, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **interface GigabitEthernet 0/2/0/0**
3. **lldp**
4. **receive disable**
5. **transmit disable**
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router#` **`configure`**	Enters global configuration mode.
Step 2	**interface GigabitEthernet 0/2/0/0** **Example:** `RP/0/RP0/CPU0:router(config)#`**`interface GigabitEthernet 0/2/0/0`**	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. Possible interface types for this procedure are: • GigabitEthernet • TenGigE
Step 3	**lldp** **Example:** `RP/0/RP0/CPU0:router(config-if)#`**`lldp`**	(Optional) Enters LLDP configuration mode for the specified interface.
Step 4	**receive disable** **Example:**	(Optional) Disables LLDP receive operations on the interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-lldp)#receive disable`	
Step 5	**transmit disable** **Example:** `RP/0/RP0/CPU0:router(config-lldp)#transmit disable`	(Optional) Disables LLDP transmit operations on the interface.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the LLDP Configuration

This section describes how you can verify the LLDP configuration both globally and for a particular interface.

Verifying the LLDP Global Configuration

To verify the LLDP global configuration status and operational characteristics, use the **show lldp** command as shown in the following example:

```
RP/0/RSP0/CPU0:router# show lldp
Wed Apr 13 06:16:45.510 DST
Global LLDP information:
        Status: ACTIVE
        LLDP advertisements are sent every 30 seconds
        LLDP hold time advertised is 120 seconds
        LLDP interface reinitialisation delay is 2 seconds
```

If LLDP is not enabled globally, the following output appears when you run the **show lldp** command:

```
RP/0/RSP0/CPU0:router# show lldp
Wed Apr 13 06:42:48.221 DST
% LLDP is not enabled
```

Verifying the LLDP Interface Configuration

To verify the LLDP interface status and configuration, use the **show lldp interface** command as shown in the following example:

```
RP/0/RSP0/CPU0:router# show lldp interface GigabitEthernet 0/1/0/7
Wed Apr 13 13:22:30.501 DST


GigabitEthernet0/1/0/7:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
```

To monitor and maintain LLDP on the system or get information about LLDP neighbors, use one of the following commands:

	Description
clear lldp	Resets LLDP traffic counters or LLDP neighbor information.
show lldp entry	Displays detailed information about LLDP neighbors.
show lldp errors	Displays LLDP error and overflow statistics.
show lldp neighbors	Displays information about LLDP neighbors.
show lldp traffic	Displays statistics for LLDP traffic.

Configuration Examples for Ethernet

This section provides the following configuration examples:

Configuring an Ethernet Interface: Example

The following example shows how to configure an interface for a 10-Gigabit Ethernet modular services card:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# flow-control ingress
RP/0/RP0/CPU0:router(config-if)# mtu 1448
RP/0/RP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

```
RP/0/RP0/CPU0:router# show interfaces TenGigE 0/0/0/1

TenGigE0/0/0/1 is down, line protocol is down
  Hardware is TenGigE, address is 0001.2468.abcd (bia 0001.81a1.6b23)
  Internet address is 172.18.189.38/27
  MTU 1448 bytes, BW 10000000 Kbit
     reliability 0/255, txload Unknown, rxload Unknown
Encapsulation ARPA,
  Full-duplex, 10000Mb/s, LR
  output flow control is on, input flow control is on
  loopback not set
ARP type ARPA, ARP timeout 01:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Configuring a Fast Ethernet Interface: Example

The following example indicates how to configure an interface for a Fast Ethernet SPA:

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# interface fastethernet 0/0/2/0
RP/0/0/CPU0:router(config-if)# ipv4 address172.30.1.2 255.255.255.224
RP/0/0/CPU0:router(config-if)# duplex full
RP/0/0/CPU0:router(config-if)# mtu 1514
RP/0/0/CPU0:router(config-if)# speed 100
RP/0/0/CPU0:router(config-if)# no shutdown
RP/0/0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/0/CPU0:router# show interfaces fastethernet 0/0/2/0

FastEthernet0/0/2/0 is up, line protocol is up

  Hardware is FastEthernet, address is 000f.f83b.30c8 (bia 000f.f83b.30c8)
  Internet address is 172.30.1.2/24
  MTU 1514 bytes, BW 1000000 Kbit
reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA,
  Duplex unknown, 100Mb/s, TX, link type is force-up
  output flow control is off, input flow control is off
  loopback not set
  ARP type ARPA, ARP timeout 04:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
0 packets input, 0 bytes, 0 total input drops
0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 0 multicast packets
         0 runts, 0 giants, 0 throttles, 0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

Configuring MAC-Accounting: Example

This example indicates how to configure MAC-accounting on an Ethernet interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# mac-accounting egress
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# exit
```

Configuring a Layer 2 VPN AC: Example

The following example indicates how to configure a Layer 2 VPN AC on an Ethernet interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RP0/CPU0:router(config-if)# l2transport
RP/0/RP0/CPU0:router(config-if-l2)# l2protocol pvstcpsv
RP/0/RP0/CPU0:router(config-if-l2)# commit
```

Configuring LLDP: Examples

The following example shows how to enable LLDP globally on the router and modify the default LLDP operational characteristics:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# lldp
RP/0/RP0/CPU0:router(config)# lldp holdtime 60
RP/0/RP0/CPU0:router(config)# lldp reinit 4
RP/0/RP0/CPU0:router(config)# lldp timer 60
RP/0/RP0/CPU0:router(config)# commit
```

The following example shows how to disable a specific Gigabit Ethernet interface for LLDP transmission:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/0
RP/0/RP0/CPU0:router(config-if)# lldp
RP/0/RP0/CPU0:router(config-lldp)# transmit disable
```

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 RouterAdvanced 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.

CHAPTER 6

Configuring Ethernet OAM

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

Feature History for Configuring Ethernet OAM

Release	Modification
Release 3.9.0	Support for the following features was introduced: • Ethernet Link OAM • Ethernet CFM
Release 3.9.1	Support for the following features was introduced: • EFD • AIS • The **ethernet cfm mep domain** command is replaced by the **ethernet cfm** and **mep domain** commands.

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 as an option to the following commands:
	  • **action capabilities-conflict**
	  • **action discovery-timeout**
	  • **action session-down**
	  • **action uni-directional link-fault**
	• Support for the Ethernet SLA feature was introduced, including some new areas of SLA support in Cisco IOS XR software including:
	  • Support for on-demand Ethernet SLA operations using the **ethernet sla on-demand operation** command.
	  • 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 4.1.0	Support for CFM Y.1731 ITU Carrier Code (ICC)-based MEG ID (MAID) format was introduced.
Release 4.3.0	Support for ITU-T Y.1731 Synthetic Loss Measurement was introduced.
Release 4.3.1	Support for ITU-T Y.1731 Loss Measurement was introduced.
Release 5.1.0	Support for Ethernet Data Plane Loopback was introduced.
Release 5.1.2	Support for Ethernet CFM down MEP was included.

- Configuring Ethernet OAM, on page 121
- Prerequisites for Configuring Ethernet OAM, on page 123

Configuring Ethernet OAM

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

Feature History for Configuring Ethernet OAM

Release	Modification
Release 3.9.0	Support for the following features was introduced: • Ethernet Link OAM • Ethernet CFM
Release 3.9.1	Support for the following features was introduced: • EFD • AIS • The **ethernet cfm mep domain** command is replaced by the **ethernet cfm** and **mep domain** commands.

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 as an option to the following commands:
	  • **action capabilities-conflict**
	  • **action discovery-timeout**
	  • **action session-down**
	  • **action uni-directional link-fault**
	• Support for the Ethernet SLA feature was introduced, including some new areas of SLA support in Cisco IOS XR software including:
	  • Support for on-demand Ethernet SLA operations using the **ethernet sla on-demand operation** command.
	  • 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 4.1.0	Support for CFM Y.1731 ITU Carrier Code (ICC)-based MEG ID (MAID) format was introduced.
Release 4.3.0	Support for ITU-T Y.1731 Synthetic Loss Measurement was introduced.
Release 4.3.1	Support for ITU-T Y.1731 Loss Measurement was introduced.
Release 5.1.0	Support for Ethernet Data Plane Loopback was introduced.
Release 5.1.2	Support for Ethernet CFM down MEP 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 or Cisco ASR 9000 Enhanced Ethernet line cards are installed on the router.

- 4-Port Gigabit Ethernet physical layer interface module (PLIM)
- 8-Port 10-Gigabit Ethernet PLIM
- 4-Port 10-Gigabit Ethernet PLIM
- 2-Port Gigabit Ethernet SPA
- 5-Port Gigabit Ethernet SPA
- 8-Port Gigabit Ethernet SPA
- 10-Port Gigabit Ethernet SPA
- 1-Port 10-Gigabit Ethernet SPA
- 1-Port 10-Gigabit Ethernet WAN SPA

Restrictions for Configuring Ethernet OAM

The following functional areas of Ethernet OAM are not supported on the Cisco CRS-1 Router in Cisco IOS XR Release 4.1:

- Hello interval configuration
- Remote loopback
- Unidirectional link-fault detection

Note Cisco CRS1-SIP-800 and CRS1-SIP-700 cards do not support symbol-error. From Cisco IOS XR Software Release 3.9 onwards, the configuration of Ethernet OAM is restricted based on the capability flag of the PLIM.

Information About Configuring Ethernet OAM

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

Ethernet Link OAM

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

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

Neighbor Discovery

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

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.

SNMP Traps

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

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 CRS Router supports these functions of ITU-T Y.1731:

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

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

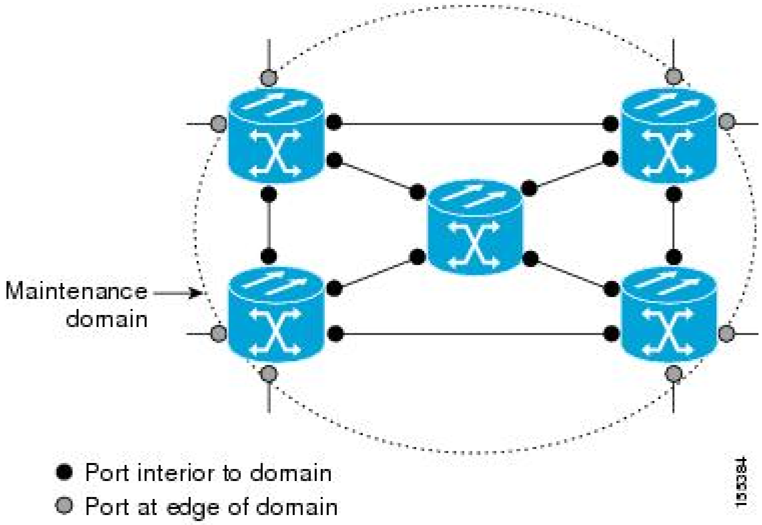
- ETH-AIS—The reception of ETH-LCK messages is also supported.
- ETH-DM, ETH-SLM—This is supported with the Ethernet SLA feature. For more information about Ethernet SLA, see the Ethernet SLA.

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

Maintenance Domains

A *maintenance domain* describes a management space for the purpose of managing and administering a network. A domain is owned and operated by a single entity and defined by the set of interfaces internal to it and at its boundary, as shown in this figure.

Figure 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:

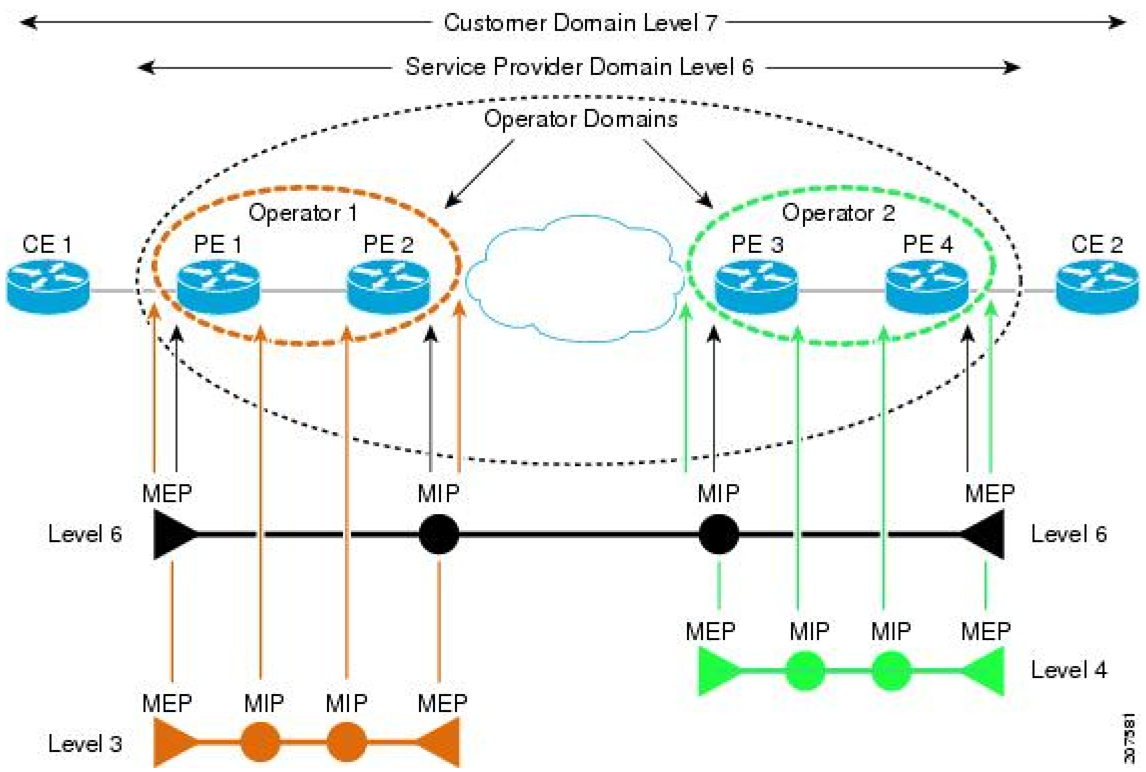
- The customer can use CFM between their CE devices, to verify and manage connectivity across the whole network.
- The service provider can use CFM between their PE devices, to verify and manage the services they are providing.
- Each operator can use CFM within their operator network, to verify and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

Each organization uses a different CFM maintenance domain.

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

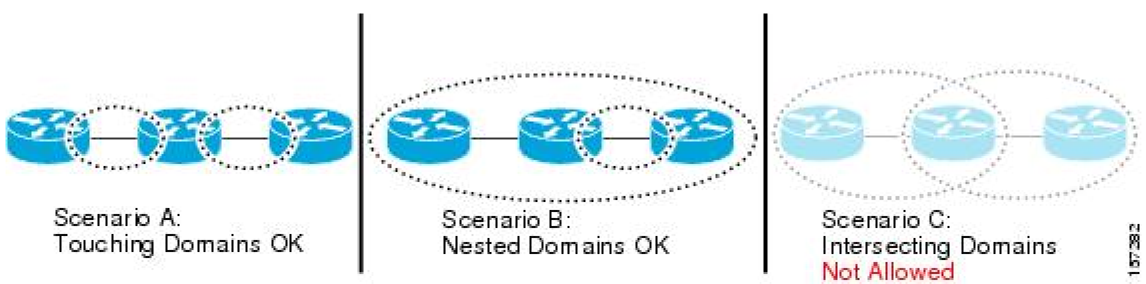
Note In CFM diagrams, the conventions are that triangles represent MEPs, pointing in the direction that the MEP sends CFM frames, and circles represent MIPs. For more information about MEPs and MIPs, see the Maintenance Points.

Figure 2: Different CFM Maintenance Domains Across a Network



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

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



Services

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

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

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

Maintenance Points

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

A maintenance point is always associated with a particular CFM service, and therefore with a particular maintenance domain at a particular level. Maintenance points generally only process CFM frames at the same level as their associated maintenance domain. Frames at a higher maintenance level are always forwarded transparently, while frames at a lower maintenance level are normally dropped. This helps enforce the maintenance domain hierarchy described in the Maintenance Domains, 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 cross-connect for the interface is found, and all services associated with that cross-connect are considered for MIP auto-creation.
- The level of the highest-level MEP on the interface is found. From among the services considered above, the service in the domain with the lowest level that is higher than the highest MEP level is selected. If there are no MEPs on the interface, the service in the domain with the lowest level is selected.

- The MIP auto-creation configuration (**mip auto-create** command) for the selected service is examined to determine whether a MIP should be created.

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

MEP and CFM Processing Overview

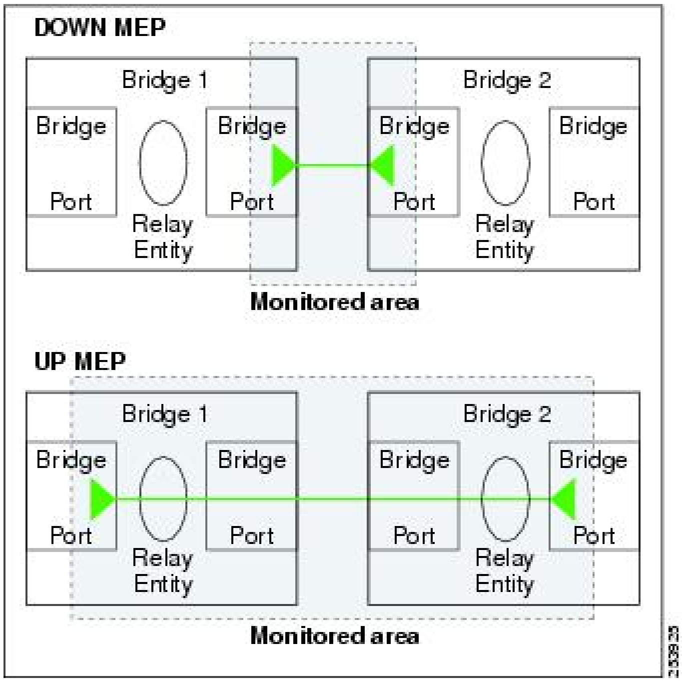
The boundary of a domain is an interface, rather than a bridge or host. Therefore, MEPs can be sub-divided into two categories:

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

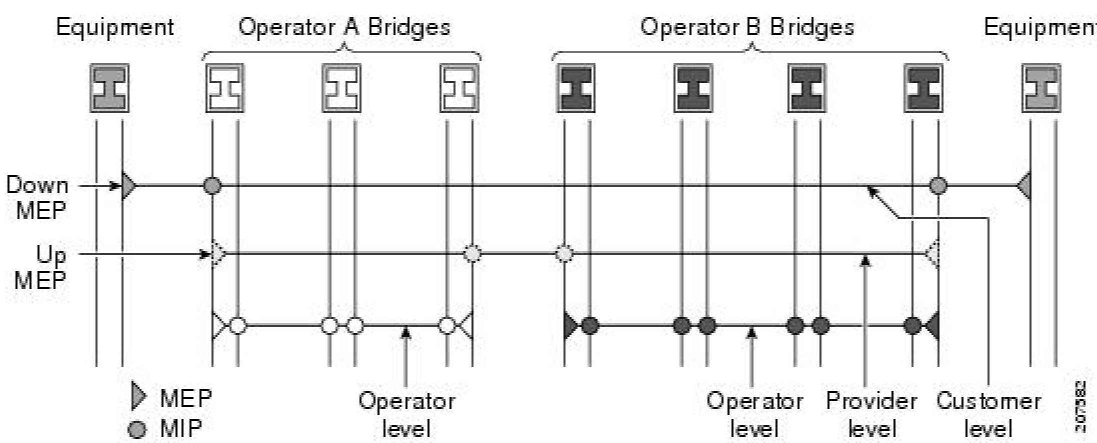
Note The terms *Down MEP* and *Up MEP* are defined in the IEEE 802.1ag and ITU-T Y.1731 standards, and refer to the direction that CFM frames are sent from the MEP. The terms should not be confused with the operational status of the MEP.

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

Figure 3: Monitored Areas for Down and Up MEPs



This figure shows maintenance points at different levels. Because domains are allowed to nest but not intersect (see 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.



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.

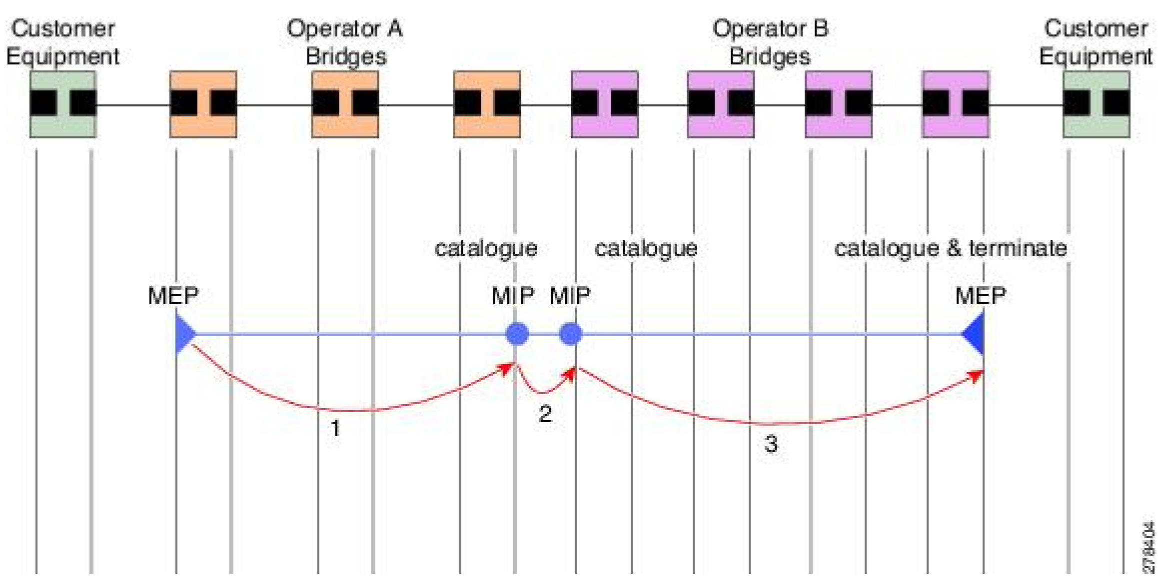
This section describes the following CFM messages:

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

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

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

Figure 4: Continuity Check Message Flow



All the MEPs in a service must transmit CCMs at the same interval. IEEE 802.1ag defines 7 possible intervals that can be used:

- 10ms (applicable on the Cisco ASR 9000 Enhanced Ethernet Line Card)
- 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).

These 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.

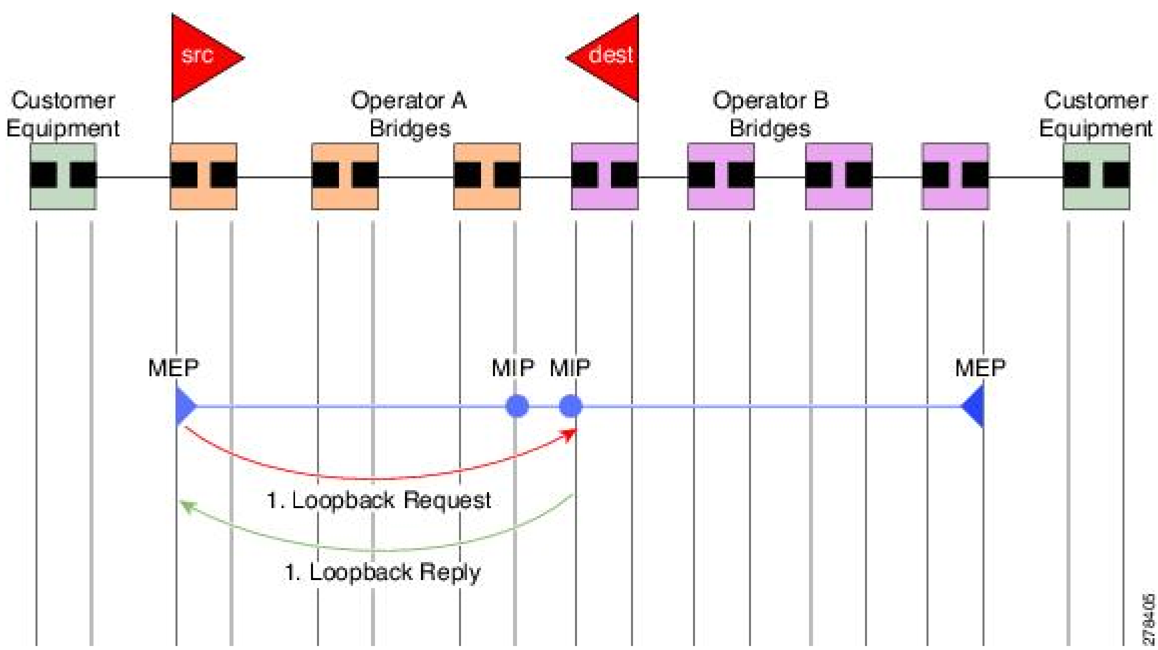
Out-of-sequence CCMs can also be detected by monitoring the sequence number in the received CCMs from each peer MEP. However, this is not considered a CCM defect.

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

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

Figure 5: Loopback Messages



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

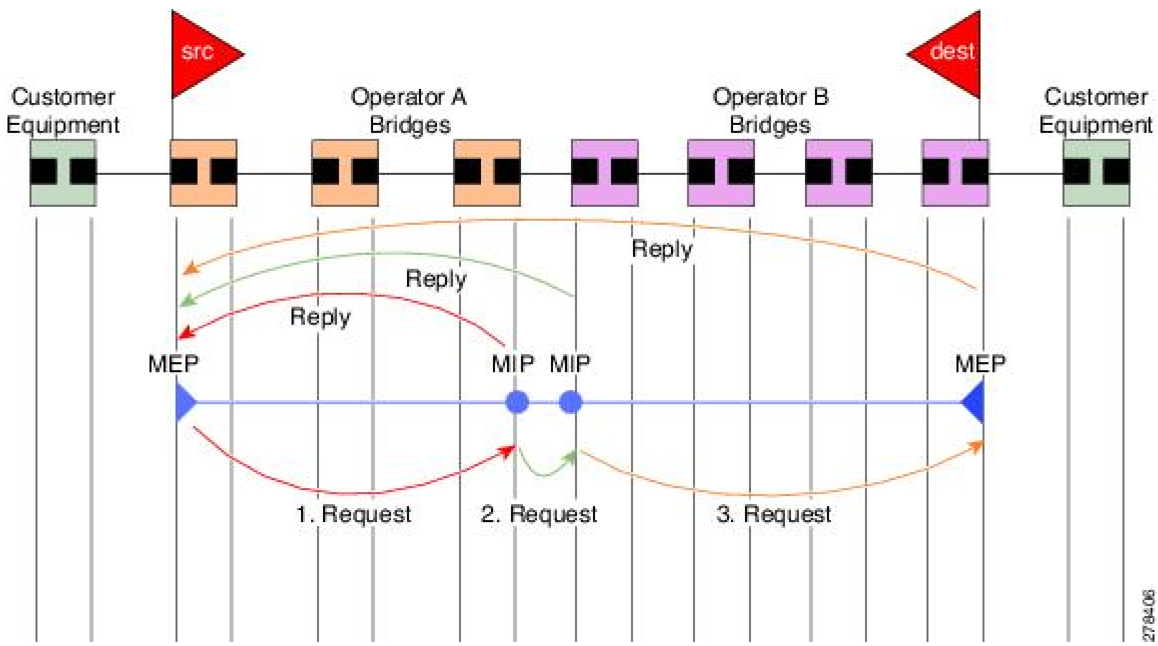
Except for one-way delay and jitter measurements, loopback messages can also be used for Ethernet SLA, if the peer does not support delay measurement.

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

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

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

Figure 6: Linktrace Message Flow



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

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

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

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

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

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

Exploratory Linktrace (Cisco)

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

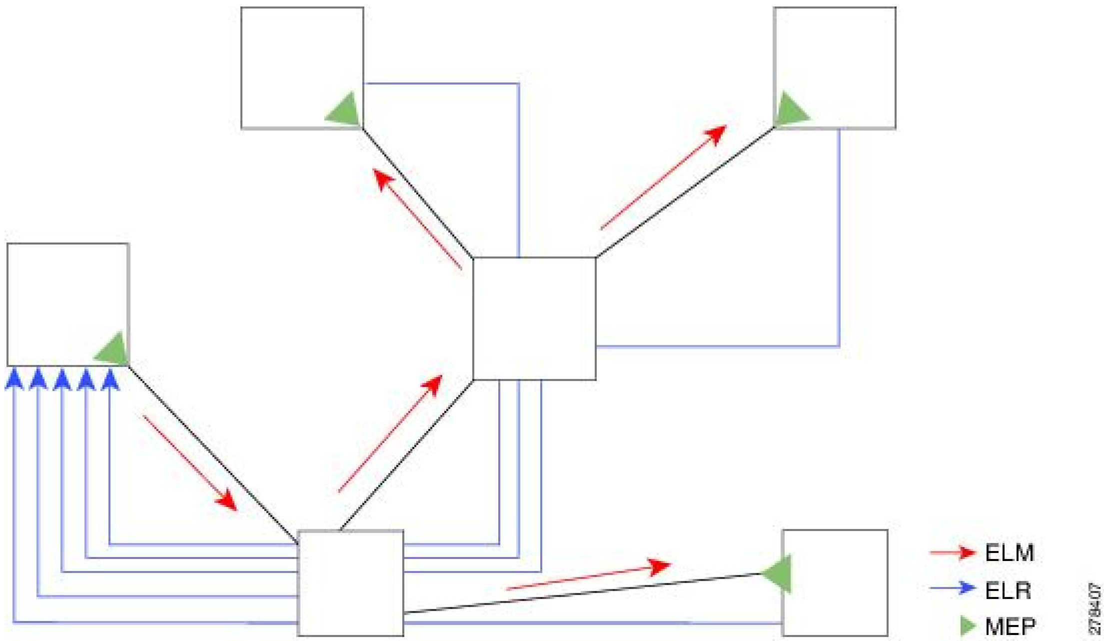
- Provide a mechanism to locate faults in cases where standard linktrace does not work, such as when a MAC address has never been seen previously in the network. For example, if a new MEP has been provisioned but is not working, standard linktrace does not help isolate a problem because no frames will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

This figure show an example of the Exploratory Linktrace message flow between MEPs.

Figure 7: Exploratory Linktrace Messages and Replies



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.

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.

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.

Synthetic Loss Measurement (ITU-T Y.1731)

Synthetic Loss Measurement (SLM) is a mechanism that injects synthetic measurement probes, and measures the loss of these probes in order to measure the loss of real data traffic. Each probe packet carries a sequence number, and the sender increments the sequence number by one for each packet that is sent and the receiver can thereby detect the lost packets by looking for missing sequence numbers.

SLM packets contain two sequence numbers; one written by the initiator into the SLM and copied by the responder into the SLR, and the other allocated by the responder and written into the SLR. These are refered to as the source-to-destination (sd) sequence number and the destination-to-source (ds) sequence number respectively.

This figure shows an example of how the sequence numbers are used to calculate the Frame Loss Ratio (FLR) in each direction.

Figure 8: Synthetic Loss Measurement

Loss Measurement (ITU-T Y.1731)

Y.1731 Loss Measurement is a mechanism that measures the actual data traffic loss between a pair of MEPs in a point-to-point Ethernet service. This is in contrast to the Synthetic Loss Measurement, which measures the frame loss of synthetic frames. By using Y.1731 Loss Measurement, you can measure the one-way loss in each direction, for each priority class and also measure the loss aggregated across all priority classes.

To enable loss measurements to be made, each MEP maintains, for each priority class, both source-to-destination and destination-to-source frame counts for its peer MEPs.

There are two Loss Measurement Mechanisms (LMM); namely, single-ended and dual-ended. Cisco IOS XR Software supports only single-ended LMM.

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.

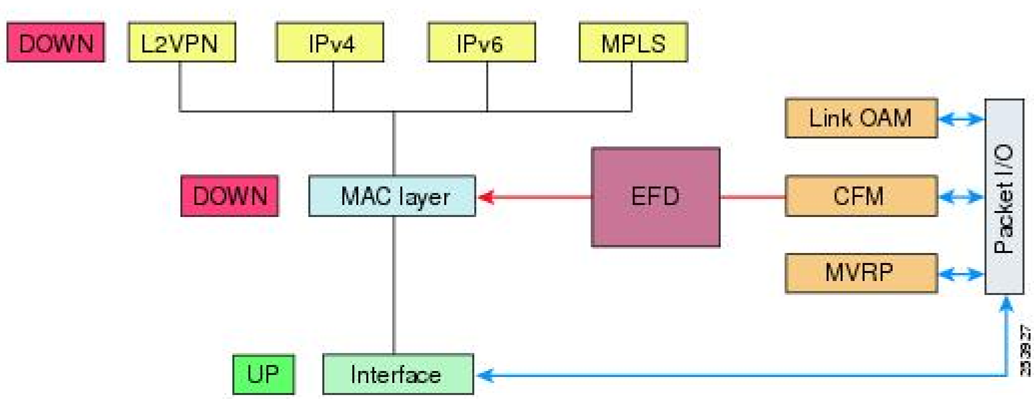
Unlike many other interface types, Ethernet interfaces do not have a line protocol, whose state is independent from that of the interface. For Ethernet interfaces, this role is handled by the physical-layer Ethernet protocol itself, and therefore if the interface is physically up, then it is available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

This figure shows CFM detection of an error on one of its sessions EFD signaling an error to the corresponding MAC layer for the interface. This triggers the MAC to go to a down state, which further triggers all higher level protocols (Layer 2 pseudowires, IP protocols, and so on) to go down and also trigger a reconvergence where possible. As soon as CFM detects there is no longer any error, it can signal to EFD and all protocols will once again go active.

Figure 9: CFM Error Detection and EFD Trigger



Ethernet SLA

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. Similarly, 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 these 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).

Y.1731 Performance Monitoring

The ITU-T Y.1731 standard defines several mechanisms that can be used for performance monitoring in Carrier Ethernet networks. These are the measurement mechanisms that were defined in the standard:

Delay Measurement: This can be used to accurately measure frame delay by exchanging CFM frames containing timestamps, and to measure inter-frame delay variation (jitter) by comparing consecutive delay measurements. Delay Measurement messages can be used to perform these measurements:

- Round-trip time
- Round-trip Jitter
- One-way delay (both SD and DS)
- One-way jitter (both SD and DS)
- SLA Probe Packet corruption count
- Out of order SLA probe packet count
- SLA probe packet loss

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

This alternative mechanism does not measure the loss of the actual data traffic, but instead injects synthetic CFM frames and measures the loss of these synthetic frames. Statistical analysis can then be used to give an approximation to the loss of data traffic. This technique is called Synthetic Loss Measurement. This has been included in the latest version of the Y.1731 standard. Synthetic Loss Measurement messages can be used to perform these measurements:

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

Loopback: This is not primarily targetted at performance monitoring, but can be used to approximate round-trip delay and jitter, such as when the peer device does not support delay measurement. Loopback messages can be used to perform these measurements:

- Round-trip time
- Round-trip jitter
- SLA probe packet corruption count
- Out of order SLA probe packet count
- SLA probe packet loss

Loss Measurement Terminology

These are the commonly used terminology in Loss Measurement Mechanism:

- **Single-ended**: A mechanism where device A sends a measurement packet to device B, which in turn sends a response back to device A. All calculations and results are done on device A.
- **Dual-ended**: A mechanism where device A sends a measurement packet to device B, which does not send a response. All calculations and results are done on device B.

- **One-way**: A measurement of the performance of packets flowing in one direction, from device A to device B, or from device B to device A.
- **Two-way**: A measurement of the performance of packets flowing from device A to device B, and back to device A.
- **Forwards**: A one-way measurement from the initiator (device A) to the receiver, or responder (device B).
- **Backwards**: A one-way measurement from the responder (device B) to the initiator (device A).

Note Cisco IOS XR Software supports only single-ended LMM.

Loss Measurement Performance Attributes

These are two primary attributes that can be calculated based on loss measurements:

- Frame Loss Ratio (FLR)
- Availability

Frame Loss Ratio is the ratio of lost packets to sent packets:

(<num_sent > - <num_rcvd>)/(<num_sent>)

It is normally expressed as a percentage. The accuracy of the measurement depends majorly on the number of packets sent.

Availability is a complex attribute, typically measured over a long period of time, such as weeks or months. The intent of this performance attribute is to measure the proportion of time when there was prolonged high loss. Cisco IOS XR Software does not track the availability.

Limitations of Data Loss Measurement

1. Data loss measurement cannot be used in a multipoint service; it can only be used in a peer-to-peer service.
2. As a Loss Measurement Reply (LMR) contains no sequence IDs, the only field, which can be used to distinguish to which probe a given LMR corresponds, is the priority level. Also, the priority level is the only field that can determine whether the LMR is in response to an on-demand or proactive operation. This limits the number of Loss Measurement probes that can be active at a time for each local MEP to 16.
3. As loss measurements are made on a per-priority class basis, QoS policies, which alter the priority of packets processed by the network element, or re-order packets can affect the accuracy of the calculations. For the highest accuracy, packets must be counted after any QoS policies have been applied.
4. The accuracy of data loss measurement is highly dependent on the number of data packets that are sent. If the volume of data traffic is low, errors with the packet counts might be magnified. If there is no data traffic flowing, no loss measurement performance attributes can be calculated. If aggregate measurements are taken, then only 2 probes can be active at the same time: one proactive and one on-demand.
5. The accuracy of data loss measurement is highly dependent on the accuracy of platform-specific packet counters. Due to hardware limitations, it may not be possible to achieve completely accurate packet counters, especially if QoS policies are applied to the packets being counted.

6. Performing data loss measurement can have an impact on the forwarding performance of network elements; this is because of the need to count, as well as forward the packets.
7. Before starting any LMM probes,it is necessary to allocate packet counters for use with LMM on both ends (assuming both ends are running Cisco IOS XR Software).

Ethernet SLA Concepts

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

Loss Measurement Terminology

A *statistic* in Ethernet SLA is a single performance parameter. These 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 frame loss from source to destination
- One-way delay from destination to source
- One-way jitter from destination to source
- One-way frame loss 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. These 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.

Note From Cisco IOS XR Release 4.3.x onwards, you can configure the Ethernet SLA profile to use Y.1731 DMM v1 frames. The restriction of 150 configured Ethernet SLA operations for each CFM MEP is removed not only for profiles using DMM frames, but also for profiles using the other supported Y.1731 frame types, such as loopback measurement and synthetic loss measurement. For interoperability purposes, it is still possible to configure operations to use DMM v0 frames. This is done by specifying a type of **cfm-delay-measurement-v0** on the **ethernet SLA profile** command. The limit of 150 configured operations for each CFM MEP still applies in this case.

- 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.
- CFM Synthetic Loss Measurement (Y.1731 SLM/SLR packets)—SLM packets contain two sequence numbers; one written by the initiator into the SLM and copied by the responder into the SLR, and the other allocated by the responder and written into the SLR. These are refered to as the source-to-destination (sd) sequence number and the destination-to-source (ds) sequence number respectively.

Note Because SLM is a statistical sampling technique, there may be some variance of the measured value around the actual loss value. Also, the accuracy of the measurement is improved by using more SLM packets for each FLR calculation.

- CFM Loss Measurement (Y.1731 LMM/LMR packets)— As LMMs and LMRs contain no sequence ID, there is a limited set of data that can be used to distinguish different Loss Measurement operations, limiting the number of concurrent operations for each MEP.

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.

Note 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).

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.

Note If bursts are configured back to back, so as to cause a continuous and uninterrupted flow of SLA packets, then packets at the end of one burst and the start of the next are used in Loss Measurement calculations.

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.

Note If probes are scheduled back to back, so as to cause a continuous and uninterrupted flow of SLA packets, then packets at the end of one probe and the start of the next are used in Loss Measurement calculations.

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 this figure.

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.
- One-way frame loss—The router also supports measurement of one-way frame loss from source to destination, or from destination to source.

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

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

Counters for packet loss, corruption and out-of-order packets are kept for each bucket, and in each case, a percentage of the total number of samples for that bucket is reported (for example, 4% packet corruption). For delay, jitter, and loss statistics, the minimum, maximum, mean and standard deviation for the whole bucket are reported, as well as the individual samples or aggregated bins. Also, the overall FLR for the bucket, and individual FLR measurements or aggregated bins are reported for synthetic loss measurement statistics. The packet loss count is the overall number of measurement packets lost in either direction and the one-way FLR measures the loss in each direction separately.

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

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

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.

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

Configuration Overview of Scheduled Ethernet SLA Operations

When you configure a scheduled Ethernet SLA operation, you perform these 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.

Ethernet Data Plane Loopback

The Ethernet Data Plane Loopback feature allows you to test services and throughput of an Ethernet port or a device using a test generator. You can verify the maximum rate of frame transmission with no frame loss. This feature allows bidirectional throughput measurement, and on-demand or out-of-service (intrusive) operation during service turn-ups. This feature can be used for testing during service turn-ups and troubleshooting of services after a turn-up.

If you need to test a service while it is live, you can do this without disrupting any of the live data traffic. To achieve this, you can use test traffic that differs from live data traffic. For example, the traffic from a test generator can contain the source MAC address of the test generator, or test traffic may be assigned a particular

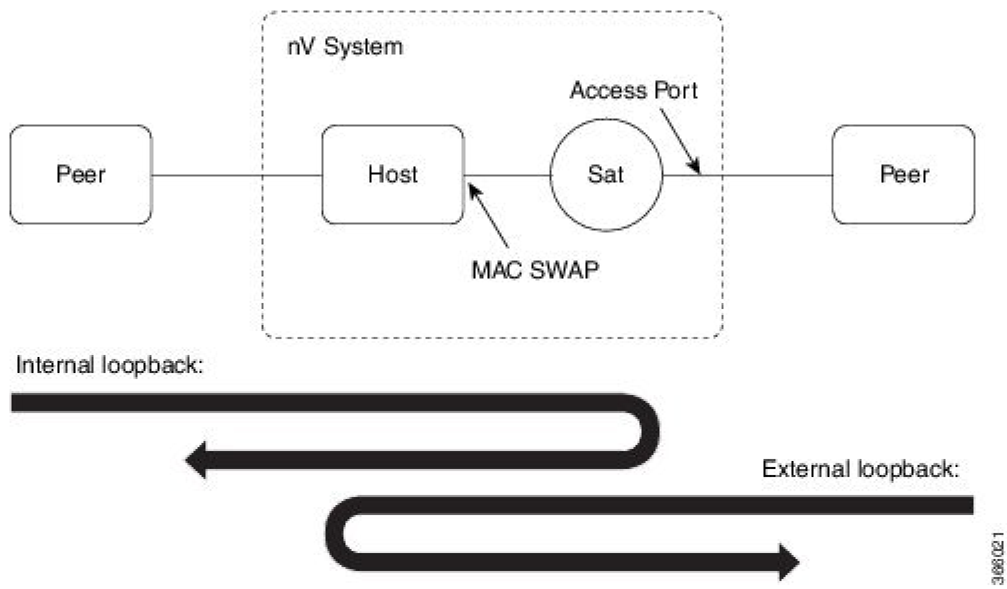
Class of Service (CoS). Irrespective of the method used, the device looping back the traffic must be able to filter out the test traffic and leave the data traffic untouched.

Note Configuring Ethernet Data Plane Loopback on a device does not indicate the start of an actual session.

Ethernet Data Plane Loopback on Satellite nV System

The Ethernet Data Plane Loopback (EDPL) is implemented on the Satellite nV System as shown in this illustration.

Figure 10: EDPL on Satellite nV System



The internal and external EDPL are realized as follows:

- **Internal Loopback**: The MAC address swap happens on the host and the frame actually gets looped back from the satellite where Layer 1 loopback needs to be turned on at the port. As the entire port is looped back on the satellite, the internal loopback for satellite ports cannot loopback or filter specific sub-interface sessions on the port. You need to enable both EDPL and port L1 internal loopback on the satellite port for this functionality.
- **External Loopback**: The external loopback is currently implemented entirely on the host because of the need to perform MAC address swap.

Features Supported for Ethernet Data Plane Loopback

The support that the Ethernet Data Plane Loopback feature provides is:

- Locally-enabled Ethernet Data Plane Loopback on all Ethernet interface types, such as physical and bundle interfaces and sub-interfaces.
- In the case of Layer 2 interfaces, support for these types of looping back of traffic:

  - External loopback – All traffic received on the ingress interface is blindly sent out of the egress interface.
  - Internal loopback – All traffic received on the egress interface is blindly injected into the ingress interface.
- In the case of Layer 3 interfaces, only external loopback is supported.
- When a Bundle interface is placed into loopback, traffic on all bundle link members are looped back.
- MAC address must always be swapped on looped-back traffic.
- Supports dropping of packets received in the non-loopback direction.
- Allows the application of multiple filters to loopback only a subset of traffic received by an interface and only drop the corresponding reverse-direction traffic.
- Provides an option to specify a time period after which the loopback is automatically terminated.
- Supports at least 100 simultaneous loopback sessions across the system.

Limitations of Ethernet Data Plane Loopback

These are the limitations of Ethernet Data Plane Loopback (EDPL):

- Layer 3 interfaces including pseudowires are not supported in internal EDPL.
- The first generation Cisco ASR 9000 Ethernet Line Cards are not supported.
- Virtual interfaces such as BVI are not supported.
- Filtering based on LLC-OUI is not supported.
- A maximum of 50 simultaneous loopback sessions are supported for each Network Processor on the linecard.
- LAG bundles that are member of Satellite nV interface over bundle inter-chassis link (also known as LAG over LAG bundles) are not supported.

How to Configure Ethernet OAM

This section provides these configuration procedures:

Configuring Ethernet Link OAM

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

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

Configuring an Ethernet OAM Profile

Perform these steps to configure an Ethernet OAM profile.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**ethernet oam profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config)# ethernet oam profile Profile_1`	Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.
Step 3	**link-monitor** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# link-monitor`	Enters the Ethernet OAM link monitor configuration mode.
Step 4	**symbol-period window** { **milliseconds** *window* | **symbols** *window* [**thousand** | **million** | **billion**]} **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period window 60000`	(Optional) Configures the window size for an Ethernet OAM symbol-period error event. If specified in milliseconds, the range is 1000 to 60000. If not specified as a multiple of 1 second, the actual window used will be rounded up to the nearest second, with thresholds scaled accordingly. If specified in symbols, the range is interface speed dependent (must be between the maximum number of symbols that could be received in 1 second and the maximum number of symbols that could be received in 1 minute). Again the actual window used is rounded up to the nearest second, with thresholds scaled accordingly. The default value is 1000 milliseconds.
Step 5	**symbol-period threshold** { **ppm** [**low** *threshold*] [**high** *threshold*] | **symbols** [**low** *threshold* [**thousand** | **million** | **billion**]] [**high** *threshold* [**thousand** | **million** | **billion**]]} **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period threshold symbols low 10000000 high 60000000`	(Optional) Configures the thresholds that trigger an Ethernet OAM symbol-period error event, in symbols or ppm (errors per million symbols). When using this command at least one of the high and low thresholds must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no action is performed in response to an event. The high threshold must not be smaller than the low threshold. If specified in ppm, the range (for both thresholds) is 1 to 1000000. If specified in symbols, the range (for both thresholds) is 1 to the maximum window size in symbols, see Step 4. The default low threshold is 1 symbol.
Step 6	**frame window milliseconds** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame window milliseconds 60`	(Optional) Configures the frame window size (in milliseconds) of an OAM frame error event. The range is from 1000 to 60000. The default value is 1000.
Step 7	**frame threshold** [**low** *threshold*] [**high** *threshold*] **Example:**	(Optional) Configures the thresholds (in symbols) that triggers an Ethernet OAM frame error event. When using this command at least one of the high and low thresholds

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000`	must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no action is performed in response to an event. The high threshold must not be smaller than the low threshold. The range is from 1 to 60000000. The default low threshold is 1.
Step 8	**frame-period window** { **milliseconds** *window* | **frames** *window* [**thousand** | **million** | **billion**]} **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window milliseconds 60000`	(Optional) Configures the window size for an Ethernet OAM frame-period error event. The range is from 100 to 60000, if defined in milliseconds. If the window is defined as say, 200ms, and the interface could receive at most say 10000 minimum size frames in 200ms, then the actual window size used will be the time taken to receive 10000 frames, rounded up to the nearest second. The thresholds will be scaled accordingly. If specified in frames, the range is interface speed dependent, but must be between the number of minimum size frames that could be received in 100ms and the number of minimum size frames that could be received in 1 minute. If the window is defined as 20000 frames, the actual window size used will be the time taken to receive 20000 frames, rounded up to the nearest second. The thresholds will be scaled accordingly. The default value is 1000 milliseconds.
Step 9	**frame-period threshold** { **ppm** [**low** *threshold*] [**high** *threshold*] | **frames** [**low** *threshold* [**thousand** | **million** | **billion**]] [**high** *threshold* [**thousand** | **million** | **billion**]]} **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period threshold ppm low 100 high 1000000`	(Optional) Configures the thresholds (either in frames or in ppm - errors per million frames) that trigger an Ethernet OAM frame-period error event. When using this command at least one of the high and low thresholds must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no action is performed in response to an event. The high threshold must not be smaller than the low threshold. The range for both thresholds is from 1 to 1000000 if specified in ppm. If specified in frames, the range is from 1 to the maximum frame-period window size in frames, see Step 4. The default low threshold is 1 ppm.
Step 10	**frame-seconds window milliseconds** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds window milliseconds 900000`	(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000. The default value is 6000.
Step 11	**frame-seconds threshold** [**low** *threshold*] [**high** *threshold*]	(Optional) Configures the thresholds (in seconds) that trigger a frame-seconds error event. When using this

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 threshold high 900`	command at least one of the high and low thresholds must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no action is performed in response to an event. The high threshold must not be smaller than the low threshold. The range is 1 to 900 The default value is 1.
Step 12	**exit** **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# exit`	Exits back to Ethernet OAM mode.
Step 13	**mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mib-retrieval`	Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.
Step 14	**connection timeout** <*timeout*> **Example:** `RP/0/RP0/CPU0:router(config-eoam)# connection timeout 30`	Configures the connection timeout period for an Ethernet OAM session. as a multiple of the hello interval. The range is 2 to 30. The default value is 5.
Step 15	**hello-interval** {**100ms**|**1s**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# hello-interval 100ms`	Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (**1s**).
Step 16	**mode** {**active**|**passive**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mode passive`	Configures the Ethernet OAM mode. The default is active.
Step 17	**require-remote mode** {**active**|**passive**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# require-remote mode active`	Requires that active mode or passive mode is configured on the remote end before the OAM session becomes active.
Step 18	**require-remote mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval`	Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.

	Command or Action	Purpose
Step 19	**action capabilities-conflict** {**disable** | **efd** | **error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action capabilities-conflict efd`	Specifies the action that is taken on an interface when a capabilities-conflict event occurs. The default action is to create a syslog entry.
Step 20	**action critical-event** {**disable** | **error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action critical-event error-disable-interface`	Specifies the action that is taken on an interface when a critical-event notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry.
Step 21	**action discovery-timeout** {**disable** | **efd** | **error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action discovery-timeout efd`	Specifies the action that is taken on an interface when a connection timeout occurs. The default action is to create a syslog entry.
Step 22	**action dying-gasp** {**disable** | **error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface`	Specifies the action that is taken on an interface when a dying-gasp notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry.
Step 23	**action high-threshold** {**disable** | **error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action high-threshold error-disable-interface`	Specifies the action that is taken on an interface when a high threshold is exceeded. The default is to take no action when a high threshold is exceeded.
Step 24	**action session-down** {**disable** | **efd** | **error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`	Specifies the action that is taken on an interface when an Ethernet OAM session goes down.
Step 25	**action session-up** { **disable** | **log** } **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-up disable`	Specifies that no action is taken on an interface when an Ethernet OAM session is established. The default action is to create a syslog entry.

	Command or Action	Purpose
Step 26	**action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface** | **log**}	Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** In Cisco IOS XR Release 4.x, this command replaces the **action link-fault** command.
Step 27	**action wiring-conflict** {**disable** | **efd** | **error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`	Specifies the action that is taken on an interface when a wiring-conflict event occurs. The default is to put the interface into error-disable state.
Step 28	**uni-directional link-fault detection** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# uni-directional link-fault detection`	Enables detection of a local, unidirectional link fault and sends notification of that fault to an Ethernet OAM peer.
Step 29	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves the configuration changes to the running configuration file and remains within the configuration session.
Step 30	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Attaching an Ethernet OAM Profile to an Interface

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

SUMMARY STEPS

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

DETAILED STEPS

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

Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

Using an EOAM profile is an efficient way of configuring multiple interfaces with a common EOAM configuration. However, if you want to use a profile but also change the behavior of certain functions for a particular interface, then you can override the profile configuration. To override certain profile settings that are applied to an interface, you can configure that command in interface Ethernet OAM configuration mode to change the behavior for that interface. In the case that there is non-default configuration used in the EOAM profile, you can use keywords representing the default behaviour (which have no effect if used in Profile configuration) in interface Ethernet OAM configuration mode to return to the default behaviour on that particular interface.

To see all of the default Ethernet OAM configuration settings, see the *Verifying the Ethernet OAM Configuration* section.

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

SUMMARY STEPS

1. **configure**
2. **interface** [GigabitEthernet | TenGigE] *interface-path-id*
3. **ethernet oam**
4. *interface-Ethernet-OAM-command* RP/0/RP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface
5. **commit**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 2	**interface** [GigabitEthernet | TenGigE] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. **Note** • The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
Step 3	**ethernet oam** **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet oam`	Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.
Step 4	*interface-Ethernet-OAM-command* RP/0/RP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface	Configures a setting for an Ethernet OAM configuration command and overrides the setting for the profile configuration, where *interface-Ethernet-OAM-command* is one of the supported commands on the platform in interface Ethernet OAM configuration mode.
Step 5	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves the configuration changes to the running configuration file and remains within the configuration session.
Step 6	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Verifying the Ethernet OAM Configuration

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

- Some of these settings are not supported on certain platforms, but the defaults are still reported. On the Cisco CRS-1 Router, the following areas are unsupported:
- Hello interval configuration
- Remote loopback
- Uni-directional link-fault detection

```
RP/0/RP0/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

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

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**traceroute cache hold-time** *minutes* **size** *entries* **Example:** `RP/0/RP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000`	(Optional) Sets the maximum limit of traceroute cache entries or the maximum time limit to hold the traceroute cache entries. The default is 100 minutes and 100 entries.
Step 5	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Services for a CFM Maintenance Domain

You can configure up to 32000 CFM services for a maintenance domain.

Before you begin

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** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]
5. **end** or **commit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	**service** *service-name* {**bridge group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling and Configuring Continuity Check for a CFM Service

The Cisco CRS-1 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 2000 CCMs-per-second sent, and up to 2000 CCMs-per-second received, per card.

Note If Ethernet SLA is configured, the overall combined packet rate for CCMs and SLA frames is 4000 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*]]

4. **service** *service-name* {**bridge group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]
5. **continuity-check interval** *time* [**loss-threshold** *threshold*]
6. **continuity-check archive hold-time** *minutes*
7. **continuity-check loss auto-traceroute**
8. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**bridge group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**continuity-check interval** *time* [**loss-threshold** *threshold*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10`	(Optional) Enables Continuity Check and specifies the time interval at which CCMs are transmitted or to set the threshold limit for when a MEP is declared down.
Step 6	**continuity-check archive hold-time** *minutes* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100`	(Optional) Configures how long information about peer MEPs is stored after they have timed out.

	Command or Action	Purpose
Step 7	**continuity-check loss auto-traceroute** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute`	(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.
Step 8	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Automatic MIP Creation for a CFM Service

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

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** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]
5. **mip auto-create** {**all** | **lower-mep-only**}
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router# ethernet cfm`	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**bridge group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**mip auto-create** {**all** | **lower-mep-only**} **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all`	(Optional) Enables the automatic creation of MIPs in an xconnect.
Step 6	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Cross-Check on a MEP for a CFM Service

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

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	**service** *service-name* {**bridge group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**mep crosscheck** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10`	Enters CFM MEP crosscheck configuration mode.
Step 6	**mep-id** *mep-id-number* [**mac-address** *mac-address*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10`	Enables cross-check on a MEP. **Note** • Repeat this command for every MEP that you want included in the expected set of MEPs for cross-check.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Other Options for a CFM Service

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router# ethernet cfm`	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**bridge group**[**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**maximum-meps** *number* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000`	(Optional) Configures the maximum number (2 to 8190) of MEPs across the network, which limits the number of peer MEPs recorded in the database.
Step 6	**log** {**ais**|**continuity-check errors**|**continuity-check mep changes**|**crosscheck errors**|**efd**} **Example:**	(Optional) Enables logging of certain types of events.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors`	
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring CFM MEPs

When you configure CFM MEPs, consider these guidelines:

- Up to 32000 local MEPs are supported per card.
- CFM maintenance points can be created only on physical Ethernet interfaces.
- A new configuration under the MEP submode called loss-measurement counters is used to allocate the packet counters used for LMM.
- CCM packet must not go through L3VPN cloud.
- LBM/LBR packet must not go through L3VPN cloud.
- LTM/LTR packet must not go through L3VPN cloud.
- DMM/DMR packet must not go through L3VPN cloud.
- SLM/SLR packet must not go through L3VPN cloud.
- LMM/LMR packet must not go through L3VPN cloud.

SUMMARY STEPS

1. **configure**

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** {**GigabitEthernet | TenGigE**} *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	Type of Ethernet interface on which you want to create a MEP. Enter **GigabitEthernet** or **TenGigE** and the physical interface or virtual interface. **Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (**?**) online help function.
Step 3	**interface** {**GigabitEthernet | TenGigE | Bundle-Ether**} *interface-path-id.subinterface* **Example:** `RP/0//CPU0:router(config)# interface gigabitethernet 0/1/0/1`	Type of Ethernet interface on which you want to create a MEP. Enter **GigabitEthernet, TenGigE,** or **Bundle-Ether** and the physical interface or virtual interface followed by the subinterface path ID. Naming notation is *interface-path-id.subinterface*. The period in front of the subinterface value is required as part of the notation. For more information about the syntax for the router, use the question mark (**?**) online help function.
Step 4	**vrf vrf-name** **Example:** `RP/0/RP0/CPU0:router(config-if)# vrf vrf_A`	Configures a VRF instance and enters VRF configuration mode. For more information on configuring VRF interfaces, refer the *Connecting MPLS VPN Customers* section in the *Cisco ASR 9000 Series MPLS Layer 3 VPN Configuration Guide*.
Step 5	**interface** {**FastEthernet | GigabitEthernet | TenGigE**} *interface-path-id* **Example:**	Type of Ethernet interface on which you want to create a MEP. Enter **FastEthernet, GigabitEthernet** or **TenGigE** and the physical interface or virtual interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	**Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (**?**) online help function.
Step 6	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet cfm`	Enters interface Ethernet CFM configuration mode.
Step 7	**mep domain** *domain-name* **service** *service-name* **mep-id** *id-number* **Example:** `RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1`	Creates a maintenance end point (MEP) on an interface and enters interface CFM MEP configuration mode.
Step 8	**cos** *cos* **Example:** `RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7`	(Optional) Configures the class of service (CoS) (from 0 to 7) for all CFM packets generated by the MEP on an interface. If not configured, the CoS is inherited from the Ethernet interface.
Step 9	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Y.1731 AIS

This section has the following step procedures:

Configuring AIS in a CFM Domain Service

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0//CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0//CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM global configuration mode.
Step 3	**domain** *name* **level** *level* **Example:** `RP/0//CPU0:router(config-cfm)# domain D1 level 1`	Specifies the domain and domain level.
Step 4	**service** *name* **bridge group** *name* **bridge-domain** *name* **Example:** `RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2`	Specifies the service, bridge group, and bridge domain.
Step 5	**service** *name* **xconnect group** *xconnect-group-name* **p2p** *xconnect-name* **Example:**	Specifies the service and cross-connect group and name.

	Command or Action	Purpose
	`RP/0//CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2`	
Step 6	**ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*] **Example:** `RP/0//CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7`	Configures Alarm Indication Signal (AIS) transmission for a Connectivity Fault Management (CFM) domain service.
Step 7	**log ais** **Example:** `RP/0//CPU0:router(config-cfm-dmn-svc)# log ais`	Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.
Step 8	**end** or **commit** **Example:** `RP/0//CPU0:router(config-sla-prof-stat-cfg)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring AIS on a CFM Interface

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0//CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface gigabitethernet** *interface-path-id* **Example:** `RP/0//CPU0:router# interface gigabitethernet 0/1/0/2`	Enters interface configuration mode.
Step 3	**ethernet cfm** **Example:** `RP/0//CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM interface configuration mode.
Step 4	**ais transmission up interval 1m cos** *cos* **Example:** `RP/0//CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7`	Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.
Step 5	**end** or **commit** **Example:** `RP/0//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 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

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0//CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0//CPU0:router(config)# ethernet cfm`	Enters CFM configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* **Example:** `RP/0//CPU0:router(config-cfm-dmn)# domain D1 level 1`	Specifies or creates the CFM domain and enters CFM domain configuration mode.
Step 4	**service** *service-name* **down-meps** **Example:** `RP/0//CPU0:router(config-cfm-dmn)# service S1 down-meps`	Specifies or creates the CFM service for down MEPS and enters CFM domain service configuration mode.
Step 5	**efd** **Example:** `RP/0//CPU0:router(config-cfm-dmn-svc)# efd`	Enables EFD on all down MEPs in the down MEPS service.
Step 6	**log efd** **Example:**	(Optional) Enables logging of EFD state changes on an interface.

	Command or Action	Purpose
	`RP/0//CPU0:router(config-cfm-dmn-svc)# log efd`	
Step 7	**end** or **commit** **Example:** `RP/0//CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the EFD Configuration

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

```
RP/0/RP0/CPU0:router# show efd interfaces

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

Verifying the CFM Configuration

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

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

Troubleshooting Tips

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

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/RP0/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:

Hop Hostname/Last            Ingress MAC/name       Egress MAC/Name        Relay
--- ------------------------ ---------------------- ---------------------- -----
  1 ios                      0001.0203.0400 [Down]                         FDB
      0000-0001.0203.0400    Gi0/0/0/0
  2 abc                                             0001.0203.0401 [Ok]    FDB
      ios                                           Not present
  3 bcd                      0001.0203.0402 [Ok]                           Hit
      abc                    GigE0/0
Replies dropped: 0
```

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

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

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 CRS Router supports SLA packet of 100 ms and longer. The overall packet rates for SLA is up to 2000 CCMs-per-second sent, and up to 2000 CCMs-per-second received, per card.
- You must define the schedule before you configure SLA probe parameters to send probes for a particular profile. It is recommended to set up the profile—probe, statistics, and schedule before any commit.

Note When the **once** keyword is used for 'send burst' ('send burst once' rather than 'send burst every'), it stops the collection of statistics with the packets that cross probe boundaries.

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

To configure a profile, perform the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet sla**
3. **profile** *profile-name* **type** {**cfm-delay-measurement** | **cfm-loopback** | **cfm-synthetic-loss-measurement**|
4. **cfm-loss-measurement**}
5. **end** or **commit**

DETAILED STEPS

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **ethernet sla**

Example:

```
RP/0/RP0/CPU0:router# ethernet sla
```

Enters the SLA configuration mode.

Step 3 **profile** *profile-name* **type** {**cfm-delay-measurement** | **cfm-loopback** | **cfm-synthetic-loss-measurement**|

Step 4 **cfm-loss-measurement**}

Example:

```
RP/0/RP0/CPU0:router(config-sla)# profile  Prof1 type cfm-loopback
```

Creates an SLA operation profile and enters the SLA profile configuration mode.

Step 5 **end** or **commit**

Example:

```
RP/0/RP0/CPU0:router(config-sla)# 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 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.

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 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**}]
2. **end** or **commit**

DETAILED STEPS

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/RP0/CPU0:router(config-sla-prof)# schedule every week on Monday at 23:30 for 1 hour
or
RP/0/RP0/CPU0:router(config-sla-prof)# schedule every day at 11:30 for 5 minutes
or
RP/0/RP0/CPU0:router(config-sla-prof)# schedule every 2 hours first at 13:45:01
or

RP/0/RP0/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/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit
```

Saves configuration changes.

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

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

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

Configuring SLA Probe Parameters in a Profile

To configure SLA probe parameters in a profile, perform these 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**}
3. or
4. **send packet** {**every** *number* {**milliseconds | seconds | minutes | hours**} **| once**}
5. **packet size** *bytes* [**test pattern {hex 0xHHHHHHHH | pseudo-random**}]
6. **priority** *priority*
7. **synthetic loss calculation packets** *number*
8. **end** or **commit**

DETAILED STEPS

Step 1 **probe**

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof)# probe
```

Enters the SLA profile probe configuration mode.

Step 2 **send burst** {**every** *number* {**seconds | minutes | hours} | once**} **packet count** *packets* **interval** *number* {**seconds | milliseconds**}

Step 3 or

Step 4 **send packet** {**every** *number* {**milliseconds | seconds | minutes | hours**} **| once**}

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-pb)# send burst every 60 seconds packet count 100 interval 100
 milliseconds
or
```

```
RP/0/RP0/CPU0:router(config-sla-prof-pb)# send burst once packet count 2 interval 1 second
or

RP/0/RP0/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.

Note When the **once** keyword for 'send burst' ('send burst once' rather than 'send burst every') is used, it stops the collection of statistics with the packets that cross probe boundaries.

Step 5 **packet size** *bytes* [**test pattern {hex 0xHHHHHHHH | pseudo-random**}]

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-pb)# packet size 9000
```

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. The packet size can be configured for SLM, loopback, and DMM/R probes.

Step 6 **priority** *priority*

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-pb)# priority 7
```

Configures the priority of outgoing SLA probe packets.

If the operation is running on an interface, which matches tagged traffic, then a priority value must be configured for the probe. This priority value must match the "on-the-wire" CoS value of the packets to be counted (after any tag rewrites). LMM packets are sent with this priority value as the CoS-value, and LMR packets must be received with the same CoS-value; otherwise, all LMRs are dropped. Note that this is the case even when aggregate counters are being collected.

If the operation is running on an interface which matches untagged traffic, then configuring a priority value is not permitted. In this case, only aggregate counters can be collected. When configuring data-loss measurement operations, configuration must also be applied to allocate the correct packet counters (matching the CoS values to be collected) on the interface, using the "loss-measurement counters" configuration under the MEP properties submode.

Step 7 **synthetic loss calculation packets** *number*

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-pb)# synthetic loss calculation packets 25
```

Configures the number of packets that must be used to make each FLR calculation in the case of synthetic loss measurements. This item can only be configured for packet types that support synthetic loss measurement.

An FLR value is calculated for each discrete block of packets. For instance, if a value of 10 is configured, the first FLR value would be calculated based on packets 0 - 9 and the second FLR value based on packets 10 - 19, and so on.

Step 8 **end** or **commit**

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-pb)# 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 SLA Statistics Measurement in a Profile

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

Before you begin

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

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

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

For more information about frequency and time synchronization, refer to the *Configuring Frequency Synchronization on the Cisco ASR 9000 Series Router* and the *Configuring PTP on the Cisco ASR 9000 Series Router* modules in the *Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide*.

Restrictions

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

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

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 | one-way-loss-ds | one-way-loss-sd**}

SUMMARY STEPS

1.
2. **aggregate** {**bins** *count* **width** *width* | **none**}
3. **buckets size** *number* {**per-probe** | **probes}**
4. **buckets archive** *number*

5. **end** or **commit**

DETAILED STEPS

Step 1 **Example:**

```
RP/0/RP0/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/RP0/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. For delay measurements and data loss measurements, the default is that all values are aggregated into 1 bin. For synthetic loss measurements, the default is aggregation disabled.

- For delay measurements, a width between 1-10000, in milliseconds, must be specified if the number of bins is at least 2.
- For jitter measurements, a width between 1-10000, in milliseconds, must be specified if the number of bins is at least 3.
- For data loss measurements, a width between 1-100, in percentage points, must be specified if the number of bins is at least 2.
- For synthetic loss measurements, a width between 1-100, in percentage points, must be specified if the number of bins is at least 2.

Step 3 **buckets size** *number* {**per-probe** | **probes**}

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# buckets size 100 per-probe
```

Configures the size of the buckets in which statistics are collected.

Step 4 **buckets archive** *number*

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# buckets archive 50
```

Configures the number of buckets to store in memory.

Step 5 **end** or **commit**

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit
```

Saves configuration changes.

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

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

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

Configuring an SLA Operation

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

SUMMARY STEPS

1. **interface** [**FastEthernet**
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. **end** or **commit**

DETAILED STEPS

Step 1 **interface** [**FastEthernet**

Example:

```
RP/0/RP0/CPU0:router(config-if)# interface gigabitethernet 0/1/0/1
```

Physical interface or virtual interface.

Note
- Use the **show interfaces** command to see a list of all interfaces currently configured on the router.

For more information about the syntax for the router, use the question mark (**?**) online help function.

Step 2 **ethernet cfm**

Example:

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

Enters interface CFM configuration mode.

Step 3 **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*

Example:

```
RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1
```

Creates a MEP on an interface and enters interface CFM MEP configuration mode.

Step 4 **sla operation profile** *profile-name* **target** {**mep-id** *id* | **mac-address** *mac-address*}

Example:

```
RP/0/RP0/CPU0:router(config-if-cfm-mep)# sla operation profile Profile_1 target mac-address
01:23:45:67:89:ab
```

Creates an operation instance from a MEP to a specified destination.

Step 5 **end** or **commit**

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit
```

Saves configuration changes.

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

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

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

Configuring an On-Demand SLA Operation

The Cisco CRS 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

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.
- 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, inlcuding these 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.
  - CFM synthetic loss measurement—One-way FLR in both directions 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.

Note When the **once** keyword is used for 'send burst' ('send burst once' rather than 'send burst every'), it stops the collection of statistics with the packets that cross probe boundaries.

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:

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**] `RP/0/RP0/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`	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.

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 0x***HHHHHHHH* | **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**] `RP/0/RP0/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`	Configures an on-demand Ethernet SLA operation for CFM loopback. The example shows a minimum configuration, but specifies the option of a minimum packet size, and 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 test pattern of 0's for padding. • Use default class of service (CoS) for the egress interface. • Measure all statistics. • Aggregate statistics into one bin. • Schedule now. • Display results on the console.

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

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

ethernet sla on-demand operation type cfm-synthetic-loss-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*} [**synthetic loss calculation packets** *number*] [**statistics measure** {**one-way-loss-ds** | **one-way-loss-sd**}][**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**] `RP/0/RP0/CPU0:router# ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mac-address 2.3.4`	Configures an on-demand Ethernet SLA operation for CFM synthetic loss measurement. The example shows a minimum configuration, that specifies the local domain and source interface and target MEP.

Verifying SLA Configuration

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

show ethernet sla configuration-errors [**domain** *domain-name*] [**interface** *interface-path-id*] [**profile** *profile-name*]	Displays information about errors that are preventing configured SLA operations from becoming active, as well as any warnings that have occurred.
show ethernet sla operations [**detail**] [**domain** *domain-name*] [**interface** *interface-path-id*] [**profile** *profile-name*]	Displays information about configured SLA operations.

Configuring UDLD

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **Example:**

```
RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **interface** [**GigabitEthernet | TenGigE**] *interface-path-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface
TenGigE 0/1/0/0
```

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

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

Step 3 **ethernet udld**

Example:

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

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

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

Example:

```
RP/0/RP0/CPU0:router(config-if-udld)# mode normal
```

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

Step 5 **message-time**

Example:

```
RP/0/RP0/CPU0:router(config-if-udld)# message-time 70
```

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

Step 6 **logging disable**

Example:

```
RP/0/RP0/CPU0:router(config-if-udld)# loggig disable
```

(Optional) This command suppresses the operational UDLD syslog messages.

Step 7 **end**

Example:

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

Ends the configuration session and exits to the EXEC mode.

Configuring Ethernet Data Plane Loopback

Perform these steps to configure Ethernet Data Plane Loopback.

Note Configuring or permitting Ethernet Data Plane Loopback is not the same as starting an actual loopback session.

SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet | TenGigE**] *interface-path-id*
3. **ethernet loopback permit {external | internal}**
4. **end**
5. or
6. **commit**

DETAILED STEPS

Step 1 **configure**

Example:

```
RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **interface** [**GigabitEthernet | TenGigE**] *interface-path-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface
TenGigE 0/1/0/0
```

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

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

Step 3 **ethernet loopback permit {external | internal}**

Example:

```
RP/0/RP0/CPU0:router(config-if-srv)# ethernet loopback permit external
```

Configures ethernet loopback externally or internally on an interface. External loopback allows loopback of traffic from wire. Internal loopback allows loopback of traffic from the bridge domain.

Step 4 **end**

Step 5 or

Step 6 **commit**

Example:

```
RP/0/RP0/CPU0:router(config-if-srv)# 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.

Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

Configuration Examples for EOAM Interfaces

This section provides the following configuration examples:

Configuring an Ethernet OAM Profile Globally: Example

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

```
configure terminal
 ethernet oam profile Profile_1
  link-monitor
   symbol-period window 60000
   symbol-period threshold 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

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

```
configure terminal
 interface TenGigE 0/1/0/0
  ethernet oam
   link-monitor
    symbol-period window 60000
    symbol-period threshold 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

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

```
configure terminal
 ethernet oam profile Profile_1
  mode passive
  action dying-gasp disable
  action critical-event disable
  action discovery-timeout disable
  action session-up disable
  action session-down disable
  action capabilities-conflict disable
  action wiring-conflict disable
  action remote-loopback disable
  action uni-directional link-fault error-disable-interface
  commit
```

```
configure terminal
 interface TenGigE 0/1/0/0
  ethernet oam
   profile Profile_1
    mode active
    action dying-gasp log
    action critical-event log
    action discovery-timeout log
    action session-up log
    action session-down log
    action capabilities-conflict log
    action wiring-conflict log
    action remote-loopback log
    action uni-directional link-fault log
    uni-directional link-fault detection
    commit
```

Clearing Ethernet OAM Statistics on an Interface: Example

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

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

Enabling SNMP Server Traps on a Router: Example

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

```
configure terminal
  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

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

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

Ethernet CFM Service Configuration: Example

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

Continuity Check for an Ethernet CFM Service Configuration: Example

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

```
   continuity-check archive hold-time 100
   continuity-check loss auto-traceroute
```

```
   continuity-check interval 100ms loss-threshold 10
   commit
```

MIP Creation for an Ethernet CFM Service Configuration: Example

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

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

Cross-check for an Ethernet CFM Service Configuration: Example

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

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

Other Ethernet CFM Service Parameter Configuration: Example

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

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

MEP Configuration: Example

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

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

Ethernet CFM Show Command: Examples

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

Example 1

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

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

Domain/Level         Service              Interface          Type   ID   MAC
-------------------- -------------------- ------------------ ------ ---- --------
fig/5                bay                  Gi0/10/0/12.23456 Dn MEP    2 44:55:66
fig/5                bay                  Gi0/0/1/0.1        MIP         55:66:77
fred/3               barney               Gi0/1/0/0.1        Up MEP    5 66:77:88!
```

Example 2

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

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

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

 * An Up MEP is configured for this domain on interface 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

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

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

 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down

Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1.234 (Up)       Up     0/0   N  A       L7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 Gi0/1/0/0.234 (Up)       Up     3/2   Y  RPC     L6
Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1.234 (Up)       Up     0/0   N  A

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

Example 4

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

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

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

```
Domain fred (level 7), Service barney
Up MEP on GigabitEthernet0/1/0/0.234, MEP-ID 2
================================================================================
St    ID MAC address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
 >     1 0011.2233.4455 Up      00:00:01         1234      0     0     0
R>     4 4455.6677.8899 Up      1d 03:04         3456      0   234     0
L      2 1122.3344.5566 Up      3w 1d 6h         3254      0     0  3254
C      2 7788.9900.1122 Test    00:13            2345      6    20  2345
X      3 2233.4455.6677 Up      00:23              30      0     0    30
I      3 3344.5566.7788 Down    00:34           12345      0   300  1234
V      3 8899.0011.2233 Blocked 00:35              45      0     0    45
 T     5 5566.7788.9900         00:56              20      0     0     0
M      6                                            0      0     0     0
U>     7 6677.8899.0011 Up      00:02             456      0     0     0

Domain fred (level 7), Service fig
Down MEP on GigabitEthernet0/10/0/12.123, 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
```

Example 5

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

```
RP/0/RP0/CPU0:router#  show ethernet cfm peer meps detail
Domain dom3 (level 5), Service ser3
Down MEP on 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
       Remote Defect received:    5
    Wrong Level:                 0
       Cross-connect W(wrong MAID): 0
       Wrong Interval:              5
       Loop (our MAC received):     0
       Config (our ID received):    0
     Last CCM received 00:00:08 ago:
       Level: 2, Version: 0, Interval: 10s
       Sequence number: 1, MEP-ID: 600
       MAID: DNS-like: dom5, String: ser5
       Chassis ID: Local: ios; Management address: 'Not specified'
       Port status: Up, Interface status: Down

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

```
      Wrong Interval:              2
      Loop (our MAC received):     0
      Config (our ID received):    0
    Last CCM received 00:15:49 ago:
      Level: 2, Version: 0, Interval: 10s
      Sequence number: 1, MEP-ID: 600
      MAID: DNS-like: dom5, String: ser5
      Chassis ID: Local: ios; Management address: 'Not specified'
      Port status: Up, Interface status: Down
```

AIS for CFM Configuration: Examples

Example 1

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

Example 2

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

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

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface gigabitethernet 0/1/0/2
RP/0/RP0/CPU0:router(config-if)# ethernet cfm
RP/0/0RP0RSP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7
```

AIS for CFM Show Commands: Examples

This section includes the following examples:

show ethernet cfm interfaces ais Command: Example

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

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

Defects (from at least one peer MEP):
 A - AIS received               I - Wrong interval
 R - Remote Defect received     V - Wrong Level
 L - Loop (our MAC received)    T - Timed out (archived)
 C - Config (our ID received)   M - Missing (cross-check)
 X - Cross-connect (wrong MAID) U - Unexpected (cross-check)
 P - Peer port down             D - Local port down

                               Trigger                   Transmission
                         AIS  ---------   Via    ----------------------------
Interface (State)        Dir  L Defects  Levels  L Int Last started  Packets
------------------------ ---  - -------  ------- - --- ------------ --------
Gi0/1/0/0.234 (Up)       Dn   5 RPC      6       7 1s  01:32:56 ago     5576
Gi0/1/0/0.567 (Up)       Up   0 M        2,3     5 1s  00:16:23 ago      983
Gi0/1/0/1.1 (Dn)         Up     D                7 60s 01:02:44 ago     3764
Gi0/1/0/2 (Up)           Dn   0 RX       1!
```

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/RP0/CPU0:router# show ethernet cfm local meps

 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down

Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1.234 (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/RSP0RP0/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)

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
```

Example 4: Detail

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

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

Domain foo (level 6), Service bar
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

This example shows how to enable EFD:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 down-meps
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# efd
```

This example shows how to enable EFD logging:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 down-meps
RP/0/RP0/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

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

```
RP/0/RP0/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/RP0/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

These 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

These 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:

Note The total length of a burst (packet count multiplied by the interval) must not exceed 1 minute.

```
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
```

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:

```
configure
 ethernet sla
  profile  Prof1 type cfm-loopback
   probe
    send burst once packet count 50 interval 1 second
    packet size 9000 test pattern hex 0xabcdabcd
    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:

```
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

These 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:

```
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:

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

```
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

These 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:

```
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:

```
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:

```
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

This figure 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:

```
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
```

Ongoing Ethernet SLA Operation Configuration: Example

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

```
 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

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

Example 1

This 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/RP0RSP0/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

This 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/RP0RSP0/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 Y.1731 SLM Configuration: Examples

These examples show how to configure the synthetic loss measurement statistics.

Example 1

This example shows the default configuration for Y.1731 SLM:

```
ethernet sla
  profile sl1 type cfm-synthetic-loss-measurement
    statistic measure one-way-loss-sd
    statistic measure one-way-loss-ds
```

Example 2

This example configures a profile named “Sl2” for synthetic loss measurements. with the parameters to configure the probe and SLM statistics:

```
ethernet sla
  profile sl2 type cfm-synthetic-loss-measurement
    probe
      send burst every 5 seconds packet count
                     100 interval 50 milliseconds
      packet size 400 test pattern hex 0xABDC1234
      synthetic loss calculation packets 200
    schedule every 1 hours for 1 minute
    statistic measure one-way-loss-sd
    statistic measure one-way-loss-ds
      aggregate bins 3 width 30
      bucket size 24 probes
```

Ethernet SLA Show Commands: Examples

These examples show how to display information about configured SLA operations:

show ethernet sla operations Command: Example 1

```
RP/0/RP0/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/RP0/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
```

These 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/RP0/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/RP0/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

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

```
RP/0/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
```

show ethernet sla statistics profile Command: Example 6

These examples show how to display statistics for synthetic loss measurement in detail:

```
RP/0/RSP0/CPU0:router#show ethernet sla statistics profile sl2 statistic one-way-loss-sd
detail

Source: Interface GigabitEthernet0/0/0/0, Domain dom1
Destination: Target MAC Address 0002.0003.0005
================================================================================
Profile 'sl1', packet type 'cfm-synthetic-loss-measurement'
Scheduled to run every 1hr first at 00:50:00 UTC for 1min
Frame Loss Ratio calculated every 10s

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

Bucket started at 04:50:00 PDT Thu 15 September 2012 lasting 1hr
    Pkts sent: 1200; Lost: 27 (2.25%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Min: 0.00%, occurred at 04:50:50 PDT Thu 15 September 2011
    Max: 5.50%, occurred at 04:50:20 PDT Thu 15 September 2011
    Mean: 2.08%; StdDev: 1.99%; Overall: 2.08%

    Measurements:
    Time                     Result  Notes
    ----------  -------------------  ----------
    04:50:00.0    1.50% (3 of 200)
    04:50:10.0    2.00% (4 of 200)
    04:50:20.0    5.50% (11 of 200)
    04:50:30.0    3.00% (6 of 200)
    04:50:40.0    0.50% (1 of 200)
    04:50:50.0    0.00% (0 of 200)
```

In the example 6, the description of the statistics that indicate the lost count and overall FLR are Lost: 27 (2.25%) and Overall: 2.08%. The lost count means that 27 SLMs were lost out of 1200, but it might not be possible to determine in which direction they were lost. The overall FLR reports the overall loss in the Source to Destination direction.

show ethernet sla statistics profile Command: Example 7

```
RP/0/RSP0/CPU0:ios#show ethernet sla statistics profile sl2 statistic one-way-loss-ds detail
Source: Interface GigabitEthernet0/0/0/0, Domain dom1
Destination: Target MAC Address 0002.0003.0005
================================================================================
Profile 'sl2', packet type 'cfm-synthetic-loss-measurement'
Scheduled to run every 1hr first at 00:55:00 UTC for 1min
Frame Loss Ratio calculated every 10s

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

Bucket started at 04:55:00 PDT Thu 15 September 2012 lasting 1 day
    Pkts sent: 28800; Lost: 14691 (51.01%); Corrupt: 0 (0.0%);
                      Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Min: 10.00%, occurred at 04:55:00 PDT Thu 15 September 2011
    Max: 68.80%, occurred at 06:55:00 PDT Thu 15 September 2011
    Mean: 52.5%; StdDev: 0.00%; Overall: 51.00%

    Bins:
    Range             Count  Cum. Count    Mean
    -----------  ----------  ----------  ------
      0 to  30%  20 (13.9%)  20 (13.9%)  21.00%
     30 to  60%  71 (49.3%)  91 (63.2%)  57.90%
     60 to 100%  49 (34.0%) 144 (100.0%) 62.00%
```

Configuration Examples for Ethernet Data Plane Loopback

This example shows how to configure Ethernet Data Plane Loopback:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/1
RP/0/RSP0/CPU0:router(config-if-srv)# ethernet loopback permit external
```

This example shows how to start an Ethernet Data Plane Loopback:

```
RP/0/RSP0/CPU0:router# ethernet loopback start local interface gigabitEthernet
 0/1/0/1
external

[source mac-address <addr>]
 [destination mac-address <addr>]
 [ether-type <etype>]
 [{dot1q <vlan-ids> [second-dot1q <vlan-ids>] |
  dot1ad <vlan-ids> [dot1q <vlan-ids>]}]
 [cos <cos>]
 [llc-oui <oui>]
 [timeout {<length> | none}]
```

This example shows how to stop an Ethernet Data Plane Loopback session:

```
RP/0/RSP0/CPU0:router# ethernet loopback stop local interface <name> id <id>
```

This example shows how to extend an Ethernet Data Plane Loopback session:

```
RP/0/RSP0/CPU0:router# ethernet loopback extend local interface <name> id <id>
 length
<length>
```

Verification

- Use the **show ethernet loopback permitted** command to display all the permitted interfaces which run Ethernet Data Plane Loopback sessions:

```
RP/0/RSP0/CPU0:router# show ethernet loopback permitted
Interface Direction
------------------------------------------------
GigabitEthernet0/0/0/0 External
GigabitEthernet0/0/0/1.100 Internal
TenGigE0/1/0/0.200 External, Internal
```

- Use the **show ethernet loopback active** command to view active sessions:

```
RP/0/RSP0/CPU0:router# show ethernet loopback active interface
TenGigE0/1/0/0.200

Local: TenGigE0/1/0/0.200, ID 1
============================================
Direction: Internal
Time out: 2 hours
Time left: 00:01:17
Status: Active
Filters:
 Dot1ad: 100-200
 Dot1q: Any
 Source MAC Address: aaaa.bbbb.cccc
 Destination MAC Address: Any
 Ethertype: 0x8902
 Class of Service: Any
 LLC-OUI: Any
Local: TenGigE0/1/0/0.200, ID 2
============================================
Direction: External
Time out: 10 minutes
Time left: 00:00:00
Status: Stopping
Filters:
 Dot1q: 500
 Second-dot1q: 200
 Source MAC Address: Any
 Destination MAC Address: Any
 Ethertype: Any
 Class of Service: 4
 LLC-OUI: Any
```

For each loopback session listed, this information is displayed:

- Header containing the Interface name and session ID, which uniquely identify the local loopback session,
- Direction which specifies the direction of the loopback,

- Time out – the time out period specified when the loopback was started,
- Time left – the amount of time left until the loopback session is automatically stopped,
- Status – the status of the loopback session,
- Filters – details of the filters specified when the loopback session was started. Similar to the start CLI, only the filters supported by the platform are displayed.

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

CHAPTER 7

Configuring Link Bundling

This module describes the configuration of link bundle interfaces on the Cisco CRS-1 Router.

A link bundle is a group of one or more ports that are aggregated together and treated as a single link.

Each bundle has a single MAC and shares a single Layer 3 configuration set, such as IP address, ACL, Quality of Service (QoS), and so on.

Note Link bundles do not have a one-to-one modular services card association. Member links can terminate on different cards.

Feature History for Configuring Link Bundling

Release	Modification
Release 3.2	This feature was introduced on the Cisco CRS-1 Router.

Release 3.3.0	This feature was updated as follows: • To support the 1:N redundancy feature, users can configure the minimum number of active links using the **bundle minimum-active links** command. • To support the 1:N redundancy feature, users can configure the minimum bandwidth in kbps using the **bundle minimum-active links** command. • Support was added for VLAN subinterfaces on Ethernet link bundles. • Output for **show bundle bundle-Ether** command and **show bundle bundle-POS** command was modified. • The **reasons** keyword was added to the **show bundle bundle-Ether** command and the **show bundle bundle-POS** command. • The **bundle id** command was changed from **bundle-id**. • BFD over bundled VLANs using static routes.
Release 3.4.0	The configuration procedures in this module were modified with enhancements.
Release 3.7.0	Note was added, specifying that link bundling is supported on the multishelf Cisco CRS-1 Router.
Release 3.8.0	This feature was updated as follows: • The **reasons** keyword was removed from the **show bundle bundle-Ether** command and the **show bundle bundle-POS** command. Now, if a port is in a state other than the distributing state, the output of both commands displays the reason. • The **hot-standby** keyword was added to the **bundle maximum-active links** command. • The **lacp fast-switchover** command was added.
Release 3.8.4	Bundle member links are put into new err-disable link interface status and admin-down protocol state when a bundle interface is shut down.
Release 3.9.0	Support for super short LACP was added. Support for load balancing was added. Support for a maximum of 64 member links per bundle was added.

Release 4.0.0	Support for the following physical layer interface modules (PLIMs) was added: • 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (14X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (20X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)
Release 4.0.1	Support for the following PLIMs was added: • 1-Port 100-Gigabit Ethernet PLIM (1X100GBE) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP) • 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)
Release 5.1.1	Support for Multichassis Link Aggregation on IPv6 traffic was included.
Release 5.1.2	Support for mixed speed member links in a bundle interface was included.
Release 6.0.0	Bundle scale support increases to 1600 bundles on A9K-RSP880-SE, A99-RP2-SE and the third generation of ASR 9000 Ethernet line card.
Release 6.2.2	Support for Layer 3 Multicast traffic over mixed speed bundles was added.

- Prerequisites for Configuring Link Bundling, on page 215
- Information About Configuring Link Bundling, on page 216
- How to Configure Link Bundling, on page 224
- Configuration Examples for Link Bundling, on page 246

Prerequisites for Configuring Link Bundling

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.

The prerequisites for link bundling depend on the platform on which you are configuring this feature. This section includes the following information:

Prerequisites for Configuring Link Bundling on a Cisco CRS-1 Router

Before configuring link bundling on a Cisco IOS XR Router, be sure that the following tasks and conditions are met:

- You know which links should be included in the bundle you are configuring.
- If you are configuring an Ethernet link bundle, you have at least one of the following Ethernet cards installed in the router:
    - 1-port 10-Gigabit Ethernet SPA (LAN and WAN-PHY)
    - 4-Port 10-Gigabit Ethernet Tunable WDMPHY Physical Layer Interface Module (PLIM)
    - 4-Port 10-Gigabit Ethernet PLIM
    - 5-Port Gigabit Ethernet SPA
    - 8-Port Gigabit Ethernet SPA (versions 1 and 2)
    - 8-Port 10-Gigabit Ethernet PLIM
    - 10-Port Gigabit Ethernet SPA
    - 42-Port Gigabit Ethernet PLIM
    - 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
    - 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
    - 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
    - 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
    - 1-Port 100-Gigabit Ethernet PLIM
- If you are configuring a POS link bundle, you have a POS line card or SPA installed in a router that is running Cisco IOS XR software.

Note For more information about physical interfaces, PLIMs, and modular services cards, refer to the *Cisco CRS-1 Carrier Routing System 8-Slot Line Card Chassis System Description*.

Information About Configuring Link Bundling

To configure link bundling, you must understand the following concepts:

Link Bundling Overview

The Link Bundling feature allows you to group multiple point-to-point links together into one logical link and provide higher bidirectional bandwidth, redundancy, and load balancing between two routers. A virtual interface is assigned to the bundled link. The component links can be dynamically added and deleted from the virtual interface.

The virtual interface is treated as a single interface on which one can configure an IP address and other software features used by the link bundle. Packets sent to the link bundle are forwarded to one of the links in the bundle.

The advantages of link bundles are as follows:

- Multiple links can span several line cards and SPAs to form a single interface. Thus, the failure of a single link does not cause a loss of connectivity.
- Bundled interfaces increase bandwidth availability, because traffic is forwarded over all available members of the bundle. Therefore, traffic can move onto another link if one of the links within a bundle fails. You can add or remove bandwidth without interrupting packet flow. For example, you can upgrade from an OC-48c PLIM modular services card to an OC-192 PLIM modular services card without interrupting traffic.

All links within a bundle must be of the same type. For example, a bundle can contain all Ethernet interfaces, or it can contain all POS interfaces, but it cannot contain Ethernet and POS interfaces at the same time.

Cisco IOS XR software supports the following methods of forming bundles of Ethernet and POS interfaces:

- IEEE 802.3ad—Standard technology that employs a Link Aggregation Control Protocol (LACP) to ensure that all the member links in a bundle are compatible. Links that are incompatible or have failed are automatically removed from a bundle.
- Ether Channel or POS Channel—Cisco proprietary technology that allows the user to configure links to join a bundle, but has no mechanisms to check whether the links in a bundle are compatible. (EtherChannel applies to Ethernet interfaces, and POS Channel applies to POS interfaces.)

Features and Compatible Characteristics of Ethernet Link Bundles

This list describes the properties and limitations of ethernet link bundles:

- Any type of Ethernet interfaces can be bundled, with or without the use of LACP (Link Aggregation Control Protocol).
- Bundle membership can span across several line cards that are installed in a single router or multiple routers in the case of MC-LAG.
- With mixed speed bundles, the total weight of a bundle is greater than the number of members in the bundle, as the weight represents the smallest active number bandwidth.
- The weight of each bundle member is the ratio of its bandwidth to the lowest bandwidth member. Total weight of the bundle is the sum of weights or relative bandwidth of each bundle member. Since the weight for a bundle member is greater than or equal to 1 and less than or equal to 10, the total member of links in a bundle is less than 64 in mixed bundle case.
- Mixed speed bundles are supported in MC-LAG.
- Physical layer and link layer configuration are performed on individual member links of a bundle.
- Configuration of network layer protocols and higher layer applications is performed on the bundle itself.
- IPv4 and IPv6 addressing is supported on ethernet link bundles.
- A bundle can be administratively enabled or disabled. Beginning in Cisco IOS XR Release 3.8.4, when you shut down a bundle interface, the member links are put into err-disable link interface status and

admin-down line protocol state. You can show the status of a bundle interface and its members using the **show interfaces** command.

- Each individual link within a bundle can be administratively enabled or disabled.
- The MAC address that is set on the bundle becomes the MAC address of the links within that bundle.
- MAC address is set on the bundle the address of the
- If a MAC address is not set on the bundle, the bundle MAC address is obtained from a pool of pre-assigned MAC addresses stored in EEPROM of the chassis midplane.
- Each link within a bundle can be configured to allow different keepalive periods on different members
- Load balancing (the distribution of data between member links) is done by flow instead of by packet.
- Upper layer protocols, such as routing updates and hellos, are sent over any member link of an interface bundle.
- All links within a single bundle must terminate on the same two systems. Both systems must be directly connected except in the case of MC-LAG.
- Bundled interfaces are point-to-point.
- A bundle can contain physical links only. Tunnels and VLAN subinterfaces cannot be bundle members. However, you can create VLANs as subinterfaces of bundles.
- An IPv4 address configuration on link bundles is identical to an IPv4 address configuration on regular interfaces.
- Multicast traffic is load balanced over the members of a bundle. For a given flow, the control plane selects the member , and all traffic for that flow is sent over that member. The system supports Layer 3 Multicast traffic over mixed speed bundles.

Characteristics of CRS-1 Series Router Link Bundles

The following list describes additional properties and limitations of link bundles that are specific to CRS-1 Series:

- Link bundling is supported on all multishelf Cisco CRS-1 Routers.
- A bundle can contain all Ethernet interfaces or all POS interfaces, but not a mix of Ethernet and POS interfaces.
- A single bundle supports a maximum of 64 physical links. If you add more than 64 links to a bundle, only 64 of the links function, and the remaining links are automatically disabled.
- A Cisco CRS Router supports a maximum of 64 bundles.
- Ethernet and POS link bundles are created in the same way as Ethernet channels and POS channels, where the user enters the same configuration on both end systems.
- For Ethernet link bundles, links within a single bundle should have the same speed.
- For POS link bundles, the links within a single bundle can have varying speeds. The fastest link can be set to a maximum speed that is four times greater than the slowest link.

- HDLC is the only supported encapsulation type for POS link bundles in Cisco IOS XR software. POS links that are configured with any other encapsulation type cannot join a bundle. Keep in mind that all POS link bundle members must be running HDLC for HDLC to work on a bundle.
- QoS is supported and is applied proportionally on each bundle member.
- Link layer protocols, such as CDP and HDLC keepalives, work independently on each link within a bundle.
- All links within a single bundle must be configured to run either POS Channel or 802.3ad. Mixed bundles are not supported.

Link Aggregation Through LACP

Aggregating interfaces on different modular services cards and on SPAs within the same services cards provides redundancy, allowing traffic to be quickly redirected to other member links when an interface or modular services card failure occurs.

The optional Link Aggregation Control Protocol (LACP) is defined in the IEEE 802 standard. LACP communicates between two directly connected systems (or peers) to verify the compatibility of bundle members. The peer can be either another router or a switch. LACP monitors the operational state of link bundles to ensure the following:

- All links terminate on the same two systems.
- Both systems consider the links to be part of the same bundle.
- All links have the appropriate settings on the peer.

LACP transmits frames containing the local port state and the local view of the partner system's state. These frames are analyzed to ensure both systems are in agreement.

IEEE 802.3ad Standard

The IEEE 802.3ad standard typically defines a method of forming Ethernet link bundles. In Cisco IOS XR software, the IEEE 802.3ad standard is used on both Ethernet and POS link bundles.

For each link configured as bundle member, the following information is exchanged between the systems that host each end of the link bundle:

- A globally unique local system identifier
- An identifier (operational key) for the bundle of which the link is a member
- An identifier (port ID) for the link
- The current aggregation status of the link

This information is used to form the link aggregation group identifier (LAG ID). Links that share a common LAG ID can be aggregated. Individual links have unique LAG IDs.

The system identifier distinguishes one router from another, and its uniqueness is guaranteed through the use of a MAC address from the system. The bundle and link identifiers have significance only to the router assigning them, which must guarantee that no two links have the same identifier, and that no two bundles have the same identifier.

The information from the peer system is combined with the information from the local system to determine the compatibility of the links configured to be members of a bundle.

The MAC address of the first link attached to a bundle becomes the MAC address of the bundle itself. The bundle uses this MAC address until that link (the first link attached to the bundle) is detached from the bundle, or until the user configures a different MAC address. The bundle MAC address is used by all member links when passing bundle traffic. Any unicast or multicast addresses set on the bundle are also set on all the member links.

Note We recommend that you avoid modifying the MAC address, because changes in the MAC address can affect packet forwarding.

Non Revertive Behavior for LACP Bundle Interface

In LACP, by default, a higher priority port would become the active port after it becomes operational again. To avoid this reversion, you can run the **lacp non-revertive** command. This configures the lower priority port to continue as the active port even after the higher priority port is capable of being operational. This avoids the traffic disruption that may happen in putting the currently active but lower priority port into standby and diverting traffic through the higher priority port that is now capable of being operational.

ICCP Based Service Multihoming

In the case of ICCP based Service Multihoming (ICCP-SM), the CE device uses two independent bundle interfaces to connect to the PoAs. Although bundle interfaces are used, they are not aggregated across the two chassis, and mLACP is not involved in the communication. The CE device configures the bundle interfaces in such a manner that all VLANs are allowed on both bundles. You can manually configure the PoAs to distribute the VLANs across the two bundles in order that individual VLANs are active(forwarding) on one bundle or PoA, and standby (blocked) on the other. The CE device initially floods a traffic flow on both bundles and learns the MAC address on the interface where it receives the response.

With ICCP-SM, you are not limited to a dual homed device. The access links can connect to a dual homed network (DHN) that are separate devices in the access network. The two bundles on the DHD or the DHN must be in a bridge domain so that L2 learning selects the link with the active set of VLANs.

Figure 11: ICCP Based Service Multihoming

If a bundle interface between the CE and the PoA fails, ICCP-SM on the PoA with the failed bundle communicates through ICCP to the other PoA's ICCP-SM. This activates the standby VLANs on the remaining bundle. A MAC flush is sent to the CE so that packets destined to hosts on the failed bundle are again flooded, in order to be learned on the newly activated bundle. The MAC flush is required because it is possible that the bundle interface failure is not detected by the CE.

In ICCP Based Service Multihoming, the total set of VLANs are split into a primary set and a secondary set and are configured on each PoA such that the primary set on one PoA is configured as secondary on the other. On each PoA, the VLANs are associated with ACs. If the VLANs are primary on a PoA and there are no faults, the associated ACs are set to forwarding. If the VLANs are secondary on a PoA, the associated ACs are blocked. ICCP-SM is only supported in VPLS cores.

Advantages of Pseudo mLACP:

Pseudo mLACP has these three major advantages over mLACP:

- Pseudo mLACP can support a Dual Homed Network (DHN), while mLACP can only support a Dual Homed Device (DHD).
- Pseudo mLACP supports per-VLAN active/active redundancy without any load-balancing requirements on the CE.
- Pseudo mLACP does not require LACP support from the DHD, or DHN. It is independent of the access redundancy mechanism; therefore, it provides a network based redundancy solution. It allows maximum flexibility for the PE-CE interoperability in terms of dual-homing redundancy and recovery.

LACP Short Period Time Intervals

As packets are exchanged across member links of a bundled interface, some member links may slow down or time-out and fail. LACP packets are exchanged periodically across these links to verify the stability and reliability of the links over which they pass. The configuration of short period time intervals, in which LACP packets are sent, enables faster detection and recovery from link failures.

Short period time intervals are configured as follows:

- In milliseconds
- In increments of 100 milliseconds
- In the range 100 to 1000 milliseconds
- The default is 1000 milliseconds (1 second)
- Up to 64 member links
- Up to 1280 packets per second (pps)

After 6 missed packets, the link is detached from the bundle.

When the short period time interval is *not* configured, LACP packets are transmitted over a member link every 30 seconds by default.

When the short period time interval is configured, LACP packets are transmitted over a member link once every 1000 milliseconds (1 second) by default. Optionally, both the transmit and receive intervals can be configured to less than 1000 milliseconds, independently or together, in increments of 100 milliseconds (100, 200, 300, and so on).

When you configure a custom LACP short period *transmit* interval at one end of a link, you must configure the same time period for the *receive* interval at the other end of the link.

Note You must always configure the *transmit* interval at both ends of the connection before you configure the *receive* interval at either end of the connection. Failure to configure the *transmit* interval at both ends first results in route flapping (a route going up and down continuously). When you remove a custom LACP short period, you must do it in reverse order. You must remove the *receive* intervals first and then the *transmit* intervals.

Load Balancing

Load balancing is a forwarding mechanism which distributes traffic over multiple links, based on Layer 3 routing information in the router. Per-flow load balancing is supported on all links in the bundle. This scheme achieves load sharing by allowing the router to distribute packets over one of the links in the bundle, that is determined through a hash calculation. The hash calculation is an algorithm for link selection based on certain parameters.

The standard hash calculation is a 3-tuple hashing, using the following parameters:

- IP source address
- IP destination address
- Router ID

7-tuple hashing can also be configured. based on Layer 3 and Layer 4 parameters:

- IP source address
- IP destination address
- Router ID
- Input interface
- IP protocol
- Layer 4 source port
- Layer 4 destination port

When per-flow load balancing and 3-tuple hashing is enabled, all packets for a certain source-destination pair will go through the same link, though there are multiple links available. Per-flow load balancing ensures that packets for a certain source-destination pair arrive in order.

Note For multicast traffic, ingress forwarding is based on the Fabric Multicast Group Identifier (FGID). Egress forwarding over the bundle is based on the bundle load balancing.

QoS and Link Bundling

On the Cisco CRS-1 Router, QoS is applied to the local instance of a bundle in the ingress direction. Each bundle is associated with a set of queues. QoS is applied to the various network layer protocols that are configured on the bundle. In the egress direction, QoS is applied on the bundle with a reference to the member links. QoS is applied based on the sum of the member bandwidths.

For complete information on configuring QoS on link bundles on the Cisco CRS-1 Router, refer to the *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco CRS Router* and the *Cisco IOS XR Modular Quality of Service Command Reference for the Cisco CRS Router*.

VLANs on an Ethernet Link Bundle

802.1Q VLAN subinterfaces can be configured on 802.3ad Ethernet link bundles. Keep the following information in mind when adding VLANs on an Ethernet link bundle:

- The maximum number of VLANs allowed per bundle is 128.
- The maximum number of bundled VLANs allowed per router is 4000.

Note The memory requirement for bundle VLANs is slightly higher than standard physical interfaces.

To create a VLAN subinterface on a bundle, include the VLAN subinterface instance with the **interface Bundle-Ether** command, as follows:

interface Bundle-Ether *interface-bundle-id.subinterface*

After you create a VLAN on an Ethernet link bundle, all VLAN subinterface configuration is supported on that link bundle.

VLAN subinterfaces can support multiple Layer 2 frame types and services, such as Ethernet Flow Points - EFPs) and Layer 3 services.

Link Bundle Configuration Overview

The following steps provide a general overview of the link bundle configuration. Keep in mind that a link must be cleared of all previous network layer configuration before it can be added to a bundle:

1. In global configuration mode, create a link bundle. To create an Ethernet link bundle, enter the **interface Bundle-Ether** command. To create a POS link bundle, enter the interface Bundle-POS command.
2. Assign an IP address and subnet mask to the virtual interface using the **ipv4 address** command.
3. Add interfaces to the bundle you created in Step 1 with the **bundle id** command in the interface configuration submode.

   You can add up to 64 links to a single bundle.
4. On a CRS-1 Series router, optionally implement 1:1 link protection for the bundle by setting the **bundle maximum-active links** command to 1. Performing this configuration causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. (The link priority is based on the value of the **bundle port-priority** command.) If the active link fails, the standby link immediately becomes the active link.

Note A link is configured as a member of a bundle from the interface configuration submode for that link.

Nonstop Forwarding During RP Switchover

Cisco IOS XR software supports nonstop forwarding during switchover between active and standby paired RP cards. Nonstop forwarding ensures that there is no change in the state of the link bundles when a switchover occurs.

For example, if an active RP fails, the standby RP becomes operational. The configuration, node state, and checkpoint data of the failed RP are replicated to the standby RP. The bundled interfaces will all be present when the standby RP becomes the active RP.

Note You do not need to configure anything to guarantee that the standby interface configurations are maintained.

Link Switchover

By default, a maximum of 64 links in a bundle can actively carry traffic on a Cisco CRS-1 Router, If one member link in a bundle fails, traffic is redirected to the remaining operational member links.

On a Cisco CRS-1 Router, you can optionally implement 1:1 link protection for a bundle by setting the **bundle maximum-active links** command to 1. By doing so, you designate one active link and one or more dedicated standby links. If the active link fails, a switchover occurs and a standby link immediately becomes active, thereby ensuring uninterrupted traffic.

If the active and standby links are running LACP, you can choose between an IEEE standard-based switchover (the default) or a faster proprietary optimized switchover. If the active and standby links are not running LACP, the proprietary optimized switchover option is used.

Regardless of the type of switchover you are using, you can disable the wait-while timer, which expedites the state negotiations of the standby link and causes a faster switchover from a failed active link to the standby link.

To do so, you can use the **lacp fast-switchover** command.

How to Configure Link Bundling

This section contains the following procedures:

Configuring Ethernet Link Bundles

This section describes how to configure an Ethernet link bundle.

Note In order for an Ethernet bundle to be active, you must perform the same configuration on both connection endpoints of the bundle.

SUMMARY STEPS

1. **configure**

2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface** {**GigabitEthernet** | **TenGigE**} interface-path-id
10. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
11. **bundle port-priority** *priority*
12. **no shutdown**
13. **exit**
14. **bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **no shutdown exit**
15. **end** or **commit**
16. **exit**
17. **exit**
18. Perform Step 1 through Step 15 on the remote end of the connection.
19. **show bundle Bundle-Ether** *bundle-id*
20. **show lacp bundle Bundle-Ether** *bundle-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3`	Creates a new Ethernet link bundle with the specified bundle-id. The range is 1 to 65535. This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface specific configuration commands are entered. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:**	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization. **Note** • The priority of the active and standby links is based on the value of the **bundle port-priority** command.
Step 7	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration submode for the Ethernet link bundle.
Step 9	**interface** {**GigabitEthernet** | **TenGigE**} interface-path-id **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0`	Enters interface configuration mode for the specified interface. Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the *rack*/*slot*/*module* format.
Step 10	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds the link to the specified bundle. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the link to the bundle without LACP support, include the optional **mode on** keywords with the command string.

	Command or Action	Purpose
		Note • If you do not specify the **mode** keyword, the default mode is **on (**LACP is not run over the port).
Step 11	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.
Step 12	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration submode for the Ethernet interface.

Step 14	**bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **no shutdown exit**
	Example:
	`RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/1`
	`RP/0/RP0/CPU0:router(config-if)# bundle id 3`
	`RP/0/RP0/CPU0:router(config-if)# bundle port-priority 2`
	`RP/0/RP0/CPU0:router(config-if)# no shutdown`
	`RP/0/RP0/CPU0:router(config-if)# exit`
	`RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/3`
	`RP/0/RP0/CPU0:router(config-if)# bundle id 3`
	`RP/0/RP0/CPU0:router(config-if)# no shutdown`
	`RP/0/RP0/CPU0:router(config-if)# exit`
		(Optional) Repeat Step 8 through Step 11 to add more links to the bundle.

Step 15	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# end` or	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before`

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# commit`	`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.
Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration mode.
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 18	Perform Step 1 through Step 15 on the remote end of the connection.	Brings up the other end of the link bundle.
Step 19	**show bundle Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3`	(Optional) Shows information about the specified Ethernet link bundle.
Step 20	**show lacp bundle Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show lacp bundle Bundle-Ether 3`	(Optional) Shows detailed information about LACP ports and their peers.

Configuring EFP Load Balancing on an Ethernet Link Bundle

This section describes how to configure Ethernet flow point (EFP) Load Balancing on an Ethernet link bundle.

By default, Ethernet flow point (EFP) load balancing is enabled. However, the user can choose to configure all egressing traffic on the fixed members of a bundle to flow through the same physical member link. This configuration is available only on an Ethernet Bundle subinterface with Layer 2 transport (**l2transport**) enabled.

Note If the active members of the bundle change, the traffic for the bundle may get mapped to a different physical link that has a hash value that matches the configured value.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id* **l2transport**
3. **bundle load-balance hash** *hash-value* [**auto**]
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **l2transport** **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3 l2transport`	Creates a new Ethernet link bundle with the specified *bundle-id* and with Layer 2 transport enabled. The range is 1 to 65535.

Step 3	**bundle load-balance hash** *hash-value* [**auto**]
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1`
	or
	`RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto`
		Configures all egressing traffic on the fixed members of a bundle to flow through the same physical member link.
		• *hash-value*—Numeric value that specifies the physical member link through which all egressing traffic in this bundle will flow. The values are 1 through 8.
		• **auto**—The physical member link through which all egressing traffic on this bundle will flow is automatically chosen.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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)?`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring VLAN Bundles

This section describes how to configure a VLAN bundle. The creation of a VLAN bundle involves three main tasks:

SUMMARY STEPS

1. Create an Ethernet bundle
2. Create VLAN subinterfaces and assign them to the Ethernet bundle.
3. Assign Ethernet links to the Ethernet bundle.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Create an Ethernet bundle	
Step 2	Create VLAN subinterfaces and assign them to the Ethernet bundle.	
Step 3	Assign Ethernet links to the Ethernet bundle.	

These tasks are describe in detail in the procedure that follows.

Note In order for a VLAN bundle to be active, you must perform the same configuration on both ends of the bundle connection.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface Bundle-Ether** *bundle-id.vlan-id*

10. **dot1q vlan**
11. **ipv4 address** *ipv4-address mask*
12. **no shutdown**
13. **exit**
14. Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.
15. **end** or **commit**
16. **exit**
17. **exit**
18. **configure**
19. **interface** {**GigabitEthernet** | **TenGigE**}*interface-path-id*
20. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
21. **bundle port-priority** *priority*
22. **no shutdown**
23. Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.
24. **end** or **commit**
25. Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.
26. **show bundle Bundle-Ether** *bundle-id*
27. **show vlan interface**
28. **show vlan trunks** [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id]* [**brief** | **summary**] [**location** *node-id*]
29. **lacp fast-switchover**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3`	Creates and names a new Ethernet link bundle. This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:**	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization. **Note** The priority of the active and standby links is based on the value of the **bundle port-priority** command.
Step 7	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits the interface configuration submode.
Step 9	**interface Bundle-Ether** *bundle-id.vlan-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3.1`	Creates a new VLAN, and assigns the VLAN to the Ethernet bundle you created in Step 2. Replace the *bundle-id* argument with the *bundle-id* you created in Step 2. Replace the *vlan-id* with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved). **Note** When you include the *.vlan-id* argument with the **interface Bundle-Ether** *bundle-id* command, you enter subinterface configuration mode.
Step 10	**dot1q vlan** **Example:**	Assigns a VLAN to the subinterface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router#(config-subif)# dot1q vlan 10`	Replace the *vlan-id* argument with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved).
Step 11	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router#(config-subif)# ipv4 address 10.1.2.3/24`	Assigns an IP address and subnet mask to the subinterface.
Step 12	**no shutdown** **Example:** `RP/0/RP0/CPU0:router#(config-subif)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	Exits subinterface configuration mode for the VLAN subinterface.
Step 14	Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.	(Optional) Adds more subinterfaces to the bundle.

Step 15	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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.

Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# end`	Exits interface configuration mode.

	Command or Action	Purpose
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 18	**configure** **Example:** `RP/0/RP0/CPU0:router # configure`	Enters global configuration mode.
Step 19	**interface** {**GigabitEthernet** | **TenGigE**}*interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0`	Enters interface configuration mode for the Ethernet interface you want to add to the Bundle. Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the rack/slot/module format. **Note** A VLAN bundle is not active until you add an Ethernet interface on both ends of the link bundle.
Step 20	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds an Ethernet interface to the bundle you configured in Step 2 through Step 13. To enable active or passive LACP on the bundle, include the optional **mode active or mode passive** keywords in the command string. To add the interface to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** If you do not specify the **mode** keyword, the default mode is **on** (LACP is not run over the port).
Step 21	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.
Step 22	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.
Step 23	Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.	—

	Command or Action	Purpose

Step 24	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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.

Step 25	Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.	Brings up the other end of the link bundle.
Step 26	**show bundle Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3`	(Optional) Shows information about the specified Ethernet link bundle. The **show bundle Bundle-Ether** command displays information about the specified bundle. If your bundle has been configured properly and is carrying traffic, the State field in the **show bundle Bundle-Ether** command output shows the number "4," which means the specified VLAN bundle port is "distributing."
Step 27	**show vlan interface** **Example:** `RP/0/RP0/CPU0:router # show vlan interface`	Displays the current VLAN interface and status configuration.
Step 28	**show vlan trunks** [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id]* [**brief** | **summary**] [**location** *node-id*] **Example:** `RP/0/RP0/CPU0:router# show vlan trunk summary`	(Optional) Displays summary information about each of the VLAN trunk interfaces. • The keywords have the following meanings: **brief**—Displays a brief summary. **summary**—Displays a full summary. **location**—Displays information about the VLAN trunk interface on the given slot.

	Command or Action	Purpose
		interface—Displays information about the specified interface or subinterface. Use the **show vlan trunks** command to verify that all configured VLAN subinterfaces on an Ethernet bundle are "up."
Step 29	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.

Configuring POS Link Bundles

This section describes how to configure a POS link bundle.

Note In order for a POS bundle to be active, you must perform the same configuration on both connection endpoints of the POS bundle.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-POS** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface POS** *interface-path-id*
10. **bundle id** *bundle-id [mode {active | on | passive}]*
11. **bundle port-priority** *priority*
12. **no shutdown**
13. **exit**
14. Repeat Step 8 through Step 11 to add more links to a bundle
15. **end** or **commit**
16. **exit**
17. **exit**
18. Perform Step 1 through Step 15 on the remote end of the connection.
19. **show bundle Bundle-POS** *number*

20. **show lacp bundle Bundle-POS** *bundle-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-POS** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)#interface Bundle-POS 2`	Configures and names the new bundled POS interface. Enters the interface configuration submode, from where interface specific configuration commands are executed. Use the **exit** command to exit from the interface configuration submode, and get back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the ip address configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 620000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented according to a proprietary optimization. **Note** • The priority of the active and standby links is based on the value of the **bundle port-priority** command.
Step 7	**lacp fast-switchover** **Example:**	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.
Step 8	**exit**	Exits the interface configuration submode.
Step 9	**interface POS** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface POS 0/1/0/0`	Enters POS interface configuration mode and specifies the POS interface name and interface-path-id notation *rack/slot/module/port*.
Step 10	**bundle id** *bundle-id [mode {active | on | passive}]* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds the link to the specified bundle. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the link to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** • If you do not specify the **mode** keyword, the default mode is **on** (LACP is not run over the port).
Step 11	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.
Step 12	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration which forces the interface administratively down. The **no shutdown** command then returns the link to an up or down state, depending on the configuration and state of the link.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router# exit`	Exits the interface configuration submode for the POS interface.
Step 14	Repeat Step 8 through Step 11 to add more links to a bundle	(Optional) Adds more links to the bundle you created in Step 2.
Step 15	**end** or **commit** **Example:**	Saves configuration changes.

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration mode.
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 18	Perform Step 1 through Step 15 on the remote end of the connection.	Brings up the other end of the link bundle.
Step 19	**show bundle Bundle-POS** *number* **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-POS 1`	(Optional) Shows information about the specified POS link bundle.
Step 20	**show lacp bundle Bundle-POS** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show lacp bundle Bundle-POS 3`	(Optional) Shows detailed information about LACP ports and their peers.

Configuring the Default LACP Short Period Time Interval

This section describes how to configure the default short period time interval for sending and receiving LACP packets on a Gigabit Ethernet interface. This procedure also enables the LACP short period.

SUMMARY STEPS

1. **configure**
2. **interface GigabitEthernet** *interface-path*
3. **bundle id** *number* **mode active**
4. **lacp period short**
5. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode.
Step 3	**bundle id** *number* **mode active** **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active`	Specifies the bundle interface and puts the member interface in active mode.
Step 4	**lacp period short** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short`	Configures a short period time interval for the sending and receiving of LACP packets, using the default time period of 1000 milliseconds or 1 second.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

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

Configuring Custom LACP Short Period Time Intervals

This section describes how to configure custom short period time intervals (less than 1000 milliseconds) for sending and receiving LACP packets on a Gigabit Ethernet interface.

Note You must always configure the *transmit* interval at both ends of the connection before you configure the *receive* interval at either end of the connection. Failure to configure the *transmit* interval at both ends first results in route flapping (a route going up and down continuously). When you remove a custom LACP short period, you must do it in reverse order. You must remove the *receive* intervals first and then the *transmit* intervals.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links*
7. **exit**
8. **interface Bundle-Ether** *bundle-id.vlan-id*
9. **dot1q vlan** *vlan-id*
10. **ipv4 address** *ipv4-address mask*
11. **no shutdown**
12. **exit**
13. Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.
14. **end** or **commit**
15. **exit**
16. **exit**
17. **show ethernet trunk bundle-ether** *instance*
18. **configure**
19. **interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id*

20. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
21. **no shutdown**
22. Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.
23. **end** or **commit**
24. Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.
25. **show bundle Bundle-Ether** *bundle-id* [**reasons**]
26. **show ethernet trunk bundle-ether** *instance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3`	Creates and names a new Ethernet link bundle. This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1`	(Optional) Designates one active link and one link in standby mode that can take over immediately for a bundle if the active link fails (1:1 protection).

	Command or Action	Purpose
		Note • The default number of active links allowed in a single bundle is 8. • If the **bundle maximum-active** command is issued, then only the highest-priority link within the bundle is active. The priority is based on the value from the **bundle port-priority** command, where a lower value is a higher priority. Therefore, we recommend that you configure a higher priority on the link that you want to be the active link.
Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits the interface configuration submode.
Step 8	**interface Bundle-Ether** *bundle-id.vlan-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3.1`	Creates a new VLAN, and assigns the VLAN to the Ethernet bundle you created in Step 2. Replace the *bundle-id* argument with the *bundle-id* you created in Step 2. Replace the *vlan-id* with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved). **Note** • When you include the *.vlan-id* argument with the **interface Bundle-Ether** *bundle-id* command, you enter subinterface configuration mode.
Step 9	**dot1q vlan** *vlan-id* **Example:** `RP/0/RP0/CPU0:router#(config-subif)# dot1q vlan 10`	Assigns a VLAN to the subinterface. Replace the *vlan-id* argument with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved).
Step 10	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router#(config-subif)# ipv4 address 10.1.2.3/24`	Assigns an IP address and subnet mask to the subinterface.
Step 11	**no shutdown** **Example:** `RP/0/RP0/CPU0:router#(config-subif)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.

	Command or Action	Purpose
Step 12	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	Exits subinterface configuration mode for the VLAN subinterface.
Step 13	Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.	(Optional) Adds more subinterfaces to the bundle.

Step 14	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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)?`
		• 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.

Step 15	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	Exits interface configuration mode.
Step 16	**exit** **Example:** `RP/0/RSP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 17	**show ethernet trunk bundle-ether** *instance* **Example:** `RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5`	(Optional) Displays the interface configuration. The Ethernet bundle instance range is from 1 through 65535.
Step 18	**configure** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router # configure`	
Step 19	**interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0`	Enters the interface configuration mode for the Ethernet interface you want to add to the Bundle. Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the rack/slot/module format. **Note** • A VLAN bundle is not active until you add an Ethernet interface on both ends of the link bundle.
Step 20	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds an Ethernet interface to the bundle you configured in Step 2 through Step 13. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the interface to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** • If you do not specify the **mode** keyword, the default mode is **on (**LACP is not run over the port).
Step 21	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.
Step 22	Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.	—

Step 23	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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)?`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 24	Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.	Brings up the other end of the link bundle.
Step 25	**show bundle Bundle-Ether** *bundle-id* [**reasons**] **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3 reasons`	(Optional) Shows information about the specified Ethernet link bundle. The **show bundle Bundle-Ether** command displays information about the specified bundle. If your bundle has been configured properly and is carrying traffic, the State field in the **show bundle Bundle-Ether** command output will show the number "4," which means the specified VLAN bundle port is "distributing."
Step 26	**show ethernet trunk bundle-ether** *instance* **Example:** `RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5`	(Optional) Displays the interface configuration. The Ethernet bundle instance range is from 1 through 65535.

Configuration Examples for Link Bundling

This section contains the following examples:

Example: Configuring an Ethernet Link Bundle

The following example shows how to join two ports to form an EtherChannel bundle running LACP:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 3
RP/0/RP0/CPU0:Router(config-if)#ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RP0/CPU0:Router(config-if)# bundle maximum-active links 1 hot-standby
RP/0/RP0/CPU0:Router(config-if)# lacp fast-switchover
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface TenGigE 0/3/0/0
RP/0/RP0/CPU0:Router(config-if)# bundle-id 3 mode active
RP/0/RP0/CPU0:Router(config-if)# bundle port-priority 1
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router(config)# interface TenGigE 0/3/0/1
RP/0/RP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RP0/CPU0:Router(config-if)# bundle port-priority 2
```

```
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# exit
```

This example shows the configuration in the case of a mixed speed bundle:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 3
RP/0/RP0/CPU0:Router(config-if)#ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RP0/CPU0:Router(config-if)# interface bundle-ether 50
RP/0/RP0/CPU0:Router(config-if)# rootlacp fast-switchover
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface TenGigE 0/3/0/0
RP/0/RP0/CPU0:Router(config-if)#
RP/0/RP0/CPU0:Router(config-if)# bundle port-priority 1
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RP0/CPU0:Router(config-if)# bundleport-priority 2
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)#
RP/0/RP0/CPU0:Router(config-if)# bundleport-priority 2
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# root
RP/0/RP0/CPU0:Router(config)# commit
RP/0/RP0/CPU0:Router(config)# end
```

The following output is shown for the **show bundle bundle-ether** command:

show bundle bundle-ether50

```
Bundle-Ether50
 Status:                  Up
 Local links <active/standby/configured>:  4 / 0 / 4
 Local bandwidth <effective/available>:   130000000 (130000000) kbps
 MAC address (source):           0011.2233.4458 (Chassis pool)
 Inter-chassis link:            No
 Minimum active links / bandwidth:     1 / 1 kbps
 Maximum active links:           64
 Wait while timer:             2000 ms
 Load balancing:              Default
 LACP:                  Operational
  Flap suppression timer:         Off
  Cisco extensions:            Disabled
 mLACP:                  Not configured
 IPv4 BFD:                Not configured

 Port         Device      State    Port ID     B/W, kbps
 -------------------- --------------- ----------- -------------- ----------
 Te0/0/0/11      Local      Active    0x8000, 0x0002  10000000
   Link is Active
 Te0/0/0/16      Local      Active    0x8000, 0x0003  10000000
   Link is Active
 Te0/0/0/27      Local      Active    0x8000, 0x0004  10000000
   Link is Active
 Hu0/6/0/1       Local      Active    0x8000, 0x0001  100000000
   Link is Active
```

In order to view the weight of a mixed speed bundle, run the **show bundle load-balancing** command. The following is the truncated output of this command.

```
show bundle load-balancing bundle-ether50 location 0/0/cpu0

<snip>

Bundle-Ether50
 Type:         Ether (L3)
 Members <current/max>: 4/64
 Total Weighting:    13
 Load balance:     Default
 Locality threshold:  65
 Avoid rebalancing?   False
 Sub-interfaces:    1

 Member Information:
  Port:         LON ULID BW
  -------------------- --- ---- --
  Hu0/6/0/1        0   0 10
  Te0/0/0/11       1   1  1
  Te0/0/0/16       2   2  1
  Te0/0/0/27       3   3  1

 Platform Information:
 =====================

         * Bundle Summary Information *
          --------------------------

Interface       : Bundle-Ether50    Ifhandle      : 0x00000ce0
Lag ID         : 1           Virtual Port    : 255
Number of Members   : 4           Local to LC    : Yes
Hash Modulo Index   : 13
MGSCP Operational Mode : No

Member Information:
LON    Interface    ifhandle   SFP  port  slot remote/rack_id
-----  ---------------  ----------  ---  ----  ---- --------------
0    Hu0/6/0/1     0x100001c0  648  116   8    0/0
1    Te0/0/0/11    0x04000380   65   9   2    0/0
2    Te0/0/0/16    0x040004c0   67   8   2    0/0
3    Te0/0/0/27    0x04000780   72   4   2    0/0

</snip>
```

Example: Configuring a VLAN Link Bundle

The following example shows how to create and bring up two VLANS on an Ethernet bundle:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1.1


RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RP0/CPU0:Router(config-subif)# ip addr 10.2.3.4/24
RP/0/RP0/CPU0:Router(config-subif)# no shutdown
```

```
RP/0/RP0/CPU0:Router(config-subif)# exit
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1.2
RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 20
RP/0/RP0/CPU0:Router(config-subif)#
RP/0/RP0/CPU0:Router(config-subif)# no shutdown
RP/0/RP0/CPU0:Router(config-subif)# exit
RP/0/RP0/CPU0:Router(config)#interface gig 0/1/5/7
RP/0/RP0/CPU0:Router(config-if)# bundle-id 1 mode active
RP/0/RP0/CPU0:Router(config-if)# commit
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router # show vlan trunks
```

Example: Configuring a POS Link Bundle

The following example shows how to join two ports to form a Packet-over-SONET (POS) link bundle:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-POS 5
RP/0/RP0/CPU0:Router(config-if)#ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface POS 0/0/1/1
RP/0/RP0/CPU0:Router(config-if)# bundle id 5
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# exit
```

Example: Configuring EFP Load Balancing on an Ethernet Link Bundle

The following example shows how to configure all egressing traffic on the fixed members of a bundle to flow through the same physical member link automatically.

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)#bundle load-balancing hash auto
RP/0/RP0/CPU0:router(config-subif)#
```

The following example shows how to configure all egressing traffic on the fixed members of a bundle to flow through a specified physical member link.

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)#bundle load-balancing hash 1
RP/0/RP0/CPU0:router(config-subif)#
```

Examples: Configuring LACP Short Periods

The following example shows how to configure the LACP short period time interval to the default time of 1000 milliseconds (1 second):

```
config
interface gigabitethernet 0/0/0/1
   bundle id 1 mode active
```

```
   lacp period short
   commit
```

The following example shows how to configure custom LACP short period transmit and receive intervals to *less than* the default of 1000 milliseconds (1 second):

Router A

```
config
interface gigabitethernet 0/0/0/1
   bundle id 1 mode active
   lacp period short
   commit
```

Router B

```
config
interface gigabitethernet 0/0/0/1
   bundle id 1 mode active
   lacp period short
   commit
```

Router A

```
config
interface gigabitethernet 0/0/0/1
    lacp period short transmit 100
   commit
```

Router B

```
config
interface gigabitethernet 0/0/0/1
    lacp period short transmit 100
   commit
```

Router A

```
config
interface gigabitethernet 0/0/0/1
   lacp period short receive 100
   commit
```

Router B

```
config
interface gigabitethernet 0/0/0/1
   lacp period short receive 100
   commit
```

CHAPTER 8

Configuring Traffic Mirroring

This module describes the configuration of the traffic mirroring feature. Traffic mirroring is sometimes called port mirroring, or switched port analyzer (SPAN).

Feature History for Traffic Mirroring

Release 4.3.0	This feature was introduced on the Cisco CRS Router.

Introduction to Traffic Mirroring

Traffic mirroring, which is sometimes called port mirroring, or Switched Port Analyzer (SPAN) is a Cisco proprietary feature that enables you to monitor Layer 3 network traffic passing in, or out of, a set of Ethernet interfaces. You can then pass this traffic to a network analyzer for analysis.

Traffic mirroring copies traffic from one or more Layer 3 interfaces or sub-interfaces and sends the copied traffic to one or more destinations for analysis by a network analyzer or other monitoring device. Traffic mirroring does not affect the switching of traffic on the source interfaces or sub-interfaces, and allows the mirrored traffic to be sent to a destination next-hop address .

Traffic mirroring was introduced on switches because of a fundamental difference between switches and hubs. When a hub receives a packet on one port, the hub sends out a copy of that packet from all ports except from the one at which the hub received the packet. In the case of switches, after a switch boots, it starts to build up a Layer 2 forwarding table on the basis of the source MAC address of the different packets that the switch receives. After this forwarding table is built, the switch forwards traffic that is destined for a MAC address directly to the corresponding port.

Layer 2 SPAN is not supported on the Cisco CRS Router.

The difference from Layer 2 SPAN is that the destination for mirrored packets is specified as a next-hop IP address rather than an explicit interface, and only Layer 3 packets are mirrored. In the Cisco IOS XR Software Release 4.3.0, it is assumed that the next-hop IP address should be looked up in the default VRF routing table.

Implementing Traffic Mirroring on the Cisco ASR 9000 Series RouterCisco CRS Router

Traffic Mirroring Terminology

- Ingress Traffic — Traffic that comes into the router.
- Egress Traffic — Traffic that goes out of the router.
- Source (SPAN) interface — An ingress interface that is monitored using the SPAN feature.
- Destination (SPAN) Nexthop — An egress Nexthop address where a network analyzer is connected.
- Monitor Session A designation for a collection of SPAN configurations consisting of many source interfaces and a set of destinations. In the Cisco IOS XR Software Release 4.3.0, only one destination is supported per monitor session.

Characteristics of the Source Port

A source port, also called a monitored port, is a routed port that you monitor for network traffic analysis. In a single traffic mirroring session, you can monitor source port traffic. Your router can support any number of source ports (up to a maximum number of 800).

A source port has these characteristics:

- It can be any port type, such as Bundle Interface, Gigabit Ethernet, 10-Gigabit Ethernet, or EFPs.

  Note Bridge group virtual interfaces (BVIs) are not supported.

- Each source port can be monitored in only one traffic mirroring session.
- Interfaces over which mirrored traffic may be routed must not be configured as a source port.
- ACL-based traffic mirroring. Traffic is mirrored based on the configuration of the global interface ACL. This is optional on the Cisco CRS Router.

In the figure above, the network analyzer is attached to a port that is configured to receive a copy of every packet that host A sends. This port is called a traffic mirroring port.

Characteristics of the Monitor Session

A monitor session is a collection of traffic mirroring configurations consisting of a single destination and, potentially, many source interfaces. For any given monitor session, the traffic from the source interfaces (called *source ports*) is sent to the destination. Some optional operations such as ACL filtering can be performed on the mirrored traffic streams. If there is more than one source port in a monitoring session, the traffic from the several mirrored traffic streams is combined at the destination. The result is that the traffic that comes out of the destination is a combination of the traffic from one or more source ports, and the traffic from each source port may or may not have ACLs applied to it.

Monitor sessions have these characteristics:

- A single Cisco CRS Router can have a maximum of eight monitor sessions.

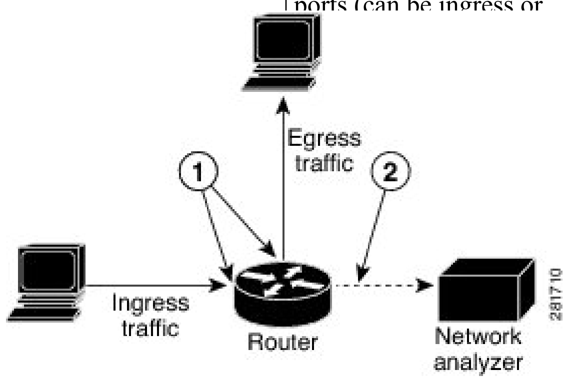
- A single monitor session can have only one destination .
- A single destination can belong to only one monitor session.
- A single Cisco CRS Router can have a maximum of 800 source ports.
- A monitor session can have a maximum of 800 source ports, as long as the maximum number of source ports from all monitoring sessions does not exceed 800.

Characteristics of the Destination

Each session must have a destination that receives a copy of the traffic from the source ports.

A destination has these characteristics:

1	Source traffic mirroring ports (can be ingress or	2	Destination traffic mirroring port



Restrictions for Traffic Mirroring

A maximum of eight monitoring sessions are supported. You can configure 800 source ports on a single monitoring session or an aggregate of 800 source ports over eight monitoring sessions.

These forms of traffic mirroring are not supported:

- Mirroring traffic to a GRE tunnel (also known as Encapsulated Remote Switched Port Analyzer [ER-SPAN] in Cisco IOS Software).
- If the destination of traffic mirroring is an nV satellite port and ICL is configured with a bundle interface, then replicated packets are not forwarded to the destination.
- MPLS traffic or tunnel traffic.
- Layer 2 traffic mirroring.
- VRF at destination ports.
- Mirroring for POS interfaces.
- Mirroring of egress traffic.

Configuring Traffic Mirroring

These tasks describe how to configure traffic mirroring:

How to Configure Layer-3 Traffic Mirroring

SUMMARY STEPS

1. **configure**
2. **monitor-session** *session-name* **[ipv4|ipv6]**
3. **destination** `next-hop` *ip address*
4. **exit**
5. **interface** *source-interface*
6. **monitor-session** *session-name* **{ipv4|ipv6}** [**direction** {**rx-only| tx-only**}]
7. **end** or **commit**
8. **show monitor-session [session-name**] **status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**monitor-session** *session-name* **[ipv4|ipv6]**
	Example:
	`RP/0/RP0/CPU0:router(config)# monitor-session mon1`
	`RP/0/RP0/CPU0:router(config-mon)#`
		Defines a monitor session and enters monitor session configuration mode. The monitor-session name is a printable string that can be at most 79 characters in length.
		Note • This command triggers entry in to the monitor-session sub-mode and creates the session. The session is non-operable until a destination is configured for the session. The destination can be either an IPv4 or IPv6 address.

Step 3	**destination** `next-hop` *ip address* **Example:** `RP/0/RP0/CPU0:router(config-mon)# destination next-hop ipv4 254.23.24.5`	Configures the destination for the current monitor-session to be a next-hop IP address (whose type matches that of the monitor-session). **Note** • This may only be specified for ipv4 and ipv6 monitor-sessions. A monitor session can be either for IPv4 or for IPv6. It cannot support both together.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`RP/0/RP0/CPU0:router(config-mon)# exit`
	`RP/0/RP0/CPU0:router(config)#`
		Exits monitor session configuration mode and returns to global configuration mode.

Step 5	**interface** *source-interface* **Example:** `RP/0/RP0/CPU0:router(config)# interface gigabitethernet0/0/0/11.10`	Enters interface configuration mode for the specified interface. The interface number is entered in *rack*/*slot*/*module*/*port* notation. For more information about the syntax for the router, use the question mark (?) online help function.
Step 6	**monitor-session** *session-name* {**ipv4**|**ipv6**} [**direction** {**rx-only**| **tx-only**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# monitor-session mon1`	Specifies the monitor session to be used on this interface. Use the **direction** keyword to specify that only ingress or egress traffic is mirrored. To support both IPv4 and IPv6 mirroring, separate monitor sessions defined for IPv4 and IPv6 must be attached to the interface. The interface name can be the name of any Ethernet interface. The monitor-session name is a printable string at most 79 characters in length. **Note** • If no type is given, ethernet is assumed. Only Rx traffic is mirrored.

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

	Command or Action	Purpose
Step 8	**show monitor-session [session-name] status** **Example:** `RP/0/RP0/CPU0:router# show monitor-session`	Displays information about the traffic mirroring session.

How to Configure ACL-Based Traffic Mirroring

Before you begin

The global interface ACL should be configured using one of these commands with the **capture** keyword:

- **ipv4 access-list**
- **ipv6 access-list**
- **ethernet-services access-list**

  For more information, refer to the *Cisco IOS XR IP Addresses and Services Command Reference for the Cisco CRS Router* or the *Cisco IOS XR Virtual Private Network Command Reference for the Cisco CRS Router*.

SUMMARY STEPS

1. **configure**
2. **monitor-session** *session-name* **[ipv4|ipv6]**
3. **destination** `next-hop` *ip address*
4. **exit**
5. **interface** *source-interface*
6. **ethernet-services access-group** *access-list-name* [**ingress** | **egress**]
7. **monitor-session** *session-name* [ipv4|ipv6] [direction {rx-only|tx-only}]
8. **acl**
9. **end** or **commit**
10. **show monitor-session [session-name] status** [**detail**] [**error**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**monitor-session** *session-name* **[ipv4|ipv6]**
	Example:
	`RP/0/RP0/CPU0:router(config)# monitor-session mon1`
	`RP/0/RP0/CPU0:router(config-mon)#`
		Defines a monitor session and enters monitor session configuration mode. The monitor-session name is a printable string that can be at most 79 characters in length.

	Command or Action	Purpose
		Note • This command triggers entry in to the monitor-session sub-mode and creates the session. The session is non-operable until a destination is configured for the session. The destination can be either an IPv4 or IPv6 address.
Step 3	**destination** `next-hop` *ip address* **Example:** `RP/0/RP0/CPU0:router(config-mon)# destination next-hop ipv4 254.23.24.5`	Configures the destination for the current monitor-session to be a next-hop IP address (whose type matches that of the monitor-session). **Note** • This may only be specified for ipv4 and ipv6 monitor-sessions. A monitor session can be either for IPv4 or for IPv6. It cannot support both together.

Step 4	**exit**
	Example:
	`RP/0/RP0/CPU0:router(config-mon)# exit`
	`RP/0/RP0/CPU0:router(config)#`
		Exits monitor session configuration mode and returns to global configuration mode.

Step 5	**interface** *source-interface* **Example:** `RP/0/RP0/CPU0:router(config)# interface gigabitethernet0/0/0/11`	Enters interface configuration mode for the specified interface. The interface number is entered in *rack*/*slot*/*module*/*port* notation. For more information about the syntax for the router, use the question mark (?) online help function.
Step 6	**ethernet-services access-group** *access-list-name* [**ingress** | **egress**] **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet-services access-group acl1 ingress`	Associates the access list definition with the interface being mirrored.
Step 7	**monitor-session** *session-name* [ipv4|ipv6] [direction {rx-only|tx-only}] **Example:** `RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 direction rx-only`	Specifies the monitor session to be used on this interface.
Step 8	**acl** **Example:** `RP/0/RP0/CPU0:router(config-if-mon)# acl`	Specifies that the traffic mirrored is according to the defined global interface ACL.
Step 9	**end** or **commit**	Saves configuration changes.

	Command or Action	Purpose

	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 10	**show monitor-session [session-name]** **status** [**detail**] [**error**] **Example:** `RP/0/RP0/CPU0:router# show monitor-session`	Displays information about the monitor session.

Troubleshooting ACL-Based Traffic Mirroring

Take note of these configuration issues:

- Even when the **acl** command is configured on the source mirroring port, if the ACL configuration command does not use the **capture** keyword, no traffic gets mirrored.
- If the ACL configuration uses the **capture** keyword, but the **acl** command is not configured on the source port, traffic is mirrored, but no access list configuration is applied.

This example shows both the **capture** keyword in the ACL definition and the **acl** command configured on the interface:

```
monitor-session tm_example
!
ethernet-services access-list tm_filter
 10 deny 0000.1234.5678 0000.abcd.abcd any capture
!
interface GigabitEthernet0/2/0/0
 monitor-session tm_example direction rx-only
 acl
!
 l2transport
 !
```

```
 ethernet-services access-group tm_filter ingress
end
```

Traffic Mirroring Configuration Examples

This section contains examples of how to configure traffic mirroring:

Viewing Monitor Session Status: Example

This example shows sample output of the **show monitor-session** command with the **status** keyword:

```
RP/0/RP0/CPU0:router# show monitor-session test status

Monitor-session test (ipv4)

Destination Nexthop 255.254.254.4
=========================================================================================
Source Interface   Dir         Status
-----------------------------------------------------------------------------------------
Gi0/0/0/2.2     Rx    Not operational (source same as destination)
Gi0/0/0/2.3      Rx    Not operational (Destination not active)
Gi0/0/0/2.4      Rx    Operational
Gi0/0/0/4       Rx    Error: see detailed output for explanation
RP/0/RP0/CPU0:router# show monitor-session test status error

Monitor-session test
Destination Nexthop ipv4 address 255.254.254.4
===============================================================
Source Interface     Status
---------------------------------------------------------------
Gi0/0/0/4    < Error: FULL Error Details >
```

Monitor Session Statistics: Example

Use the **show monitor-session** command with the **counters** keyword to show the statistics/counters (received/transmitted/dropped) of different source ports. For each monitor session, this command displays a list of all source interfaces and the replicated packet statistics for that interface.

The full set of statistics displayed for each interface is:

- RX replicated packets and octets
- TX replicated packets and octets
- Non-replicated packet and octets

```
RP/0/RP00/CPU0:router# show monitor-session counters

Monitor-session ms1
 GigabitEthernet0/2/0/19.10
  Rx replicated: 1000 packets, 68000 octets
  Tx replicated: 1000 packets, 68000 octets
  Non-replicated: 0 packets, 0 octets
```

Use the **clear monitor-session counters** command to clear any collected statistics. By default this command clears all stored statistics; however, an optional interface filter can be supplied.

```
RP/0/RP00/CPU0:router# clear monitor-session counters
```

Layer 3 ACL-Based Traffic Mirroring: Example

This example shows how to configure Layer 3 ACL-based traffic mirroring:

```
RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# monitor-session ms1
RP/0/RP00/CPU0:router(config-mon)# destinationnext-hop 10.1.1.0
RP/0/RP00/CPU0:router(config-mon)# commit

RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface gig0/2/0/11
RP/0/RP00/CPU0:router(config-if)# ipv4 access-group span ingress
RP/0/RP00/CPU0:router(config-if)# monitor-session ms1
RP/0/RP00/CPU0:router(config-if-mon)# commit

RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# ipv4 access-list span
RP/0/RP00/CPU0:router(config-ipv4-acl)# 5 permit ipv4 any any dscp 5 capture
RP/0/RP00/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any
RP/0/RP00/CPU0:router(config-ipv4-acl)# commit
```

Troubleshooting Traffic Mirroring

When you encounter any issue with traffic mirroring, begin troubleshooting by checking the output of the **show monitor-session status** command. This command displays the recorded state of all sessions and source interfaces:

```
Monitor-session sess1
<Session status>
================================================================================
Source Interface   Dir  Status
-------------------- ---- ----------------------------------------------------
Gi0/0/0/0       Both <Source interface status>
Gi0/0/0/2       Both <Source interface status>
```

In the preceding example, the line marked as `<Session status>` can indicate one of these configuration errors:

Session Status	Explanation
Session is not configured globally	The session does not exist in global configuration. Check **show run** command output to ensure that a session with a correct name has been configured.
Destination next-hop IPv4/IPv6 address <addr> is not configured	The IPv4 or IPv6 address that has been configured as the destination does not exist.
Destination next-hop IPv4 address <addr> not reachable	The IPv4 or IPv6 address that has been configured as the destination is not reachable or is not in the Up state. You can verify the status of the destination using the **show monitor-session status detail** command.

The <Source interface status> can report these messages:

Source Interface Status	Explanation
Operational	Everything appears to be working correctly in traffic mirroring PI. Please follow up with the platform teams in the first instance, if mirroring is not operating as expected.
Not operational (Session is not configured globally)	The session does not exist in global configuration. Check the **show run** command output to ensure that a session with the right name has been configured.
Not operational (destination not known)	The session exists, but it either does not have a destination interface specified, or the destination interface named for the session does not exist (for example, if the destination is a sub-interface that has not been created).
Not operational (destination not active)	The destination interface or pseudowire is not in the Up state. See the corresponding *Session status* error messages for suggested resolution.
Not operational (source state <down-state>)	The source interface is not in the Up state. You can verify the state using the **show interfaces** command. Check the configuration to see what might be keeping the interface from coming up (for example, a sub-interface needs to have an appropriate encapsulation configured).
Error: see detailed output for explanation	Traffic mirroring has encountered an error. Run the **show monitor-session status detail** command to display more information.

The **show monitor-session status detail** command displays full details of the configuration parameters, and of any errors encountered. For example:

RP/0/RP0**`show monitor-session status detail`**

Here are additional trace and debug commands:

```
RP/0/RP00/CPU0:router# show monitor-session platform trace ?

 all   Turn on all the trace
 errors Display errors
 events Display interesting events

RP/0/RP00/CPU0:router# show monitor-session trace ?

 process Filter debug by process

RP/0/RP00/CPU0:router# debug monitor-session platform ?

 all   Turn on all the debugs
 errors VKG SPAN EA errors
 event  VKG SPAN EA event
 info  VKG SPAN EA info
```

```
RP/0/RP00/CPU0:router# debug monitor-session platform all

RP/0/RP00/CPU0:router# debug monitor-session platform event

RP/0/RP00/CPU0:router# debug monitor-session platform info

RP/0/RP00/CPU0:router# show monitor-session status ?

 detail   Display detailed output
 errors   Display only attachments which have errors
 internal Display internal monitor-session information
 |      Output Modifiers

RP/0/RP00/CPU0:router# show monitor-session status

RP/0/RP00/CPU0:router# show monitor-session status errors

RP/0/RP00/CPU0:router# show monitor-session status internal
```

CHAPTER 9

Configuring Virtual Loopback and Null Interfaces

This module describes the configuration of loopback and null interfaces. Loopback and null interfaces are considered virtual interfaces.

A virtual interface represents a logical packet switching entity within the router. Virtual Interfaces have a global scope and do not have an associated location. Virtual interfaces have instead a globally unique numerical ID after their names. Examples are Loopback 0, Loopback1, and Loopback 99999. The ID is unique per virtual interface type to make the entire name string unique such that you can have both Loopback 0 and Null 0.

Loopback and null interfaces have their control plane presence on the active route processor (RP). The configuration and control plane are mirrored onto the standby RP and, in the event of a switchover, the virtual interfaces move to the ex-standby, which then becomes the newly active RP.

Prerequisites for Configuring Virtual Interfaces

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.

Information About Configuring Virtual Interfaces

To configure virtual interfaces, you must understand the following concepts:

Virtual Loopback Interface Overview

A virtual loopback interface is a virtual interface with a single endpoint that is always up. Any packet transmitted over a virtual loopback interface is immediately received by the selfsame interface. Loopback interfaces emulate a physical interface.

In Cisco IOS XR software virtual loopback interfaces perform the following functions:

- Loopback interfaces can act as a termination address for routing protocol sessions. This allows routing protocol sessions to stay up even if the outbound interface is down.
- You can ping the loopback interface to verify that the router IP stack is working properly.

In applications where other routers or access servers attempt to reach a virtual loopback interface, you must configure a routing protocol to distribute the subnet assigned to the loopback address.

Packets routed to the loopback interface are rerouted back to the router or access server and processed locally. IP packets routed out the loopback interface but not destined to the loopback interface are dropped. Under these two conditions, the loopback interface can behave like a null interface.

Null Interface Overview

A null interface functions similarly to the null devices available on most operating systems. This interface is always up and can never forward or receive traffic; encapsulation always fails. The null interface provides an alternative method of filtering traffic. You can avoid the overhead involved with using access lists by directing undesired network traffic to the null interface.

The only interface configuration command that you can specify for the null interface is the **ipv4 unreachables** command. With the **ipv4 unreachables** command, if the software receives a nonbroadcast packet destined for itself that uses a protocol it does not recognize, it sends an Internet Control Message Protocol (ICMP) protocol unreachable message to the source. If the software receives a datagram that it cannot deliver to its ultimate destination because it knows of no route to the destination address, it replies to the originator of that datagram with an ICMP host unreachable message.

The Null0 interface is created by default on the RP during boot and cannot be removed. The **ipv4 unreachables** command can be configured for this interface, but most configuration is unnecessary because this interface just discards all the packets sent to it.

The Null0 interface can be displayed with the **show interfaces null0** command.

Virtual Management Interface Overview

Configuring an IPv4 virtual address enables you to access the router from a single virtual address with a management network without prior knowledge of which RP is active. An IPv4 virtual address persists across route processor (RP) switchover situations. For this to happen, the virtual IPv4 address must share a common IPv4 subnet with a management Ethernet interface on both RPs.

On a Cisco CRS-1 Router where each RP has multiple management Ethernet interfaces, the virtual IPv4 address maps to the management Ethernet interface on the active RP that shares the same IP subnet.

Active and Standby RPs and Virtual Interface Configuration

The standby RP is available and in a state in which it can take over the work from the active RP should that prove necessary. Conditions that necessitate the standby RP to become the active RP and assume the active RP's duties include:

- Failure detection by a watchdog
- Administrative command to take over
- Removal of the active RP from the chassis

If a second RP is not present in the chassis while the first is in operation, a second RP may be inserted and automatically becomes the standby RP. The standby RP may also be removed from the chassis with no effect on the system other than loss of RP redundancy.

After switchover, the virtual interfaces all are present on the standby (now active) RP. Their state and configuration are unchanged and there has been no loss of forwarding (in the case of tunnels) over the interfaces during the switchover. The routers use nonstop forwarding (NSF) over bundles and tunnels through the switchover of the host RP.

Note The user need not configure anything to guarantee that the standby interface configurations are maintained.

Protocol configuration such as tacacs source-interface, snmp-server trap-source, ntp source, logging source-interface do not use the virtual management IP address as their source by default. Use the **ipv4 virtual address use-as-src-addr** command to ensure that the protocol uses the virtual IPv4 address as its source address. Alternatively, you can also configure a loopback address with the designated or desired IPv4 address and set that as the source for protocols such as TACACS+ using the **tacacs source-interface** command.

How to Configure Virtual Interfaces

This section contains the following procedures:

Configuring Virtual Loopback Interfaces

This task explains how to configure a basic loopback interface.

Restrictions

The IP address of a loopback interface must be unique across all routers on the network. It must not be used by another interface on the router, and it must not be used by an interface on any other router on the network.

SUMMARY STEPS

1. **configure**
2. **interface loopback** *interface-path-id*
3. **ipv4 address** *ip-address*
4. **end** or **commit**
5. **show interfaces** *type interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface loopback** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Loopback 3`	Enters interface configuration mode and names the new loopback interface.
Step 3	**ipv4 address** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38/32`	Assigns an IP address and subnet mask to the virtual loopback interface using the **ipv4 address** configuration command.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**show interfaces** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces Loopback 3`	(Optional) Displays the configuration of the loopback interface.

Configuring Null Interfaces

This task explains how to configure a basic Null interface.

SUMMARY STEPS

1. **configure**
2. **interface null 0**

3. **end** or **commit**
4. **show interfaces null 0**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface null 0** **Example:** `RP/0/RP0/CPU0:router#(config)# interface null 0`	Enters null0 interface configuration mode.

Step 3	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-null0)# end`
	or
	`RP/0/RP0/CPU0:router(config-null0)# 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.

Step 4	**show interfaces null 0** **Example:** `RP/0/RP0/CPU0:router# show interfaces null0`	Verifies the configuration of the null interface.

Configuring Virtual IPV4 Interfaces

This task explains how to configure an IPv4 virtual interface.

SUMMARY STEPS

1. **configure**
2. **ipv4 address virtual address** *ipv4-*
3. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ipv4 address virtual address** *ipv4-* **Example:** `RP/0/RP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8`	Defines an IPv4 virtual address for the management Ethernet interface.

Step 3	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-null0)# end`
	or
	`RP/0/RP0/CPU0:router(config-null0)# 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.

Configuration Examples for Virtual Interfaces

This section provides the following configuration examples:

Configuring a Loopback Interface: Example

The following example indicates how to configure a loopback interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Loopback 3
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38/32
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Loopback 3

Loopback3 is up, line protocol is up
 Hardware is Loopback interface(s)
 Internet address is 172.18.189.38/32
 MTU 1514 bytes, BW Unknown
   reliability 0/255, txload Unknown, rxload Unknown
 Encapsulation Loopback, loopback not set
 Last clearing of "show interface" counters never
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 total input drops
   0 drops for unrecognized upper-level protocol
   Received 0 broadcast packets, 0 multicast packets
   0 packets output, 0 bytes, 0 total output drops
   Output 0 broadcast packets, 0 multicast packets
```

Configuring a Null Interface: Example

The following example indicates how to configure a null interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Null 0
RP/0/RP0/CPU0:router(config-null0)# ipv4 unreachables
RP/0/RP0/CPU0:router(config-null0)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Null 0

Null0 is up, line protocol is up
 Hardware is Null interface
 Internet address is Unknown
 MTU 1500 bytes, BW Unknown
   reliability 0/255, txload Unknown, rxload Unknown
 Encapsulation Null, loopback not set
 Last clearing of "show interface" counters never
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 total input drops
   0 drops for unrecognized upper-level protocol
   Received 0 broadcast packets, 0 multicast packets
   0 packets output, 0 bytes, 0 total output drops
   Output 0 broadcast packets, 0 multicast packets
```

Configuring a Virtual IPv4 Interface: Example

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8
RP/0/RP0/CPU0:router(config-null0)# commit
```

CHAPTER 10

Configuring Clear Channel SONET Controllers

This module describes the configuration of clear channel SONET controllers.

SONET controller configuration is a preliminary step toward Packet-over-SONET/SDH (POS) configuration on routers using Cisco IOS XR software.

SONET allows you to define optical signals and a synchronous frame structure for multiplexed digital traffic. It is a set of standards defining the rates and formats for optical networks specified in American National Standards Institute (ANSI) T1.105, ANSI T1.106, and ANSI T1.117.

The commands for configuring the Layer 1 SONET controllers are provided in the *Cisco IOS XR Interface and Hardware Component Command Reference*.

Feature History for Configuring SONET Controllers on Cisco IOS XR Software

Release	Modification
Release 2.0	This feature was introduced on the Cisco CRS-1 Router.
Release 3.3.0	Support was added on the Cisco CRS-1 Router for the Cisco 1-Port OC-768c/STM-256c POS PLIM.
Release 3.4.0	Support was added on the Cisco CRS-1 Router for the following hardware: • Cisco 2 port OC-48/STM-16 POS SPA • Cisco 4 port OC-48/STM-16 POS SPA
Release 3.8.0	The **delay trigger line** command was updated to **line delay trigger in the following sections:** • How to Configure Clear Channel SONET Controllers • Configuring a Hold-off Timer to Prevent Fast Reroute from being Triggered

Release 4.0.1	Support for APS was added on the Cisco CRS-1 Router for the following hardware: • 1-Port OC-192c/STM-64 POS/RPR XFP SPA • 2-Port and 4-Port OC-48c/STM-16 POS SPA • 4-Port OC-3c/STM-1 POS SPA • 4-Port and 8-Port OC-3c/STM-1 POS SPA • 8-Port OC-3c/STM-1 POS SPA
Release 4.3.0	Support for APS was added on the Cisco CRS-3 Router for the following hardware: • 1-Port OC-192c/STM-64 POS/RPR XFP SPA • 2-Port and 4-Port OC-48c/STM-16 POS SPA • 4-Port OC-3c/STM-1 POS SPA • 4-Port and 8-Port OC-3c/STM-1 POS SPA • 8-Port OC-3c/STM-1 POS SPA

Prerequisites for Configuring Clear Channel SONET Controllers

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 SONET controllers, be sure that the following tasks and conditions are met:

- You have at least one of the following physical layer interface module (PLIM) cards installed in your chassis:
  - Cisco 4-Port OC-3c/STM-1 POS SPA
  - Cisco 8-Port OC-12c/STM-4 POS SPA
  - Cisco 16-Port OC-48c/STM-16c POS
  - Cisco 4-Port OC-192c/STM-64c POS
  - Cisco 1-Port OC-192c/STM-64 POS/RPR XFP SPA
  - Cisco 1-Port OC-768c/STM-256c POS PLIM

- You know how to apply the specify the SONET controller name and instance identifier with the generalized notation *rack/slot/module/port*. The SONET controller name and instance identifier are required with the **controller sonet** command.

Information About Configuring SONET Controllers

To configure SONET controllers, you must understand the following concepts:

SONET Controller Overview

In routers supporting Cisco IOS XR software, the physical ports on certain line cards are called controllers. Before you can configure a POS, SRP, or serial interface, you need to configure the SONET controller.

The commands used to configure the physical SONET port are grouped under the SONET controller configuration mode. To get to the SONET controller configuration mode, enter the **controller sonet** command in global configuration mode. You can also preconfigure a SONET controller using the **controller preconfigure sonet** global configuration command.

The router uses SONET controllers for Layer 1 and Layer 2 processing.

Note Path UNEQ is not supported on the OC-768 card. Therefore, UNEQ-P and PPLM alarms are not reported for any unequipped C2 byte that is received on an OC-768 interface. Cisco supports all error codes from the ERDI-P standard except for the UNEQ-P code.

Default Configuration Values for SONET Controllers

The table below describes some default configuration parameters that are present on SONET controllers.

Table 7: SONET Controller Default Configuration Values

Parameter	Default Value	Configuration File Entry
Reporting of the following alarms for a SONET controller: • Bit 1 (B1) bit error rate (BER) threshold crossing alert (TCA) errors • Bit 2 (B2) BER TCA errors • Signal failure BER errors • Section loss of frame (SLOF) errors • Section loss of signal (SLOS) errors	**enabled**	To disable reporting of any alarms enabled by default, use the **no report** [**b1-tca** | **b2-tca** | **sf-ber** | **slof** | **slos**] command in SONET/SDH configuration mode. To enable reporting of line alarm indication signal (LAIS), line remote defect indication (LRDI), or signal degradation BER errors, use the **report** [**lais** | **lrdi** | **sd-ber**] command in SONET/SDH configuration mode.

Parameter	Default Value	Configuration File Entry
Reporting of the following alarms for a SONET path controller: • Bit 3 (B3) BER TCA errors • Path loss of pointer (PLOP) errors	**enabled**	To disable B3 BER TCA or PLOP reporting on the SONET path controller, enter the **no report b3-tca** or **no report plop** command in SONET/SDH path configuration submode. To enable reporting of path alarm indication signal (PAIS), path payload mismatch (PPLM), path remote defect indication (PRDI), or path trace identity mismatch (PTIM) errors, use the **report** [**pais** | **pplm** | **prdi** | **ptim** command in SONET/SDH path configuration submode.
Synchronous payload envelope (SPE) scrambling	**enabled**	To disable SPE scrambling on a SONET controller, enter the **path scrambling disable** command in SONET controller configuration submode.
Keepalive timer	**enabled**	To turn off the keepalive timer, enter the **keepalive disable** command in interface configuration mode.

SONET APS

The automatic protection switching (APS) feature allows switchover of interfaces in the event of failure, and is often required when connecting SONET equipment to telco equipment. APS refers to the mechanism of using a *protect* interface in the SONET network as the backup for *working* interface. When the working interface fails, the protect interface quickly assumes its traffic load. The working interfaces and their protect interfaces make up an *APS group*.

In Cisco IOS XR software, SONET APS configuration defines a working line and a protection line for each redundant line pair. The working line is the primary or preferred line, and communications take place over that line as long as the line remains operative. If a failure occurs on the working line, APS initiates a switchover to the protection line. For proper APS operation between two routers, a working line on one router must also be the working line on the other router, and the same applies to the protection line.

In a SONET APS group, each connection may be bidirectional or unidirectional, and revertive or non-revertive. The same signal payload is sent to the working and protect interfaces. The working and protect interfaces can terminate in two ports of the same card, or in different cards in the same router, or in two different routers.

The protect interface directs the working interface to activate or deactivate in the case of degradation, loss of channel signal, or manual intervention. If communication between the working and protect interfaces is lost, the working router assumes full control of the working interface as if no protect circuit existed.

In an APS group, each line is called a *channel*. In bidirectional mode, the receive and transmit channels are switched as a pair. In unidirectional mode, the transmit and receive channels are switched independently. For example, in bidirectional mode, if the receive channel on the working interface has a loss of channel signal, both the receive and transmit channels are switched.

How to Configure Clear Channel SONET Controllers

This section contains the following procedures:

Configuring a Clear Channel SONET Controller

This task explains how to configure SONET controllers as a prerequisite to configuring POS and SRP or serial interfaces.

Before you begin

- You need to have a supported POS line card or channelized SPA installed in a router that is running the corresponding supported Cisco IOS XR software release.
- If you want to ensure recovery from fiber or equipment failures, then configure SONET APS on the router as describe in the Configuring SONET APS.

SUMMARY STEPS

1. **configure**
2. **controller sonet** *interface-path-id*
3. **clock source** {**internal** | **line**}
4. **line delay trigger** *value*
5. **line delay clear** *value*
6. **framing** {**sdh** | **sonet**}
7. **loopback** {**internal** | **line**}
8. **overhead** {**j0** | **s1s0**} *byte-value*
9. **path** *keyword [values]*
10. **end** or **commit**
11. **show controllers sonet** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**controller sonet** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller sonet 0/1/0/0`	Enters SONET controller configuration submode and specifies the SONET controller name and instance identifier with the *rack/slot/module/port* notation.
Step 3	**clock source** {**internal** | **line**} **Example:** `RP/0/RP0/CPU0:router(config-sonet)# clock source internal`	Configures the SONET port transmit clock source, where the **internal** keyword sets the internal clock and **line** keyword sets the clock recovered from the line. • Use the **line** keyword whenever clocking is derived from the network. Use the **internal** keyword when two routers are connected back-to-back or over fiber for which no clocking is available. • The **line** clock is the default. **Note** • Internal clocking is required for SRP interfaces.
Step 4	**line delay trigger** *value* **Example:** `RP/0/RP0/CPU0:router(config-sonet)# line delay trigger 3000`	(Optional) Configures the SONET line delay trigger values, where the trigger values are in the range from 0 through 60000 milliseconds, and the default delay trigger value is 0 milliseconds.
Step 5	**line delay clear** *value* **Example:** `RP/0/RP0/CPU0:router(config-sonet)# line delay clear 4000`	(Optional) Configures the amount of time before a SONET line delay trigger alarm is cleared. The range is from 1000 through 180000 milliseconds, and the default is 10 seconds.
Step 6	**framing** {**sdh** | **sonet**} **Example:** `RP/0/RP0/CPU0:router(config-sonet)# framing sonet`	(Optional) Configures the controller framing with either the **sdh** keyword for Synchronous Digital Hierarchy (SDH) framing or the **sonet** keyword for SONET framing. SONET framing (**sonet**) is the default.
Step 7	**loopback** {**internal** | **line**} **Example:** `RP/0/RP0/CPU0:router(config-sonet)# loopback internal`	(Optional) Configures the SONET controller for loopback, where the **internal** keyword selects internal (terminal) loopback, or the **line** keyword selects line (facility) loopback.
Step 8	**overhead** {**j0** | **s1s0**} *byte-value* **Example:** `RP/0/RP0/CPU0:router(config-sonet)# overhead s1s0`	(Optional) Configures the controller's overhead, where the **j0** keyword specifies the STS identifier (J0/C1) byte, and the **s1s0** keyword specifies bits s1 and s0 of H1 byte. • The default byte value for the **j0** keyword is 0xcc, and the default byte value for the **s1s0** keyword is 0.

	Command or Action	Purpose
		• The range of valid values for **j0** and **s1s0** is 0 through 255.
Step 9	**path** *keyword [values]* **Example:** `RP/0/RP0/CPU0:router(config-sonet)# path delay trigger 25`	(Optional) Configures SONET controller path values. Keyword definitions are as follows: • **ais-shut**—Set sending path alarm indication signal (PAIS) when shut down. • **b3-ber-prdi**—Enable sending of a path-level remote defect indication (PRDI) when the bit error rate (BER) bit interleaved parity (BIP) threshold is exceeded. • **delay clear** *value*—Set the amount of time before a Synchronous Transport Signal (STS) path delay trigger alarm is cleared. Replace the *value* argument with a number in the range from 0 through 180000 milliseconds. The default value is 10 seconds. • **delay trigger** *value* —Set SONET path delay values or delay trigger value. Replace the *value* argument with a number in the range from 0 through 60000 milliseconds. The default value is 0 milliseconds. • **overhead** [**c2** *byte-value* | **j1** *line*] —Set SONET POH byte or bit values. Enter the **c2** keyword to specify STS SPE content (C2) byte, and replace the *byte-value* argument with a number in the range from 0 through 255. Enter the **j1** keyword to configure the SONET path trace (J1) buffer, and replace the *line* argument with the path trace buffer identifier (in ASCII text). • **report [b3-tca | pais | plop | pplm | prdi | ptim]**—Set SONET path alarm reporting. Specifies which alarms are reported and which bit error rate (BER) thresholds will signal an alarm. By default, B3 BER threshold crossing alert (TCA) and path loss of pointer (PLOP) reporting are enabled. Specifying the **pais** keyword sets PAIS reporting status; **pplm** sets path payload mismatch (PPLM) defect reporting status; **prdi** sets path remote defect indication reporting status; and **ptim** sets path trace identity mismatch (PTIM) defect reporting status. The **no report b3-tca** and **no report plop** commands in SONET/SDH path configuration submode disable B3 BER TCA and PLOP reporting status, respectively. • **scrambling disable**—Disable SPE scrambling. Note that SPE scrambling is enabled by default.

	Command or Action	Purpose
		• **threshold b3-tca** *BER*—Set SONET path BER threshold value. Replace the *BER* argument with a number in the range from 3 through 9. The threshold value is interpreted as a negative exponent of 10 when determining the bit error rate. For example, a value of 5 implies a bit error rate of 10 to the minus 5. The default BER threshold value is 6. • **uneq-shut**—Sets sending Unequipped (UNEQ) when shut down.

Step 10	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-sonet)# end`
	or
	`RP/0/RP0/CPU0:router(config-sonet)# 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.

Step 11	**show controllers sonet** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show controllers sonet 0/1/0/0`	Verifies the SONET controller configuration.

Configuring SONET APS

SONET APS offers recovery from fiber (external) or equipment (interface and internal) failures at the SONET line layer. This task explains how to configure basic automatic protection switching (APS) on the router and how to configure more than one protect or working interface on a router by using the **aps group** command.

To verify the configuration or to determine if a switchover has occurred, use the **show aps** command.

Before you begin

Before you configure SONET APS, be sure that you have a supported POS line card or channelized SPA installed in a router that is running Cisco IOS XR software.

Restrictions

Before you configure SONET APS, consider the following restictions:

- For proper APS operation between two routers, a working line on one router must also be the working line on the other router, and the same applies to the protection line.

SUMMARY STEPS

1. **configure**
2. **aps group** *number*
3. **channel** {**0** | **1**} **local sonet** *interface*
4. Repeat Step 3 for each channel in the group.
5. **exit**
6. **interface loopback** *number*
7. **ipv4 address** *ip-address mask*
8. **exit**
9. Execute any one the following commands:
    - **interface pos** *interface-path-id*
    - **interface serial** *interface-path-id*
10. **ipv4 address** *ip-address mask*
11. Execute any one the following commands:
    - **pos crc** {**16** | **32**}
    - **crc** {**16** | **32**}
12. **encapsulation** {**frame-relay** | **hdlc** | **ppp**}
13. **keepalive** {*interval* | **disable**}[*retry*]
14. **no shutdown**
15. Repeat Step 9 through Step 13 for each channel in the group.
16. **exit**
17. **controller sonet** *interface-path-id*
18. **ais-shut**
19. **path scrambling disable**
20. **clock source** {**internal** | **line**}
21. Repeat Step 16 through Step 19 for each channel in the group.
22. **end** or **commit**
23. **exit**
24. **exit**
25. **show aps**
26. **show aps group** [*number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**aps group** *number* **Example:** `RP/0/RP0/CPU0:router(config)# aps group 1`	Adds an APS group with a specified number and enters APS group configuration mode. • Use the **aps group** command in global configuration mode. • To remove a group, use the **no** form of this command, as in: **no aps group** *number*, where the value range is from 1–255. **Note** • To use the **aps group** command, you must be a member of a user group associated with the proper task IDs for aps commands. • The **aps group** command is used even when a single protect group is configured.
Step 3	**channel** {**0** | **1**} **local sonet** *interface* **Example:** `RP/0/RP0/CPU0:router(config-aps)# channel 0 local SONET 0/0/0/1`	Creates a channel for the APS group. **0** designates a protect channel, and **1** designates a working channel. **Note** • If the protect channel is local, it must be assigned using the **channel** command *before* any of the working channels is assigned.
Step 4	Repeat Step 3 for each channel in the group.	—
Step 5	**exit**	Exits APS group configuration mode and enters global configuration mode.
Step 6	**interface loopback** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface loopback 1`	(Optional) Configures a loopback interface if a two-router APS is desired and enters interface configuration mode for a loopback interface. **Note** • In this example, the loopback interface is used as the interconnect.
Step 7	**ipv4 address** *ip-address mask* **Example:**	Assigns an IPV4 address and subnet mask to the loopback interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.0.1 255.255.255.224`	
Step 8	**exit**	Exits interface configuration mode for a loopback interface, and enters global configuration mode.

Step 9	Execute any one the following commands:
	• **interface pos** *interface-path-id*
	• **interface serial** *interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config)# interface POS 0/2/0/0`
	`or`
	`RP/0/RP0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0`
		Connects the interface for the channel selected in Step 3, and enters interface configuration mode.
		For serial interfaces, specifies the complete interface number with the *rack/slot/module/port/T3Num/T1num:instance* notation.

Step 10	**ipv4 address** *ip-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.0.1 255.255.255.224`	Assigns an IPv4 address and subnet mask to the interface.

Step 11	Execute any one the following commands:
	• **pos crc** {**16** | **32**}
	• **crc** {**16** | **32**}
	Example:
	`RP/0/RP0/CPU0:router(config-if)# pos crc 32`
	`or`
	`RP/0/RP0/CPU0:router(config-if)# crc 32`
		Selects a CRC value for the channel. Enter the **16** keyword to specify 16-bit CRC mode, or enter the **32** keyword to specify 32-bit CRC mode. For POS interfaces, the default CRC is 32. For serial interfaces, the default is 16.

Step 12	**encapsulation** {**frame-relay** | **hdlc** | **ppp**} **Example:** `RP/0/RP0/CPU0:router(config-if)# encapsulation ppp`	(Serial interfaces only) Set the Layer 2 encapsulation of an interface.
Step 13	**keepalive** {*interval* | **disable**}[*retry*] **Example:** `RP/0/RP0/CPU0:router(config-if)# keepalive disable`	Sets the keepalive timer for the channel, where: • *interval*—Number of seconds (from 1 to 30) between keepalive messages. The default is 10. • **disable**—Turns off the keepalive timer. • *retry*—(Optional) Number of keepalive messages (from 1 to 255) that can be sent to a peer without a response before transitioning the link to the down state. The default is 5 for interfaces with PPP

	Command or Action	Purpose
		encapsulation, and 3 for interfaces with HDLC encapsulation. The **keepalive** command does not apply to interfaces using Frame Relay encapsulation.
Step 14	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration. • The removal of the shutdown configuration removes the forced administrative down on the interface, enabling that interface to move to an up or down state (assuming the parent SONET layer is not configured administratively down).
Step 15	Repeat Step 9 through Step 13 for each channel in the group.	—
Step 16	**exit**	Exits interface configuration mode, and enters global configuration mode.
Step 17	**controller sonet** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller sonet 0/1/0/0`	Enters SONET controller configuration mode and specifies the SONET controller name and instance identifier with the *rack/slot/module/port* notation.
Step 18	**ais-shut** **Example:** `RP/0/RP0/CPU0:router(config-sonet)# ais-shut`	Configures SONET path values such as alarm indication signal (AIS) at shut down.
Step 19	**path scrambling disable** **Example:** `RP/0/RP0/CPU0:router(config-sonet)# path scrambling disable`	(Optional) Disables synchronous payload envelope (SPE) scrambling. **Note** • SPE scrambling is enabled by default.
Step 20	**clock source** {**internal** | **line**} **Example:** `RP/0/RP0/CPU0:router(config-sonet)# clock source internal`	Configures the SONET port TX clock source, where the **internal** keyword sets the internal clock and the **line** keyword sets the clock recovered from the line. • Use the **line** keyword whenever clocking is derived from the network; use the **internal** keyword when two routers are connected back-to-back or over fiber for which no clocking is available. • The line clock (**line**) is the default.
Step 21	Repeat Step 16 through Step 19 for each channel in the group.	—

	Command or Action	Purpose

Step 22	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-sonet)# end`
	or
	`RP/0/RP0/CPU0:router(config-sonet)# 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.

Step 23	**exit**	Exits SONET controller configuration mode, and enters global configuration mode.
Step 24	**exit**	Exits global configuration mode, and enters EXEC mode.
Step 25	**show aps** **Example:** `RP/0/RP0/CPU0:router# show aps`	(Optional) Displays the operational status for all configured SONET APS groups.
Step 26	**show aps group** [*number*] **Example:** `RP/0/RP0/CPU0:router# show aps group 3`	(Optional) Displays the operational status for configured SONET APS groups. **Note** • The **show aps group** command is more useful than the **show aps** command when multiple groups are defined.

Configuring a Hold-off Timer to Prevent Fast Reroute from Being Triggered

When APS is configured on a router, it does not offer protection for tunnels; because of this limitation, fast reroute (FRR) still remains the protection mechanism for Multiprotocol Label Switching (MPLS) traffic-engineering.

When APS is configured in a SONET core network, an alarm might be generated toward a router downstream. If the router downstream is configured with FRR, you may want to configure a hold-off timer at the SONET level to prevent FRR from being triggered while the CORE network is doing a restoration. Perform this task to configure the delay.

Before you begin

Configure SONET APS, as describe in the *Configuring SONET APS* section.

SUMMARY STEPS

1. **configure**
2. **controller sonet** *interface-path-id*
3. **line delay trigger** *value* or **path delay trigger** *value*
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller sonet** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller sonet 0/6/0/0`	Enters SONET configuration mode.

Step 3	**line delay trigger** *value* or **path delay trigger** *value*
	Example:
	`RP/0/RP0/CPU0:router(config-sonet)# line delay trigger 250`
	`or`
	`RP/0/RP0/CPU0:router(config-sonet)# path delay trigger 300`
		Configures SONET port delay trigger values in milliseconds.
		• Tip The commands in Step 2 and Step 3 can be combined in one command string and entered from global configuration mode like this: **controller sonet** *r/s/m/p* **line delay trigger** or **controller sonet** *r/s/m/p* **path delay trigger**.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-sonet)# end`
	or
	`RP/0/RP0/CPU0:router(config-sonet)# 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.

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

Configuration Examples for SONET Controllers

This section contains the following examples:

SONET Controller Configuration: Example

The following example shows the commands and output generated when you are performing the configuration of a SONET controllers following the steps outlined in the *Configuring a Clear Channel SONET Controller* section. This example shows the usage of every optional command, along with listings of options within commands where relevant. An actual configuration may or may not include all these commands.

```
configure
controller sonet 0/1/0/0
 ais-shut
 clock source internal
 framing sonet
 loopback internal
Loopback is a traffic-effecting operation
 overhead s1s0 1
 path ais-shut
 path delay trigger 0
 path overhead j1 line l1
 path report pais
 path scrambling disable
 path threshold b3-tca 6
 path uneq-shut
 report pais
 threshold b2-tca 4
 commit
```

SONET APS Group Configuration: Example

The following example shows SONET Local (one router) APS configuration.

```
aps group 1
 channel 0 local SONET 0/0/0/1
 channel 1 local SONET 0/0/0/2
 signalling sonet
 commit
```

```
show aps
show aps group 3
```

The following example shows SONET Remote (two routers) APS configuration.

```
RP/0/0/CPU0:router(config)# aps group 1
 channel 0 local SONET 0/0/0/1
 channel 1 remote 172.18.69.123
 signalling sonet
 commit
show aps
show aps group 3
RP/0/0/CPU0:router(config)#
```

CHAPTER 11

Configuring Clear Channel T3/E3 Controllers and Channelized T3 and T1/E1 Controllers

This module describes the configuration of clear channel T3/E3 controllers .

You must configure the T3/E3 controller before you can configure an associated serial interface.

Feature History for Configuring T3/E3 Controller Interfaces

Release	Modification
Release 3.4.1	This feature was introduced on the Cisco CRS-1 Router for the 4-Port Clear Channel T3/E3 SPA.

Prerequisites for Configuring T3/E3 Controllers

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 T3/E3 controllers, be sure that you have one of the following supported SPAs installed in the router:

- Cisco 2-Port and 4-Port Clear Channel T3/E3 SPA

Information About T3/E3 Controllers and Serial Interfaces

The 2-Port and 4-Port Clear Channel T3/E3 SPAs support clear channel services over serial lines only.

If a controller is not channelized, then it is a clear channel controller, and the full bandwidth of its associated serial line is dedicated to a single channel that carries serial services.

When a T3 controller is channelized, it is logically divided into smaller bandwidth T1 or E1 controllers, depending on which mode of channelization you select. The sum of the bandwidth of the serial interfaces on the T1 or E1 controllers cannot exceed the bandwidth of the T3 controller that contains those channelized T1 or E1 controllers.

When you channelize a T3 controller, each individual T1 or E1 controller is automatically further channelized into DS0 time slots. A single T1 controller carries 24 DS0 time slots, and a single E1 controller carries 31 DS0 time slots. Users can divide these DS0 time slots up into individual channel groups. Each channel group can support a single serial interface.

When a controller is channelized, and channel groups have been created, services are provisioned on the associated serial interfaces.

The channelization feature in this release allows the following types of channelization:

- A single T3 controller into 28 T1 controllers, for a total controller size of 44210 kbps.
- A single T3 controller into 21 E1 controllers, for a total controller size of 43008 kbps.
- A single T1 controller supports up to 1.536 MB.
- A single E1 controller supports up to 2.048 MB.

Note A single shared port adapter (SPA) can support up to 448 channel groups.

This section includes the following additional topics:

6 simultaneous BERT sessions among first three physical ports and 6 simultaneous BERT sessions on 4th port.

All interfaces configured on a SONET/SDH controller for the 1-Port Channelized OC-3/STM-1 SPA should be IC-SSO protected or none of them should be IC-SSO protected.

For detailed information about loopback support, see the “Loopback Support” section.

Configuration Overview

Configuring a channelized T3 controller and its associated serial interfaces is a 4-step process:

1. Configure the T3 controller, and set the mode for that controller to T1 or E1.
2. Configure the T1 or E1 controller.
3. Create channel groups and assign DS0 time slots to these channel groups as desired.
4. Configure the serial interfaces that are associated with the individual channel groups, as described in the Configuring Serial Interfaces, on page 371 module later in this document.

Default Configuration Values for T3 and E3 Controllers

This table describes the default configuration parameters that are present on the T3 and E3 controllers.

Note
-
-

Table 8: T3 and E3 Controller Default Configuration Values

Parameter	Default Value	Configuration File Entry
Frame type for the data line	For T3: C-bit framing For E3: G.751	**framing** {**auto-detect** | **c-bit** | **m23**}
Clocking for individual T3/E3 links	**internal**	**clock source** {**internal** | **line**}
Cable length	224 feet	**cablelength** *feet*
Maintenance data link (MDL) messages (T3 only)	**disable**	**mdl transmit** {**idle-signal** | **path** | **test-signal**} {**disable** | **enable**}
National reserved bits for an E3 port (E3 only)	**enable**, and the bit pattern value is 1.	**national bits** {**disable** | **enable**}

Note When configuring clocking on a serial link, you must configure one end to be **internal**, and the other end to be **line**. If you configure **internal** clocking on both ends of a connection, framing slips occur. If you configure **line** clocking on both ends of a connection, the line does not come up.

How to Configure Clear Channel T3/E3 Controllers

The T3/E3 controllers are configured in the physical layer control element of the Cisco IOS XR software configuration space. This configuration is described in the following tasks:

Setting the Card Type

By default, the 2-Port and 4-Port Clear Channel T3/E3 SPAs boot in T3 mode. If you want to use the 2-Port or 4-Port Clear Channel T3/E3 SPA in E3 mode, you must change the default setting of the **hw-module subslot card type** command as described in this section.

Note The **hw-module subslot card type** command configures all ports on the SPA to be the same type.

Caution The SPA is automatically reset when the **hw-module subslot card type** command is committed.

Note The **hw-module subslot card type** command applies to the 2-Port and 4-Port Clear Channel T3/E3 SPAs only. The 2-Port and 4-Port Channelized T3 SPA runs in T3 mode only.

Before you begin

If you have previously configured the interfaces on the 2-Port or 4-Port Clear Channel T3/E3 SPA and now you want to change the card type, you must delete any previously defined T3/E3 controller and serial interface configurations. Use the **no controller** [**e1** and **no interface serial** commands to revert the controller and interface configurations to their defaults.

Restrictions

This task is applicable to 2-Port and 4-Port Clear Channel T3/E3 SPAs only.

SUMMARY STEPS

1. **configure**
2. **hw-module subslot** *subslot-id* **cardtype** {**e1**}
3. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module subslot** *subslot-id* **cardtype** {**e1**} **Example:** `RP/0/RP0/CPU0:router(config)# hw-module subslot 0/1/0 cardtype e3`	Sets the serial mode for the SPA. • t3—Specifies T3 connectivity of 44,210 kbps through the network, using B3ZS coding. This is the default setting. • e3—Specifies a wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 34,010 kbps.
Step 3	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config)# end` or	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]:`

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# commit`	- 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 a Clear Channel E3 Controller

Before you begin

You must first use the **hw-module subslot cardtype** command to set the card to support E3.

Restrictions

- If you configure an option that is not valid for your controller type, you receive an error when you commit the configuration.
- A single SPA cannot support a mixture of T3 and E3 interfaces.
- This task is applicable to 2-Port and 4-Port Clear Channel T3/E3 SPAs only.

SUMMARY STEPS

1. **configure**
2. **controller e3** *interface-path-id*
3. **mode serial**
4. **no shutdown**
5. **end** or **commit**
6. **show controllers e3** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**controller e3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0`	Specifies the E3 controller name in the notation *rack/slot/module/port* and enters E3 configuration mode.
Step 3	**mode serial** **Example:** `RP/0/RP0/CPU0:router(config-e3)# mode serial`	Configures the mode of the port to be clear channel serial. **Note** • This step is required for the 2-Port and 4-Port Channelized T3 SPA only. The 2-Port and 4-Port Clear Channel T3/E3 SPA run in serial mode by default.
Step 4	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-e3)# no shutdown`	Removes the shutdown configuration. • The removal of the shutdown configuration removes the forced administrative down on the controller, enabling the controller to move to an up or a down state.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-e3)# end`
	or
	`RP/0/RP0/CPU0:router(config-e3)# 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.

Step 6	**show controllers e3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show controllers e3 0/1/0/0`	(Optional) Displays information about the E3 controllers.

What to do next

- Modify the default configuration that is running on the E3 controller you just configured, as described in the "Modifying the Default E3 Controller Configuration" section later in this module.
- Configure a bit error rate test (BERT) on the controller to test its integrity, as described in the "Configuring BERT" section later in this module.
- Configure the associated serial interface, as described in the Configuring Serial Interfaces on the Cisco ASR 9000 Series Router module later in this document.

Modifying the Default E3 Controller Configuration

This task explains how to modify the default E3 controller configuration, which is described in the "Default Configuration Values for T3 and E3 Controllers" section earlier in this module.

SUMMARY STEPS

1. **configure**
2. **controller e3** *interface-path-id*
3. **clock source {internal | line}**
4. **cablelength** *feet*
5. **framing {g751 | g832}**
6. **national bits** {**disable** | **enable**}
7. **no shutdown**
8. **end** or **commit**
9. **show controllers e3** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller e3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0`	Specifies the E3 controller name in the notation *rack/slot/module/port* and enters E3 configuration mode.
Step 3	**clock source {internal | line}** **Example:**	(Optional) Sets the clocking for individual E3 links.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-e3)# clock source internal`	**Note** • The default clock source is **internal**. • When configuring clocking on a serial link, you must configure one end to be **internal**, and the other end to be **line**. If you configure **internal** clocking on both ends of a connection, framing slips occur. If you configure **line** clocking on both ends of a connection, the line does not come up.
Step 4	**cablelength** *feet* **Example:** `RP/0/RP0/CPU0:router(config-e3)# cablelength 250`	(Optional) Specifies the distance of the cable from the router to the network equipment. **Note** • The default cable length is 224 feet.
Step 5	**framing {g751 | g832}** **Example:** `RP/0/RP0/CPU0:router(config-e3)# framing g832`	(Optional) Selects the frame type for the E3 port. Possible E3 frame types are G.751 and G.832. **Note** • The default framing for E3 is G.751.
Step 6	**national bits** {**disable** | **enable**} **Example:** `RP/0/RP0/CPU0:router(config-e3)# national bits enable`	(Optional) Enables or disables the 0x1F national reserved bit pattern on the E3 port. **Note** • The E3 national bit is enabled by default, and the bit pattern value is 1.
Step 7	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-e3)# no shutdown`	Removes the shutdown configuration. • The removal of the shutdown configuration removes the forced administrative down on the controller, enabling the controller to move to an up or a down state.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-e3)# end`
	or
	`RP/0/RP0/CPU0:router(config-e3)# 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.

	Command or Action	Purpose
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 9	**show controllers e3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show controllers e3 0/1/0/0`	(Optional) Displays information about the E3 controllers.

What to do next

- Modify the default configuration that is running on the T3 controller you just configured, as described in the "Modifying the Default T3 Controller Configuration" section later in this module.
- Configure BERT on the controller to test its integrity, as described in the "Configuring BERT" section later in this module.
- Configure the associated serial interface, as described in the *Configuring Serial Interfaces on the Cisco ASR 9000 Series Router* module later in this document.

Configuring a Clear Channel T3 Controller

Before you begin

You must use the **hw-module subslot cardtype** command to set the card to support T3, as described in the Setting the Card Typesection on earlier in this module.

Restrictions

- This task is applicable to 2-Port and 4-Port Clear Channel T3/E3 SPAs only.
- If you configure an option that is not valid for your controller type, you receive an error when you commit the configuration.
- A single SPA cannot support a mixture of T3 and E3 interfaces.

SUMMARY STEPS

1. **configure**
2. **controller t3** *interface-path-id*
3. **mode serial**
4. **no shutdown**
5. **end** or **commit**
6. **show controllers t3** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller t3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0`	Specifies the T3 controller name in the *rack/slot/module/port* notation and enters T3 configuration mode.
Step 3	**mode serial** **Example:** `RP/0/RP0/CPU0:router(config-t3)# mode serial`	Configures the mode of the port to be clear channel serial. **Note** • This step is required for the 2-Port and 4-Port Channelized T3 SPA only. The 2-Port and 4-Port Clear Channel T3/E3 SPA runs in serial mode by default.
Step 4	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-t3)# no shutdown`	Removes the shutdown configuration. • The removal of the shutdown configuration removes the forced administrative down on the controller, enabling the controller to move to an up or a down state.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-t3)# end`
	or
	`RP/0/RP0/CPU0:router(config-t3)# 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.

	Command or Action	Purpose
Step 6	**show controllers t3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show controllers t3 0/1/0/0`	(Optional) Displays information about the T3 controllers.

What to do next

- Modify the default configuration that is running on the T3 controller you just configured, as described in the “Modifying the Default T3 Controller Configuration” section later in this module.
- Configure BERT on the controller to test its integrity, as described in the “Configuring BERT” section later in this module.
- Configure the associated serial interface, as described in the Configuring Serial Interfaces on the Cisco ASR 9000 Series Router module.
- When a T3 controller is in clear channel mode, it carries a single serial interface.
- The T3 controllers are configured in the T3 configuration mode.

Modifying the Default T3 Controller Configuration

This task explains how to modify the default T3 controller configuration, which is described in the Default Configuration Values for T3 and E3 Controllers section.

Before you begin

You must configure a clear channel controller, as described in one of the following sections:

- Configuring a Clear Channel T3 Controller

SUMMARY STEPS

1. **configure**
2. **controller T3** *interface-path-id*
3. **clock source {internal | line}**
4. **cablelength** *feet*
5. **framing** {**auto-detect** | **c-bit** | **m23**}
6. **mdl transmit** {**idle-signal** | **path** | **test-signal**} {**disable** | **enable**}
7. **mdl string** {**eic** | **fi** | **fic** | **gen-number** | **lic** | **port-number** | **unit**} *string*
8. **no shutdown**
9. **end** or **commit**
10. **show controllers t3** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller T3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0`	Specifies the T3 controller name in the notation *rack/slot/module/port* and enters T3 configuration mode.
Step 3	**clock source {internal | line}** **Example:** `RP/0/RP0/CPU0:router(config-t3)# clock source internal`	(Optional) Sets the clocking for the T3 port. **Note** • The default clock source is **internal**. • When configuring clocking on a serial link, you must configure one end to be **internal**, and the other end to be **line**. If you configure **internal** clocking on both ends of a connection, framing slips occur. If you configure **line** clocking on both ends of a connection, the line does not come up.
Step 4	**cablelength** *feet* **Example:** `RP/0/RP0/CPU0:router(config-t3)# cablelength 250`	(Optional) Specifies the distance of the cable from the router to the network equipment. **Note** • The default cable length is 224 feet.
Step 5	**framing** {**auto-detect** | **c-bit** | **m23**} **Example:** `RP/0/RP0/CPU0:router(config-t3)# framing c-bit`	(Optional) Selects the frame type for the T3 port. **Note** • The default frame type for T3 is C-bit.
Step 6	**mdl transmit** {**idle-signal** | **path** | **test-signal**} {**disable** | **enable**} **Example:** `RP/0/RP0/CPU0:router(config-t3)# mdl transmit path enable`	(Optional) Enables Maintenance Data Link (MDL) messages on the T3 port. **Note** • MDL messages are supported only when the T3 framing is C-bit parity. • MDL message are disabled by default.
Step 7	**mdl string** {**eic** | **fi** | **fic** | **gen-number** | **lic** | **port-number** | **unit**} *string* **Example:** `RP/0/RP0/CPU0:router(config-t3)# mdl fi facility identification code`	(Optional) Specifies the values of the strings sent in the MDL messages.

	Command or Action	Purpose
Step 8	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-t3)# no shutdown`	Removes the shutdown configuration. • The removal of the shutdown configuration removes the forced administrative down on the controller, enabling the controller to move to an up or a down state.

Step 9	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-t3)# end`
	or
	`RP/0/RP0/CPU0:router(config-t3)# 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.

Step 10	**show controllers t3** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show controllers t3 0/1/0/0`	(Optional) Displays information about the T3 controllers.

What to do next

- If you configured a clear channel T3 controller, perform the following tasks:
  - Configure BERT on the controller to test its integrity, as described in the "Configuring BERT" section on page 474 later in this module.
  - Configure the associated serial interface, as described in the *Configuring Serial Interfaces on the Cisco ASR 9000 Series Router* module.

Configuring BERT

Depending on your hardware support, BERT is supported on each of the T3/E3 . It is done only over an unframed T3/E3 signal and is run on only one port at a time. It is also supported on individual channel groups.

To view the BERT results, use the **show controllers t3** command in EXEC mode. The BERT results include the following information:

- Type of test pattern selected
- Status of the test
- Interval selected
- Time remaining on the BER test
- Total bit errors
- Total bits received

BERT is data intrusive. Regular data cannot flow on a line while the test is in progress. The line is put in an alarm state when BERT is in progress and restored to a normal state after BERT has been terminated.

Configuring BERT on T3/E3 Controllers

This task explains how to enable a bit error rate test (BERT) pattern on a T3/E3 line.

You must have configured a clear channel T3/E3 controller.

Restrictions

Valid patterns for all controllers and channel groups include: 0s, 1s, 2^15, 2^20, 2^20-QRSS, 2^23, and alt-0-1.

Additional valid patterns for T1 controllers include: 1in8, 3in24, 55Daly, and 55Octet. Additional valid patterns for channel groups include: 2^11 and 2^9.

Before you begin

You must have configured a clear channel T3/E3 controller.

SUMMARY STEPS

1. **configure**
2. **controller [t3 | e3**] *interface-path-id*
3. **bert pattern** *pattern*
4. **bert interval** *time*
5. **bert error** [*number*]
6. **end** or **commit**
7. **exit**
8. **exit**
9. **bert [t3 | e3**] *interface-path-id* [**channel-group** *channel-group-number]* [**error**] **start**
10. **bert [t3 | e3**] *interface-path-id* [**channel-group** *channel-group-number]* **stop**
11. **show controllers** [**t3 | e3**] *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller [t3 | e3]** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0`	Specifies the controller name and instance in the notation *rack/slot/module/por*t, and enters T3 or E3 controller configuration mode.
Step 3	**bert pattern** *pattern* **Example:** `RP/0/RP0/CPU0:router(config-t3)# bert pattern 2^15`	Enables a specific bit error rate test (BERT) pattern on a controller. **Note** You must use the **bert** command in EXEC mode to start the BER test.
Step 4	**bert interval** *time* **Example:** `RP/0/RP0/CPU0:router(config-t3)# bert pattern 2^15`	(Optional) Specifies the duration of a bit error rate test (BERT) pattern on a T3/E3 or T1 line. The interval can be a value from 1 to 14400.
Step 5	**bert error** [*number*] **Example:** `RP/0/RP0/CPU0:router(config-t3)# bert error 10`	Specifies the number of BERT errors to introduce into the bit stream. Range is from 1 to 255.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-t3)# end`
	or
	`RP/0/RP0/CPU0:router(config-t3)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router(config-t3)# exit`	Exits T3/E3 or T1 controller configuration mode.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode.

Step 9	**bert [t3 | e3]** *interface-path-id* [**channel-group** *channel-group-number]* [**error**] **start**
	Example:
	`RP/0/RP0/CPU0:router# bert t3 0/3/0/0 start`
	`RP/0/RP0/CPU0:router# bert t3 0/3/0/0 error`
		Starts the configured BERT test on the specified T3/E3 or T1 controller.
		Note • You can include the optional **error** keyword to inject errors into the running BERT stream.

Step 10	**bert [t3 | e3]** *interface-path-id* [**channel-group** *channel-group-number]* **stop** **Example:** `RP/0/RP0/CPU0:router# bert t3 0/3/0/0 stop`	Stops the configured BERT test on the specified T3/E3 or T1 controller.
Step 11	**show controllers** [**t3 | e3**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show controllers t3 0/3/0/0`	Displays the results of the configured BERT. Configure the serial interfaces that are associate with the controllers you tested, as described in the Configuring Serial Interfaces on the Cisco ASR 9000 Series Router module.

What to do next

Configure the serial interfaces that are associate with the controllers you tested, as described in the *Configuring Serial Interfaces on the Cisco ASR 9000 Series Router* module.

Configuration Examples

This section contains the following examples:

Configuring a Clear Channel T3 Controller: Example

The following example shows configuration for a clear channel T3 controller:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)#controller T3 0/3/2/0
RP/0/RP0/CPU0:router(config-t3)#clock source internal
RP/0/RP0/CPU0:router(config-t3)#mode serial
RP/0/RP0/CPU0:router(config-t3)#cablelength 4
RP/0/RP0/CPU0:router(config-t3)#framing c-bit
RP/0/RP0/CPU0:router(config-t3)#commit
```

Configuring BERT on a T3 Controller: Example

The following example shows how to configure a BERT on a T3 controller, and then display the results of the BERT:

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# controller t3 0/3/0/1
RP/0/RP0/CPU0:router(config-t3)# bert pattern 0s

Run bert from exec mode for the bert config to take effect

RP/0/RP0/CPU0:router(config-t3)#exit
RP/0/RP0/CPU0:router(config)# exit

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]
RP/0/RP0/CPU0:router# bert t3 0/3/0/1 start


RP/0/RP0/CPU0:router# bert t3 0/3/0/1 stop


RP/0/RP0/CPU0:router# show controllers t3 0/3/0/1

  T30/3/0/1 is up
  No alarms detected.
  MDL transmission is disabled
    EIC: , LIC: , FIC: , UNIT:
     Path FI:
     Idle Signal PORT_NO:
     Test Signal GEN_NO:
  FEAC code received: No code is being received
  Framing is C-BIT Parity, Line Code is B3ZS, Clock Source is Internal
  Data in current interval (108 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
  Data in Interval 2:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
  Data in Interval 3:
     0 Line Code Violations, 0 P-bit Coding Violation
```

```
0 C-bit Coding Violation, 0 P-bit Err Secs
0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
0 Unavailable Secs, 0 Line Errored Secs
0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
```

CHAPTER 12

Configuring Dense Wavelength Division Multiplexing Controllers

This module describes the configuration of dense wavelength division multiplexing (DWDM) controllers.

DWDM is an optical technology that is used to increase bandwidth over existing fiber-optic backbones. DWDM can be configured on supported 10-Gigabit Ethernet (GE) or Packet-over-SONET/SDH physical layer interface modules (PLIMs). After you configure the DWDM controller, you can configure an associated POS or 10-Gigabit Ethernet interface.

Feature History for Configuring DWDM Controller Interfaces

Release	Modification
Release 3.3.0	This feature was introduced on theCisco CRS-1 Router. Support was added for the Cisco 1-Port OC-768c/STM-256c DWDM PLIM and Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM.
Release 3.4.0	Support was added for user configuration of the laser, TTI strings, and BDI insertion, as well as performance monitoring.
Release 3.9.1	Support for IPoDWDM was added.

Configuring Dense Wavelength Division Multiplexing Controllers

This module describes the configuration of dense wavelength division multiplexing (DWDM) controllers.

DWDM is an optical technology that is used to increase bandwidth over existing fiber-optic backbones. DWDM can be configured on supported 10-Gigabit Ethernet (GE) or Packet-over-SONET/SDH physical layer interface modules (PLIMs). After you configure the DWDM controller, you can configure an associated POS or 10-Gigabit Ethernet interface.

Feature History for Configuring DWDM Controller Interfaces

Release	Modification
Release 3.3.0	This feature was introduced on theCisco CRS-1 Router. Support was added for the Cisco 1-Port OC-768c/STM-256c DWDM PLIM and Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM.
Release 3.4.0	Support was added for user configuration of the laser, TTI strings, and BDI insertion, as well as performance monitoring.
Release 3.9.1	Support for IPoDWDM was added.

Prerequisites for Configuring DWDM Controller Interfaces

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 a DWDM controller, be sure that you have installed one of these cards that support DWDM:

- Cisco 1-Port OC-768c/STM-256c DWDM PLIM
- Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM

New DWDM Configuration Requirement

Cisco IOS XR Software Release 3.9.0 introduces new commands in addition to an important change to the default laser state for all of the DWDM physical layer interface modules (PLIMs) supported on the Cisco CRS-1 router, which impacts the required configuration to support those cards.

This change affects all models of the following hardware on the Cisco CRS-1 router:

- Cisco 1-Port OC-768c/STM-256c DWDM PLIM
- Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM

Summary of Important DWDM Changes in Cisco IOS XR Software Release 3.9.0 and Later Releases

- The **laser off** and **shutdown (DWDM)** commands are replaced by the **admin-state out-of-service** command.
- The default state of the laser has changed from "On" to "Off" for all PLIMs. Therefore, the laser for all DWDM controllers must explicitly be turned on using the **admin-state in-service** command in DWDM configuration mode.

Information About the DWDM Controllers

DWDM support in Cisco IOS XR software is based on the Optical Transport Network (OTN) protocol that is specified in ITU-T G.709. This standard combines the benefits of SONET/SDH technology with the multiwavelength networks of DWDM. It also provides for forward error correction (FEC) that can allow a reduction in network costs by reducing the number of regenerators used.

Note Configuring two ends of an OTN link with different FEC modes is not supported. Even if different FEC modes are configured, the FEC mismatch alarm will not be raised. Interface may experience continuous port flap in addition to continuous bit interleaved parity (BIP) errors at both OTN and LAN level.

To enable multiservice transport, OTN uses the concept of a wrapped overhead (OH). To illustrate this structure:

- Optical channel payload unit (OPU) OH information is added to the information payload to form the OPU. The OPU OH includes information to support the adaptation of client signals.
- Optical channel data unit (ODU) OH is added to the OPU to create the ODU. The ODU OH includes information for maintenance and operational functions to support optical channels.
- Optical channel transport unit (OTU) OH together with the FEC is added to form the OTU. The OTU OH includes information for operational functions to support the transport by way of one or more optical channel connections.
- Optical channel (OCh) OH is added to form the OCh. The OCh provides the OTN management functionality and contains four subparts: the OPU, ODU, OTU, and frame alignment signal (FAS). See figure below.

Figure 12: OTN Optical Channel Structure

Information about IPoDWDM

Cisco IOS XR software includes the IP over Dense Wavelength Division Multiplexing (IPoDWDM) feature.

IPoDWDM is supported on these hardware devices:

- Cisco 1-Port OC-768c/STM-256c DWDM PLIM
- Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM

The Cisco CRS-1 Series 10 Gigabit Ethernet DWDM PLIM supports the following hardware features:

- Four line-rate 10 Gigabit Ethernet full duplex interfaces
- Per-port flexibility for optical reach - selected using the appropriate XENPAK pluggable optical modules
- Compatible with all Cisco CRS-1 Series chassis
- Supports in-use insertion and removal without the need to power down the chassis
- Simple configuration, monitoring, and maintenance

IPoDWDM currently provides these software features:

- Proactive Maintenance
- Shared Risk Link Group (SRLG)

Proactive Maintenance

Proactive maintenance automatically triggers Forward Error Correction-Fast Re-Route (FEC-FRR). Proactive maintenance requires coordinated maintenance between Layer 0 (L0) and Layer 3 (L3). L0 is the DWDM optical layer. FEC-FRR is an L3 protection mechanism. FEC-FRR detects failures before they happen and corrects errors introduced during transmission or that are due to a degrading signal.

Shared Risk Link Group (SRLG)

The Shared Risk Link Group (SRLG) provides shared risk information between the DWDM optical layer (L0) and the router layer (L3), and the applications that use the shared risk information. An SRLG is a set of links that share a resource whose failure may affect all links in the set.

System administrators can configure the following IPoDWDM features:

- Shared Risk Link Group (SRLG) and Optical Layer DWDM port, see Configuring the Optical Layer DWDM Ports.
- Administrative state of DWDM optical ports, see Configuring the Administrative State of DWDM Optical Ports.
- FEC-FRR trigger threshold, window size, revert threshold, and revert window size, see Configuring Proactive FE-FRR Triggering.

FEC-FRR Triggering

FEC-FRR can be configure to be triggered by the following alarms:

- ais – Alarm Indication Signal (AIS)
- bdi – Backward Defect Indication (BDI)
- *bdiO – Backward Defect Indication - Overhead (BDI-O)
- *bdiP – Backward Defect Indication - Payload (BDI-P)

- *deg – Degraded (DEG)
- lck – Locked (LCK)
- lof – Loss of Frame (LOF)
- lom – Loss of Multi Frame
- los – Loss of Signal (LOS)
- *losO – Loss of Signal - Overhead (LOS-O)
- *losP – Loss of Signal - Payload (LOS-P)
- oci – Open Connection Indication (OCI)
- plm – Payload Mismatch (PLM)
- *ssf – Server Signal Failure (SSF)
- *ssfO – Server Signal Failure - Overhead (SSF-O)
- *ssfP – Server Signal Failure - Payload (SSF-P)
- tim – Trace Identifier Mismatch (TIM)

Signal Logging

DWDM statistic data, such as EC, UC and alarms, are collected and stored in the log file on the DWDM line card.

How to Configure DWDM Controllers

The DWDM controllers are configured in the physical layer control element of the Cisco IOS XR software configuration space. This configuration is done using the **controller dwdm** command, and is described in the following tasks:

Note All interface configuration tasks for the POS or Gigabit Ethernet interfaces still must be performed in interface configuration mode.

Configuring the Optical Parameters

This task describes how to configure the receive power threshold and the wavelength parameters for the DWDM controller. You should verify that the optical parameters are configured correctly for your DWDM installation and if necessary, perform this task.

Before you begin

The **rx-los-threshold**, **wavelength** and **transmit-power** commands can be used only when the controller is in the shutdown state. Use the **shutdown** command.

Restrictions

The transmit power level and receive LOS threshold are configurable only on the Cisco Cisco 1-Port OC-768c/STM-256c DWDM PLIM.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **admin-state {maintenance | out-of-service}**
4. **commit**
5. **rx-los-threshold** *power-level*
6. **wavelength** *channel-number*
7. **transmit-power** *power-level*
8. **end** or **commit**
9. **admin-state in-service**
10. **show controllers dwdm** *interface-path-id* **[optics | wavelength-map]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:Router# configure`	Enters global configuration mode.
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/0`	Specifies the DWDM controller name in the notation *rack/slot/module/port* and enters DWDM configuration mode.
Step 3	**admin-state {maintenance | out-of-service}** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# admin-state maintenance`	Specifies the DWDM interface administrative state. You must put the controller in **maintenance** or **out-of-service** state before you can use the DWDM configuration commands.
Step 4	**commit** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# commit`	Saves configuration changes. This performs the shutdown from the previous step. When the controller has been shut down, you can proceed with the configuration.
Step 5	**rx-los-threshold** *power-level* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# rx-los-threshold -10`	Configures the transponder receive power threshold. Values are in units of 0.1 dBm and can range from -350 to 50. This corresponds to a range of -35 dBm to 5 dBm.

	Command or Action	Purpose
Step 6	**wavelength** *channel-number* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# wavelength 1`	Configures the channel number corresponding to the first wavelength. Values can range from 1 to 185, but not all channels are supported on all PLIMs. Use the **show controller dwdm** command with the **wavelength-map** keyword to determine which channels and wavelengths are supported on a specific controller. **Note** • There is no cross-checking to determine if the chosen wavelength is being used on another port on the same PLIM or on another PLIM in the system.
Step 7	**transmit-power** *power-level* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# transmit-power 10`	Configures the transponder transmit power. Values are in units of 0.1 dBm and can range from -190 to +10. This corresponds to a range of -19 dBm to +1 dBm.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# end`
	or
	`RP/0/RP0/CPU0:Router(config-dwdm)# 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.

Step 9	**admin-state in-service** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# admin-state in-service`	Places the DWDM port in In Service (IS) state, to support all normal operation.

	Command or Action	Purpose
Step 10	**show controllers dwdm** *interface-path-id* **[optics | wavelength-map]** **Example:** `RP/0/RP0/CPU0:Router# show controller dwdm 0/1/0/0 optics`	Displays the output power level, input power level, wavelength, and laser bias current monitoring information.

Troubleshooting Tips

You will get an error message if you try to commit configuration changes to the controller when it is in the up state. You must use the **admin-states maintenance** or **admin-states out-of-service** command before you can use the DWDM configuration commands.

Configuring G.709 Parameters

Before you begin

The **g709 disable**, **loopback**, and **g709 fec** commands can be used only when the controller is in the shutdown state. Use the **admin-state** command.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **admin-state maintenance** or **admin-state out-of-service**
4. **commit**
5. **g709 disable**
6. **loopback {internal | line}**
7. **g709 fec** {**disable | standard**}
8. **g709** {**odu | otu**} **report** *alarm* **disable**
9. **g709 otu overhead tti** {**expected | sent**} {**ascii | hex**} *tti-string*
10. **end** or **commit**
11. **admin-state in-service**
12. **show controllers dwdm** *interface-path-id* **g709**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:Router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/0`	Specifies the DWDM controller name in the notation *rack/slot/module/port* and enters DWDM configuration mode.
Step 3	**admin-state maintenance** or **admin-state out-of-service** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# admin-state out-of-service`	Disables the DWDM controller. You must disable the controller before you can use the DWDM configuration commands.
Step 4	**commit** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# commit`	Saves configuration changes. This performs the shutdown from the previous step. When the controller has been shut down, you can proceed with the configuration.
Step 5	**g709 disable** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# g709 disable`	(Optional) Disables the G.709 wrapper. The wrapper is enabled by default. **Note** • The **g709 disable** command is available on the Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM only.
Step 6	**loopback** {**internal** | **line**} **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# loopback internal`	(Optional) Configures the DWDM controller for loopback mode.
Step 7	**g709 fec** {**disable** | **standard**} **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# g709 fec disable`	(Optional) Configures the forward error correction mode (FEC) for the DWDM controller. By default, enhanced FEC is enabled.
Step 8	**g709** {**odu** | **otu**} **report** *alarm* **disable** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# g709 odu bdi disable`	(Optional) Disables the logging of selected optical channel data unit (ODU) alarms or optical channel transport unit (OTU) alarms to the console for a DWDM controller. By default, all alarms are logged to the console.
Step 9	**g709 otu overhead tti** {**expected** | **sent**} {**ascii** | **hex**} *tti-string* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# g709 otu overhead tti expected ascii test OTU 5678`	Configures a transmit or expected Trail Trace Identifier (TTI) that is displayed in the **show controller dwdm** command.

	Command or Action	Purpose

Step 10	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# end`
	or
	`RP/0/RP0/CPU0:Router(config-dwdm)# 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.

Step 11	**admin-state in-service** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# admin-state in-service`	Places the DWDM port in In Service (IS) state, to support all normal operation.
Step 12	**show controllers dwdm** *interface-path-id* **g709** **Example:** `RP/0/RP0/CPU0:Router# show controller dwdm 0/1/0/0 optics`	Displays the G.709 Optical Transport Network (OTN) protocol alarms and counters for Bit Errors, along with the FEC statistics and threshold-based alerts.

What to do next

All interface configuration tasks for the POS or Gigabit Ethernet interfaces still must be performed in interface configuration mode. Refer to the corresponding modules in this book for more information.

This task describes how to customize the alarm display and the thresholds for alerts and forward error correction (FEC). You need to use this task only if the default values are not correct for your installation.

How to Perform Performance Monitoring on DWDM Controllers

Performance monitoring parameters are used to gather, store, set thresholds for, and report performance data for early detection of problems. Thresholds are used to set error levels for each performance monitoring parameter. During the accumulation cycle, if the current value of a performance monitoring parameter reaches or exceeds its corresponding threshold value, a threshold crossing alert (TCA) can be generated. The TCAs provide early detection of performance degradation.

Performance monitoring statistics are accumulated on a 15-minute basis, synchronized to the start of each quarter-hour. They are also accumulated on a daily basis starting at midnight. Historical counts are maintained for thirty-three 15-minute intervals and two daily intervals.

Performance monitoring is described in the following task:

Configuring DWDM Controller Performance Monitoring

This task describes how to configure performance monitoring on DWDM controllers and how to display the performance parameters.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **pm** {**15-min** | **24-hour**} **fec threshold** {**ec-bits** | **uc-words**} *threshold*
4. **pm** {**15-min** | **24-hour**} **optics threshold** {**lbc** | **opr** | **opt**} {**max** | **min**} *threshold*
5. **pm** {**15-min** | **24-hour**} **otn threshold** *otn-parameter threshold*
6. **pm** {**15-min** | **24-hour**} **fec report** {**ec-bits** | **uc-words**} **enable**
7. **pm** {**15-min** | **24-hour**} **optics report** {**lbc** | **opr** | **opt**} {**max-tca** | **min-tca**} **enable**
8. **pm** {**15-min** | **24-hour**} **otn report** *otn-parameter* **enable**
9. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:Router# configure`	Enters global configuration mode.
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/0`	Specifies the DWDM controller name in the notation *rack/slot/module/port* and enters DWDM configuration mode.

Step 3	**pm** {**15-min** | **24-hour**} **fec threshold** {**ec-bits** | **uc-words**} *threshold*
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min fec threshold ec-bits 49000000`
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min fec threshold uc-words xxxxxx`
		Configures a performance monitoring threshold for specific parameters on the FEC layer.

	Command or Action	Purpose

Step 4	**pm** {**15-min** | **24-hour**} **optics threshold** {**lbc** | **opr** | **opt**} {**max** | **min**} *threshold*
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opt max xxx`
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold lbc min xxx`
		Configures a performance monitoring threshold for specific parameters on the optics layer.

Step 5	**pm** {**15-min** | **24-hour**} **otn threshold** *otn-parameter threshold*
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min otn threshold bbe-pm-ne xxx`
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min otn threshold es-sm-fe xxx`
		Configures a performance monitoring threshold for specific parameters on the optical transport network (OTN) layer. OTN parameters can be as follows:
		• **bbe-pm-fe**—Far-end path monitoring background block errors (BBE-PM)
		• **bbe-pm-ne**—Near-end path monitoring background block errors (BBE-PM)
		• **bbe-sm-fe**—Far-end section monitoring background block errors (BBE-SM)
		• **bbe-sm-ne**—Near-end section monitoring background block errors (BBE-SM)
		• **bber-pm-fe**—Far-end path monitoring background block errors ratio (BBER-PM)
		• **bber-pm-ne**—Near-end path monitoring background block errors ratio (BBER-PM)
		• **bber-sm-fe**—Far-end section monitoring background block errors ratio (BBER-SM)
		• **bber-sm-ne**—Near-end section monitoring background block errors ratio (BBER-SM)
		• **es-pm-fe**—Far-end path monitoring errored seconds (ES-PM)
		• **es-pm-ne**—Near-end path monitoring errored seconds (ES-PM)
		• **es-sm-fe**—Far-end section monitoring errored seconds (ES-SM)
		• **es-sm-ne**—Near-end section monitoring errored seconds (ES-SM)
		• **esr-pm-fe**—Far-end path monitoring errored seconds ratio (ESR-PM)
		• **esr-pm-ne**—Near-end path monitoring errored seconds ratio (ESR-PM)

	Command or Action	Purpose
		• **esr-sm-fe**—Far-end section monitoring errored seconds ratio (ESR-SM) • **esr-sm-ne**—Near-end section monitoring errored seconds ratio (ESR-SM) • **fc-pm-fe**—Far-end path monitoring failure counts (FC-PM) • **fc-pm-ne**—Near-end path monitoring failure counts (FC-PM) • **fc-sm-fe**—Far-end section monitoring failure counts (FC-SM) • **fc-sm-ne**—Near-end section monitoring failure counts (FC-SM) • **ses-pm-fe**—Far-end path monitoring severely errored seconds (SES-PM) • **ses-pm-ne**—Near-end path monitoring severely errored seconds (SES-PM) • **ses-sm-fe**—Far-end section monitoring severely errored seconds (SES-SM) • **ses-sm-ne**—Near-end section monitoring severely errored seconds (SES-SM) • **sesr-pm-fe**—Far-end path monitoring severely errored seconds ratio (SESR-PM) • **sesr-pm-ne**—Near-end path monitoring severely errored seconds ratio (SESR-PM) • **sesr-sm-fe**—Far-end section monitoring severely errored seconds ratio (SESR-SM) • **sesr-sm-ne**—Near-end section monitoring severely errored seconds ratio (SESR-SM) • **uas-pm-fe**—Far-end path monitoring unavailable seconds (UAS-PM) • **uas-pm-ne**—Near-end path monitoring unavailable seconds (UAS-PM) • **uas-sm-fe**—Far-end section monitoring unavailable seconds (UAS-SM) • **uas-sm-ne**—Near-end section monitoring unavailable seconds (UAS-SM)

	Command or Action	Purpose

Step 6	**pm** {**15-min** | **24-hour**} **fec report** {**ec-bits** | **uc-words**} **enable**
	
	Example:
	
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min fec report ec-bits enable`
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min fec report uc-words enable`
		Configures threshold crossing alert (TCA) generation for specific parameters on the FEC layer.

Step 7	**pm** {**15-min** | **24-hour**} **optics report** {**lbc** | **opr** | **opt**} {**max-tca** | **min-tca**} **enable**
	
	Example:
	
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report opt enable`
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report lbc enable`
		Configures TCA generation for specific parameters on the optics layer.

Step 8	**pm** {**15-min** | **24-hour**} **otn report** *otn-parameter* **enable**
	
	Example:
	
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min otn report bbe-pm-ne enable`
	`RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min otn report es-sm-fe enable`
		Configures TCA generation for specific parameters on the optical transport network (OTN) layer. OTN parameters are shown in Step 5.

Step 9	**end** or **commit**
	
	Example:
	
	`RP/0/RP0/CPU0:Router(config-dwdm)# end`
	
	or
	
	`RP/0/RP0/CPU0:Router(config-dwdm)# 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 IPoDWDM

This section provides the following configuration procedures:

Configuring the SRLG and Optical Layer DWDM Ports

Use this procedure to configure the Shared Risk Link Group (SRLG) and Optical Layer DWDM ports.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **network srlg** *value1 value2 value3*
4. **network port id** *id-number*
5. **network connection id** *id-number*
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:Router# config`	Enters global configuration mode.
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1`	Specifies the DWDM controller and enters DWDM controller mode.
Step 3	**network srlg** *value1 value2 value3* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# network srlg value1 value2 value3`	Configures the Shared Risk Link Group (SRLG).
Step 4	**network port id** *id-number* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# network port id 1/0/1/1`	Assigns an identifier number to a port for the Multi Service Transport Protocol (MSTP).
Step 5	**network connection id** *id-number* **Example:**	Configures a connection identifier for the Multi Service Transport Protocol (MSTP).

	Command or Action	Purpose
	`RP/0/RP0/CPU0:Router(config-dwdm)# network connection id 1/1/1/1`	

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# end`
	or
	`RP/0/RP0/CPU0:Router(config-dwdm)# 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 the Administrative State of DWDM Optical Ports

Use this procedure to configure the administrative state and optionally set the maintenance embargo flag.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **admin-state** {**in-service** | **maintenance** | **out-of-service**}
4. **exit**
5. **interface pos** *interface-path-id*
6. or
7. **interface tengige** *interface-path-id*
8. **maintenance disable**
9. **end** or **commit**

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:Router# config`	
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:Routerconfig)# controller dwdm 0/1/0/1`	Specifies the DWDM controller and enters DWDM controller mode.
Step 3	**admin-state** {**in-service** | **maintenance** | **out-of-service**} **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# admin-state maintenance`	Specifies the transport administration state.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# exit`	Exits to the previous mode.
Step 5	**interface pos** *interface-path-id*	
Step 6	or	

Step 7	**interface tengige** *interface-path-id*
	Example:
	`RP/0/RP0/CPU0:Router(config)# interface pos 1/0/1/1`
	`or`
	`RP/0/RP0/CPU0:Router(config)# interface tengige 1/0/1/1`
		Specifies the interface and enters interface configuration mode.

Step 8	**maintenance disable** **Example:** `RP/0/RP0/CPU0:Router(config-if)# maintenance disable`	Provisions the maintenance embargo flag, which prevents maintenance activities from being performed on an interface.

Step 9	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# end`
	or
	`RP/0/RP0/CPU0:Router(config-dwdm)# 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.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Proactive FEC-FRR Triggering

Use this procedure to configure automatic triggering of Forward Error Correction-Fast Re-Route (FEC-FRR).

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **proactive**
4. **logging signal** *file-name*
5. **proactive trigger threshold** *x-coefficient y-power*
6. **proactive trigger window** *window*
7. **proactive revert threshold** *x-coefficient y-power*
8. **proactive revert window** *window*
9. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:Router# config`	Enters global configuration mode.
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1`	Specifies the DWDM controller and enters DWDM controller mode.
Step 3	**proactive** **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# proactive enable`	Enables automatic triggering of FEC-FRR.
Step 4	**logging signal** *file-name* **Example:**	Enables10 millisecond proactive monitoring of FEC-FRR.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:Router(config-dwdm)# logging signal LogFile1`	
Step 5	**proactive trigger threshold** *x-coefficient y-power* **Example:** `RP/0/RP0/CPU0:Routerconfig-dwdm)# proactive trigger threshold 1 9`	Configures the trigger threshold of FEC-FRR in the form of *x*E-*y*.
Step 6	**proactive trigger window** *window* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# proactive trigger window 10000`	Configures the trigger window (in milliseconds) in which FRR may be triggered.
Step 7	**proactive revert threshold** *x-coefficient y-power* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# proactive revert threshold 1 9`	Configures the revert threshold (in the form of *x*E-*y*) to trigger reverting from the FEC-FRR route back to the original route.
Step 8	**proactive revert window** *window* **Example:** `RP/0/RP0/CPU0:Router(config-dwdm)# proactive revert window 600000`	Configures the revert window in which reverting from the FEC-FRR route back to the original route is triggered.

Step 9	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:Router(config-dwdm)# end`
	or
	`RP/0/RP0/CPU0:Router(config-dwdm)# 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.

Configuration Examples

This section includes these examples:

Turning On the Laser: Example

Note This is a required configuration beginning in Cisco IOS XR Software Release 3.9.0. The DWDM cards will not operate without this configuration.

This example shows how to turn on the laser and place a DWDM port in In Service (IS) state:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:Router(config-dwdm)# admin-state in-service
RP/0/RP0/CPU0:Router(config-dwdm)# commit
```

Turning Off the Laser: Example

Note This configuration replaces the **laser off** and **shutdown (DWDM)** configuration commands.

This example shows how to turn off the laser, stop all traffic and place a DWDM port in Out of Service (OOS) state:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:Router(config-dwdm)# admin-state out-of-service
RP/0/RP0/CPU0:Router(config-dwdm)# commit
```

DWDM Controller Configuration: Examples

This example shows how to bring the DWDM controller down before using the configuration commands:

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/0/0/0
RP/0/RP0/CPU0:Router(config-dwdm)# maintenance out-of-service
RP/0/RP0/CPU0:Router(config-dwdm)# commit
RP/0/RP0/CPU0:Router(config-dwdm)# rx-los-threshold 0
RP/0/RP0/CPU0:Router(config-dwdm)# wavelength 1
RP/0/RP0/CPU0:Router(config-dwdm)# transmit-power 0
RP/0/RP0/CPU0:Router(config-dwdm)# maintenance in-service
RP/0/RP0/CPU0:Router(config-dwdm)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: y
RP/0/RP0/CPU0:Oct 15 12:35:54.299 : config[65732]: %MGBL-LIBTARCFG-6-COMMIT : Configuration
 committed by user 'lab'.   Use 'show configuration commit changes 1000000312' to view the
 changes.
```

```
RP/0/RP0/CPU0:Oct 15 12:35:54.403 : config[65732]: %MGBL-SYS-5-CONFIG_I : Configured from
console by lab
```

This example shows how to customize the alarm display and the thresholds for alerts and forward error correction (FEC):

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/0
RP/0/RP0/CPU0:Router(config-dwdm)# maintenance out-of-service
RP/0/RP0/CPU0:Router(config-dwdm)# commit
RP/0/RP0/CPU0:Router(config-dwdm)# g709 disable
RP/0/RP0/CPU0:Router(config-dwdm)# loopback internal
RP/0/RP0/CPU0:Router(config-dwdm)# g709 fec standard
RP/0/RP0/CPU0:Router(config-dwdm)# g709 odu bdi disable
RP/0/RP0/CPU0:Router(config-dwdm)# maintenance in-service
RP/0/RP0/CPU0:Router(config-dwdm)# commit
```

DWDM Performance Monitoring: Examples

This example shows how to configure performance monitoring for the optics parameters and how to display the configuration and current statistics:

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/2/0/0

RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opt max 2000000
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opt min 200
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold lbc max 3000000
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold lbc min 300
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opr max 4000000
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opr min 400
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report opt max-tca enable
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report opt min-tca enable
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report opr max-tca enable
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report opr min-tca enable
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report lbc max-tca enable
RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics report lbc min-tca enable
RP/0/RP0/CPU0:Router(config-dwdm)# exit
RP/0/RP0/CPU0:Router(config)# exit

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:y

LC/0/2/CPU0:Jul 12 04:10:47.252 : plim_4p_10ge_dwdm[194]: %L1-PMENGINE-4-TCA : Port DWDM
0/2/0/0 reports OPTICS TX-PWR-MIN(NE) PM TCA with current value 0, threshold 200 in current
 15-min interval window
LC/0/2/CPU0:Jul 12 04:10:47.255 : plim_4p_10ge_dwdm[194]: %L1-PMENGINE-4-TCA : Port DWDM
0/2/0/0 reports OPTICS RX-PWR-MIN(NE) PM TCA with current value 68, threshold 400 in current
 15-min interval window
RP/0/RP1/CPU0:Jul 12 04:09:05.443 : config[65678]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'lab'. Use 'show configuration commit changes 1000000001' to view the
changes.
RP/0/RP1/CPU0:Jul 12 04:09:05.604 : config[65678]: %MGBL-SYS-5-CONFIG_I : Configured from
console by lab

RP/0/RP0/CPU0:Router# show controllers dwdm 0/2/0/0 pm interval 15-min optics 0

Optics in the current interval [ 4:15:00 - 04:26:02 Wed Jul 12 2006]
             MIN     AVG     MAX  Threshold  TCA   Threshold  TCA
                                  (min)   (enable) (max)   (enable)
LBC[mA ] :  3605    4948    6453    300        YES     3000000   YES
```

```
OPT[uW]  :  2593    2593    2593    200        YES     2000000   YES
OPR[uW]  :  69      69      70      400        YES     4000000   YES
```

IPoDWDM Configuration: Examples

This section includes the following examples:

SRLG and Optical Layer DWDM Port Configuration: Examples

This example shows how to configure a Shared Risk Link Group (SRLG) and Optical Layer DWDM ports.

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:Router(config-dwdm)# network srlg value1 value2 value3
RP/0/RP0/CPU0:Router(config-dwdm)# network port id 1/0/1/1
RP/0/RP0/CPU0:Router(config-dwdm)# network connection id 1/1/1/1
```

Administrative State of DWDM Optical Ports Configuration: Examples

The following examples show how to configure the administrative state and optionally set the maintenance embargo flag:

For POS Interface

```
RP/0/0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/0/CPU0:Router(config-dwdm)# admin-state in-service
RP/0/0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router(config)# interface pos 1/0/1/1
RP/0/0/CPU0:Router(config-if)# maintenance disable
RP/0/0/CPU0:Router(config-if)# commit
```

For TenGigabit Interface

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:Router(config-dwdm)# admin-state in-service
RP/0/RP0/CPU0:Router(config-dwdm)# exit
RP/0/RP0/CPU0:Router(config)# interface tengige 1/0/1/1
RP/0/RP0/CPU0:Router(config-if)# maintenance disable
RP/0/RP0/CPU0:Router(config-if)# commit
```

Proactive FEC-FRR Triggering Configuration: Examples

This example shows how to configure automatic triggering of Forward Error Correction-Fast Re-Route (FEC-FRR):

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:Router(config-dwdm)#proactive
RP/0/RP0/CPU0:Router(config-dwdm)# logging signal LogFile1
RP/0/RP0/CPU0:Router(config-dwdm)# proactive trigger threshold 1 9
RP/0/RP0/CPU0:Router(config-dwdm)# proactive trigger window 10000
RP/0/RP0/CPU0:Router(config-dwdm)# proactive revert threshold 1 9
RP/0/RP0/CPU0:Router(config-dwdm)# proactive revert window 600000
```

CHAPTER 13

Configuring POS Interfaces

This module describes the configuration of Packet-over-SONET/SDH (POS) interfaces.

POS interfaces provide secure and reliable data transmission over SONET and Synchronous Digital Hierarchy (SDH) frames using Cisco High-Level Data Link Control (HDLC) protocol or Point-to-Point Protocol (PPP) encapsulation.

The commands for configuring Layer 1 POS interfaces are provided in the *Cisco IOS XR Interface and Hardware Component Command Reference*.

Feature History for Configuring POS Interfaces

Release	Modification
Release 2.0	This feature was introduced on the Cisco CRS-1 Router.
Release 3.2	Support for the following hardware was introduced on the Cisco CRS-1 Router: • 1-Port OC-192c/STM-64 POS/RPR XFP SPA • 4-Port OC-3c/STM-1 POS SPA • SIP-800
Release 3.3.0	Support for the 8-Port OC-12c/STM-4 POS SPA was introduced on the Cisco CRS-1 Router.
Release 3.4.0	Support was added on the Cisco CRS-1 Router for the following hardware: • Cisco 2-port OC-48c/STM16c POS SPA • Cisco 4-port OC-48c/STM16c POS SPA
Release 3.4.1	Support was added on the Cisco CRS-1 Router for the Cisco 1-Port OC-192c/STM-64 POS/RPR VSR Optics SPA.

Prerequisites for Configuring POS Interfaces

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 POS interfaces, be sure that the following conditions are met:

- You know the IP address of the interface you will assign to the new POS interface configuration.
- You have configured one of the following controller types:
  - A SONET controller, as described in the Configuring Clear Channel T3/E3 Controllers and Channelized T3 and T1/E1 Controllers, on page 287 module.
  - A DWDM controller, as described in the Configuring Dense Wavelength Division Multiplexing Controllers, on page 305 module.

Note POS DWDM controller configuration is supported on the Cisco 1-Port OC-768c/STM-256c DWDM PLIM only.

Information About Configuring POS Interfaces

To configure POS interfaces, you must understand the following concepts:

On the Cisco CRS-1 Router, a single POS interface carries data using PPP or Cisco HDLC encapsulation. Frame Relay is not supported on the Cisco CRS-1 Router.

The router identifies the POS interface address by the physical layer interface module (PLIM) card rack number, slot number, bay number, and port number that are associated with that interface. If a subinterface and permanent virtual circuits (PVCs) are configured under the POS interface, then the router includes the subinterface number in the POS interface path ID.

Default Settings for POS Interfaces

When a POS interface is brought up and no additional configuration commands are applied, the default interface settings shown in this table are present. These default settings can be changed by configuration.

Table 9: POS Modular Services Card and PLIM Default Interface Settings

Parameter	Configuration File Entry	Default Settings
Keepalive	**keepalive** {*interval* [*retry*] | **disable**} **no keepalive**	Interval of 10 seconds Retry of: • 5 (with PPP encapsulation) • 3 (with HDLC encapsulation)
Encapsulation	**encapsulation** [**hdlc** | **ppp**]	**hdlc**
Maximum transmission unit (MTU)	**mtu** *bytes*	4474 bytes
Cyclic redundancy check (CRC)	**crc** [**16** | **32**]	**32**

Note Default settings do not appear in the output of the **show running-config** command.

Cisco HDLC Encapsulation

Cisco High-Level Data Link Controller (HDLC) is the Cisco proprietary protocol for sending data over synchronous serial links using HDLC. Cisco HDLC also provides a simple control protocol called Serial Line Address Resolution Protocol (SLARP) to maintain serial link keepalives. HDLC is the default encapsulation type for serial interfaces under Cisco IOS XR software. Cisco HDLC is the default for data encapsulation at Layer 2 (data link) of the Open System Interconnection (OSI) stack for efficient packet delineation and error control.

Note Cisco HDLC is the default encapsulation type for the serial interfaces.

Cisco HDLC uses keepalives to monitor the link state, as described in the Keepalive Timer.

PPP Encapsulation

PPP is a standard protocol used to send data over synchronous serial links. PPP also provides a Link Control Protocol (LCP) for negotiating properties of the link. LCP uses echo requests and responses to monitor the continuing availability of the link.

Note When an interface is configured with PPP encapsulation, a link is declared down, and full LCP negotiation is re-initiated after three ECHOREQ packets are sent without receiving an ECHOREP response.

PPP provides the following Network Control Protocols (NCPs) for negotiating the properties of data protocols that run on the link:

- IP Control Protocol (IPCP)—negotiates IP properties

- Multiprotocol Label Switching control processor (MPLSCP)—negotiates MPLS properties
- Cisco Discovery Protocol control processor (CDPCP)—negotiates CDP properties
- IPv6CP—negotiates IP Version 6 (IPv6) properties
- Open Systems Interconnection control processor (OSICP)—negotiates OSI properties

PPP uses keepalives to monitor the link state, as described in the Keepalive Timer.

PPP supports the following authentication protocols, which require a remote device to prove its identity before allowing data traffic to flow over a connection:

- Challenge Handshake Authentication Protocol (CHAP)—CHAP authentication sends a challenge message to the remote device. The remote device encrypts the challenge value with a shared secret and returns the encrypted value and its name to the local router in a response message. The local router attempts to match the remote device's name with an associated secret stored in the local username or remote security server database; it uses the stored secret to encrypt the original challenge and verify that the encrypted values match.
- Microsoft Challenge Handshake Authentication Protocol (MS-CHAP)—MS-CHAP is the Microsoft version of CHAP. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a personal computer using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server.
- Password Authentication Protocol (PAP)—PAP authentication requires the remote device to send a name and a password, which are checked against a matching entry in the local username database or in the remote security server database.

Note For more information on enabling and configuring PPP authentication protocols, see the Configuring PPP Authentication, on page 405 module later in this manual.

Use the **ppp authentication** command in interface configuration mode to enable CHAP, MS-CHAP, and PAP on a POS interface.

Note Enabling or disabling PPP authentication does not effect the local router's willingness to authenticate itself to the remote device.

Keepalive Timer

Cisco keepalives are useful for monitoring the link state. Periodic keepalives are sent to and received from the peer at a frequency determined by the value of the keepalive timer. If an acceptable keepalive response is not received from the peer, the link makes the transition to the down state. As soon as an acceptable keepalive response is obtained from the peer or if keepalives are disabled, the link makes the transition to the up state.

If three keepalives are sent to the peer and no response is received from peer, then the link makes the transition to the down state. ECHOREQ packets are sent out only when LCP negotiation is complete (for example, when LCP is open).

Use the **keepalive** command in interface configuration mode to set the frequency at which LCP sends ECHOREQ packets to its peer. To restore the system to the default keepalive interval of 10 seconds, use the **keepalive** command with **no** argument. To disable keepalives, use the **keepalive disable** command. For both PPP and Cisco HDLC, a keepalive of 0 disables keepalives and is reported in the **show running-config** command output as **keepalive disable**.

To remove the **keepalive** command from the configuration entirely, use the **no keepalive** command. You must remove the **keepalive** command from an interface configuration before you can configure Frame Relay encapsulation on that interface. Frame Relay interfaces do not support keepalives.

Note During MDR, the keepalive interval must be 10 seconds or more.

When LCP is running on the peer and receives an ECHOREQ packet, it responds with an echo reply (ECHOREP) packet, regardless of whether keepalives are enabled on the peer.

Keepalives are independent between the two peers. One peer end can have keepalives enabled while the other end has them disabled. Even if keepalives are disabled locally, LCP still responds with ECHOREP packets to the ECHOREQ packets it receives. Similarly, LCP also works if the period of keepalives at each end is different.

Note Use the **debug chdlc slarp packet** command and other Cisco HDLC **debug** commands to display information about the Serial Line Address Resolution Protocol (SLARP) packets that are sent to the peer after the keepalive timer has been configured.

How to Configure a POS Interface

This section contains the following procedures:

Bringing Up a POS Interface

This task describes the commands you can use to bring up a POS interface.

Before you begin

You must have a POS line card or SPA installed in a router that is running Cisco IOS XR software.

Restrictions

The configuration on both ends of the POS connection must match for the interface to be active.

SUMMARY STEPS

1. **show interfaces**
2. **configure**
3. **interface pos** *interface-path-id*
4. **ipv4 address** *ipv4_address/prefix*
5. **no shutdown**

6. **end** or **commit**
7. **exit**
8. **exit**
9. Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.
10. **show ipv4 interface brief**
11. **show interfaces pos** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show interfaces** **Example:** `RP/0/RP0/CPU0:router# show interfaces`	(Optional) Displays configured interfaces. • Use this command to also confirm that the router recognizes the PLIM card.
Step 2	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 3	**interface pos** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0`	Specifies the POS interface name and notation *rack/slot/module/port,* and enters interface configuration mode.
Step 4	**ipv4 address** *ipv4_address/prefix* **Example:** `RP/0/RP0/CPU0:router (config)#ipv4 address 10.46.8.6/24`	Assigns an IP address and subnet mask to the interface. **Note** • Skip this step if you are configuring Frame Relay encapsulation on this interface. For Frame Relay, the IP address and subnet mask are configured under the subinterface.
Step 5	**no shutdown** **Example:** `RP/0/RP0/CPU0:router (config-if)# no shutdown`	Removes the shutdown configuration. **Note** • Removal of the shutdown configuration eliminates the forced administrative down on the interface, enabling it to move to an up or down state (assuming the parent SONET layer is not configured administratively down).
Step 6	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router (config-if)# end` or	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before`

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# commit`	`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.
Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode.

Step 9	Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.
	Example:
	`RP/0/RP0/CPU0:router# show interfaces`
	
	`RP/0/RP0/CPU0:router# configure`
	`RP/0/RP0/CPU0:router (config)# interface pos 0/3/0/0`
	`RP/0/RP0/CPU0:router (config-if)# no shutdown`
	`RP/0/RP0/CPU0:router (config-if)# commit`
	`RP/0/RP0/CPU0:router (config-if)# exit`
	`RP/0/RP0/CPU0:router (config)# exit`
		Note • The configuration on both ends of the POS connection must match.

Step 10	**show ipv4 interface brief** **Example:** `RP/0/RP0/CPU0:router # show ipv4 interface brief`	Verifies that the interface is active and properly configured. If you have brought up a POS interface properly, the “Status” field for that interface in the **show ipv4 interface brief** command output shows “Up.”
Step 11	**show interfaces pos** *interface-path-id* **Example:**	(Optional) Displays the interface configuration.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show interfaces pos 0/3/0/0`	

What to do next

To modify the default configuration of the POS interface you just brought up, see the “Configuring Optional POS Interface Parameters” section on page 594.

Configuring Optional POS Interface Parameters

This task describes the commands you can use to modify the default configuration on a POS interface.

Before you begin

Before you modify the default POS interface configuration, you must bring up the POS interface and remove the shutdown configuration, as described in the Bringing Up a POS Interface.

Restrictions

The configuration on both ends of the POS connection must match for the interface to be active.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id*
3. **encapsulation** [**hdlc** | **ppp**]
4. **pos crc** {**16** | **32**}
5. **mtu** *value*
6. **end** or **commit**
7. **exit**
8. **exit**
9. **show interfaces pos** [*interface-path-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0`	Specifies the POS interface name and notation *rack/slot/module/port,* and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**encapsulation** [**hdlc** | **ppp**] **Example:** `RP/0/RP0/CPU0:router(config-if)# encapsulation hdlc`	(Optional) Configures the interface encapsulation parameters and details such as HDLC or PPP. **Note** • The default encapsulation is **hdlc**.
Step 4	**pos crc** {**16** | **32**} **Example:** `RP/0/RP0/CPU0:router(config-if)# pos crc 32`	(Optional) Configures the CRC value for the interface. Enter the **16** keyword to specify 16-bit CRC mode, or enter the **32** keyword to specify 32-bit CRC mode. **Note** • The default CRC is **32**.
Step 5	**mtu** *value* **Example:** `RP/0/RP0/CPU0:router(config-if)# mtu 4474`	(Optional) Configures the MTU value. • The default value is 4474. • The POS MTU range is 64–9216.

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

Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:**	Exits global configuration mode and enters EXEC mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router (config)# exit`	
Step 9	**show interfaces pos** [*interface-path-id*] **Example:** `RP/0/RP0/CPU0:router# show interface pos 0/3/0/0`	(Optional) Displays general information for the specified POS interface.

What to do next

- To configure PPP authentication on POS interfaces where PPP encapsulation is enabled, see the Configuring PPP on the Cisco ASR 9000 Series Router module later in this manual.
- To modify the keepalive interval on POS interfaces that have Cisco HDLC or PPP encapsulation enabled, see the "Modifying the Keepalive Interval on POS Interfaces" section on page 601.

Modifying the Keepalive Interval on POS Interfaces

Perform this task to modify the keepalive interval on POS interfaces that have Cisco HDLC or PPP encapsulation enabled.

Note When you enable Cisco HDLC or PPP encapsulation on a POS interface, the default keepalive interval is 10 seconds. Use this procedure to modify that default keepalive interval.

Cisco HDLC is enabled by default on POS interfaces.

Before you begin

Before you can modify the keepalive timer configuration, you must ensure that Cisco HDLC or PPP encapsulation is enabled on the interface. Use the **encapsulation** command to enable Cisco HDLC or PPP encapsulation on the interface, as described in the Configuring Optional POS Interface Parameters.

Restrictions

During MDR, the keepalive interval must be 10 seconds or more.

SUMMARY STEPS

1. **configure**
2. **interface pos** *interface-path-id*
3. **keepalive** {*seconds* [*retry-count*] | **disable**} or **no keepalive**
4. **end** or **commit**
5. **show interfaces pos** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface pos** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0`	Specifies the POS interface name and notation *rack/slot/module/port* and enters interface configuration mode.

Step 3	**keepalive** {*seconds* [*retry-count*] | **disable**} or **no keepalive**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# keepalive 3`
	or
	`RP/0/RP0/CPU0:router(config-if)# no keepalive`
		Specifies the number of seconds between keepalive messages, and optionally the number of keepalive messages that can be sent to a peer without a response before transitioning the link to the down state.
		• Use the **keepalive disable** command, the **no keepalive**, or the **keepalive** command with an argument of 0 to disable the keepalive feature entirely.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**show interfaces pos** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces POS 0/3/0/0`	(Optional) Verifies the interface configuration.

Configuration Examples for POS Interfaces

This section provides the following configuration examples:

Bringing Up and Configuring a POS Interface with Cisco HDLC Encapsulation: Example

The following example shows how to bring up a basic POS interface with Cisco HDLC encapsulation:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

The following example shows how to configure the interval between keepalive messages to be 10 seconds:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# keepalive 10
RP/0/RP0/CPU0:router(config-if)# commit
```

Configuring a POS Interface with PPP Encapsulation: Example

The following example shows how to create and configure a POS interface with PPP encapsulation:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RP0/CPU0:router# show interfaces POS 0/3/0/0

POS0/3/0/0 is down, line protocol is down
  Hardware is Packet over SONET
  Internet address is 172.18.189.38/27
  MTU 4474 bytes, BW 2488320 Kbit
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation PPP, crc 32, controller loopback not set, keepalive set (
10 sec)
  LCP Closed
  Closed: IPCP
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
```

```
0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 0 multicast packets
         0 runts, 0 giants, 0 throttles, 0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

CHAPTER 14

Configuring SRP Interfaces

This module describes how to configure the Spatial Reuse Protocol (SRP) on supported Cisco Dynamic Packet Transport (DPT) interfaces.

SRP is a MAC-layer protocol developed by Cisco and is used in conjunction with Cisco DPT products. DPT products deliver scalable Internet service, reliable IP-aware optical transport, and simplified network operations. These solutions allow you to scale and distribute your IP services across a reliable optical packet ring infrastructure.

Note Throughout the remainder of this publication, the term SRP is used to describe features related to DPT products.

Feature History for Configuring SRP Interfaces on Cisco IOS XR Software

Release	Modification
Release 3.2.2	This feature was introduced on the Cisco CRS-1 Router and is supported only on the 4-port OC-192c/STM-64c POS/DPT PLIM.
Release 3.4.0	This command was first supported on the 16-port OC-48c/STM-16c POS/DPT PLIM.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.
Release 3.8.0	Support for this feature was added on the Cisco CRS-1 Router for the following shared port adapters (SPAs): • 1-port OC-192/STM-64 POS/RPR SPA XFP optics • 4-port OC-48/STM-16 POS/RPR SPA • 2-port OC-48/STM-16 POS/RPR SPA

Release 3.9.0	No modification.

Prerequisites for Configuring SRP Interfaces

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 SRP interfaces, be sure that the following conditions are met:

- You know the interface IP address you will assign to the new SRP interface configuration.
- The hardware that you are using supports SRP. SRP is currently supported on the following PLIMs and SPAs:
  - 4-port OC-192c/STM-64c POS/DPT PLIM
  - 16-port OC-48c/STM-16c POS/DPT PLIM
  - 1-port OC-192/STM-64 POS/RPR SPA XFP optics
  - 4-port OC-48/STM-16 POS/RPR SPA
  - 2-port OC-48/STM-16 POS/RPR SPA

Information About Configuring SRP Interfaces

Spatial bandwidth reuse is possible due to the packet destination-stripping property of SRP. Older technologies incorporate source stripping, where packets traverse the entire ring until they are removed by the source. Even if the source and destination nodes are next to each other on the ring, packets continue to traverse the entire ring until they return to the source to be removed. SRP provides more efficient use of available bandwidth by having the destination node remove the packet after it is read. This provides more bandwidth for other nodes on the SRP ring.

SRP rings consists of two counter rotating fibers, known as outer and inner rings, both concurrently used to carry data and control packets. SRP uses both explicit control packets and control information piggybacked inside data packets (control packets handle tasks such as keepalives, protection switching, and bandwidth control propagation). Control packets propagate in the opposite direction from the corresponding data packets, ensuring that the data takes the shortest path to its destination. The use of dual fiber-optic rings provides a high-level of packet survivability. In the event of a failed node or a fiber cut, data is transmitted over the alternate ring.

SRP rings are media independent and can operate over a variety of underlying technologies, including SONET/SDH, wavelength division multiplexing (WDM), and dark fiber. This ability to run SRP rings over any embedded fiber transport infrastructure provides a path to packet-optimized transport for high- bandwidth

IP networks. The Figure below shows an SRP ring created with a Cisco CRS-1 Router and a Cisco 12000 Series Router.

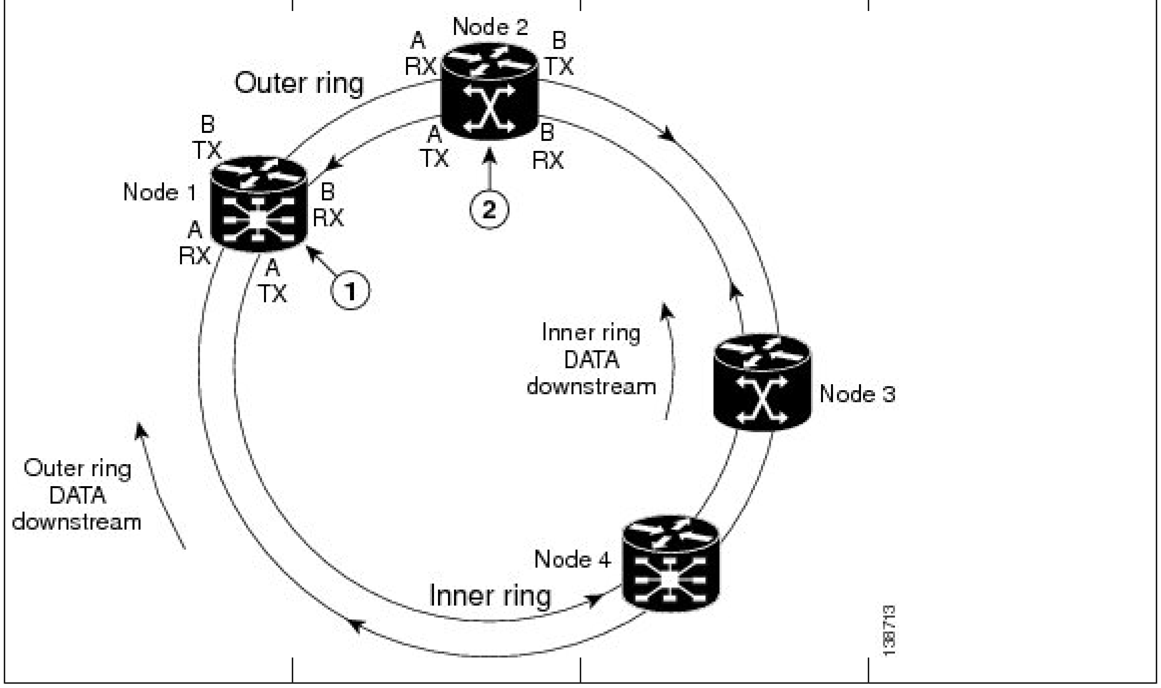
To distinguish between the two rings, one is referred to as the “inner” ring and the other as the “outer” ring. SRP operates by sending data packets in one direction (downstream) and sending the corresponding control packets in the opposite direction (upstream) on the other fiber. This allows SRP to use both fibers concurrently to maximize bandwidth for packet transport and to accelerate control signal propagation for adaptive bandwidth utilization, and for self-healing purposes.

As shown in the below figure, an SRP node uses SRP Side A to receive (RX) outer ring data and transmit (TX) inner ring data. The node uses SRP Side B to receive (RX) inner ring data and transmit (TX) outer ring data. Side A on one node connects to Side B on an adjacent SRP node.

The commands for configuring SRP interfaces are provided in the *Cisco IOS XR Interface and Hardware Component Command Reference*.

Figure 13: SRP Ring Example

1	Cisco CRS-1 Router	2	Cisco 12000 Series Router



SRR operates with standard IPS in the following ways:

- SRR relies on IPS to detect the local failure status of the node. It monitors the IPS status of both sides for an eventual “Signal Fail” or “Forced Switch” to occur.
- SRR doesn't consider remote failures transmitted through IPS short or long path messages.
- Whenever SRR enables the single ring mode, it will prevent IPS from wrapping the node by enabling IPS lockout on the node.
- When the single ring mode is not enabled by SRR, the SRP ring will follow standard IPS behavior and wrap accordingly.

How to Configure an SRP Interface

This section contains the following procedures:

Enabling SRP on a PLIM Port

To enable the use of SRP on a PLIM port, you must perform this task. By default, POS/DPT PLIMs support only POS.

Restrictions

On the 4-port OC-192c/STM-64c POS/DPT PLIM, each port pair (0 and 1 or 2 and 3) must be configured the same. If you configure port 0 to be SRP and do not configure port 1 to be SRP, the configuration does not work.

On the 16-port OC-48c/STM-16c POS/DPT PLIM, each group of four ports must be configured the same. If you want to use ports 0 and 1 as a single SRP interface, you must configure all four ports: 0, 1, 2, and 3 to be SRP. Similarly, ports 4-7, 8-11, and 12-15 must be configured the same, as either SRP or POS, for the configuration to work.

SUMMARY STEPS

1. **configure**
2. **hw-module port** *port-number-1* **srp location** *instance*
3. **hw-module port** *port-number-2* **srp location** *instance*
4. **hw-module port** *port-number-3* **srp location** *instance*
5. **hw-module port** *port-number-4* **srp location** *instance*
6. **end** or **commit**
7. **hw-module location node-id reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module port** *port-number-1* **srp location** *instance* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0`	Enables SRP functionality on the first port. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The first, lower-numbered port must be even, for example 0 or 2. • On the 16-port OC-48c/STM-16c POS/DPT PLIM, groups of four consecutive ports must be configured the same: ports 0-3, 4-7, 8-11 and 12-15.

	Command or Action	Purpose
Step 3	**hw-module port** *port-number-2* **srp location** *instance* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 1 srp location 0/5/cpu0`	Enables SRP functionality on the second port. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd. • On the 16-port OC-48c/STM-16c POS/DPT PLIM, groups of four consecutive ports must be configured the same: ports 0-3, 4-7, 8-11, and 12-15.
Step 4	**hw-module port** *port-number-3* **srp location** *instance* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 2 srp location 0/5/cpu0`	Enables SRP functionality on the third port, for 16-port OC-48c/STM-16c POS/DPT PLIMs. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The first, lower-numbered port must be even; for example, 0 or 2. • On the 16-port OC-48c/STM-16c POS/DPT PLIM, groups of four consecutive ports must be configured the same: ports 0-3, 4-7, 8-11, and 12-15.
Step 5	**hw-module port** *port-number-4* **srp location** *instance* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 3 srp location 0/5/cpu0`	Enables SRP functionality on the fourth port, for 16-port OC-48c/STM-16c POS/DPT PLIMs. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd. • On the 16-port OC-48c/STM-16c POS/DPT PLIM, groups of four consecutive ports must be configured the same: ports 0-3, 4-7, 8-11, and 12-15.

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

	Command or Action	Purpose
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**hw-module location node-id reload** **Example:** `RP/0/RP0/CPU0:router# hw-module location 0/5/cpu0 reload`	Reloads the PLIM and makes the **hw-module port** command become effective.

Note You must reload the PLIM to enable this configuration change and create the SRP interface.

After you complete this procedure, the following SRP interfaces are available to be configured on the PLIM in slot 5:

- 0/5/0/0, which comprises ports 0/5/0/0 and 0/5/0/1
- 0/5/0/2, which comprises ports 0/5/0/2 and 0/5/0/3

Enabling SRP on an OC-48/STM-16 SPA Port

To enable the use of SRP on an OC-48/STM-16 SPA port, you must perform this task.

Restrictions

All ports on a 4-port OC-48/STM-16 POS/RPR SPA or 2-port OC-48/STM-16 POS/RPR SPA must function in either POS mode or SRP mode. Therefore, if you plan to use SRP, you must enable it on all SPA ports.

SUMMARY STEPS

1. **configure**
2. **hw-module port** *port-number-1* **srp location** *instance* **spa-bay** *number*
3. **hw-module port** *port-number-2* **srp location** *instance* **spa-bay** *number*
4. **hw-module port** *port-number-3* **srp location** *instance* **spa-bay** *number*
5. **hw-module port** *port-number-4* **srp location** *instance* **spa-bay** *number*
6. **end** or **commit**
7. **hw-module subslot** *subslot-id* **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module port** *port-number-1* **srp location** *instance* **spa-bay** *number* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0 spa-bay 2`	Enables SRP functionality on the first port. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The first, lower-numbered port must be even, for example 0 or 2.
Step 3	**hw-module port** *port-number-2* **srp location** *instance* **spa-bay** *number* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 1 srp location 0/5/cpu0 spa-bay 2`	Enables SRP functionality on the second port. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd.
Step 4	**hw-module port** *port-number-3* **srp location** *instance* **spa-bay** *number* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 2 srp location 0/5/cpu0 spa-bay 2`	Enables SRP functionality on the third port, for 4-port OC-48/STM-16 POS/RPR SPAs. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The first, lower-numbered port must be even; for example, 0 or 2.
Step 5	**hw-module port** *port-number-4* **srp location** *instance* **spa-bay** *number* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 3 srp location 0/5/cpu0 spa-bay 2`	Enables SRP functionality on the fourth port, for 4-port OC-48/STM-16 POS/RPR SPAs. **Note** • An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd.

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

	Command or Action	Purpose
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**hw-module subslot** *subslot-id* **reload** **Example:** `RP/0/RP0/CPU0:router# hw-module subslot 0/5/cpu0 reload`	Reloads the SPA and makes the **hw-module port** command become effective. **Note** • You must reload the SPA to enable this configuration change and create the SRP interface. After you complete this procedure, the following SRP interfaces are available to be configured on the SPA in slot 5: • 0/5/0/0, which comprises ports 0/5/0/0 and 0/5/0/1 • 0/5/0/2, which comprises ports 0/5/0/2 and 0/5/0/3

Note To disable the use of SRP on the OC-48/STM-16 ports associated with a particular SPA, perform the same steps in this section except in Step 2 through Step 5, use the **no hw-module port** *port-number* **srp location** *instance* **spa-bay** *number* command.

If you have incorrectly configured an OC-48/STM-16 port and you want to remove that configuration, you can specify the **no hw-module port** *port-number* **srp location** *instance* **spa-bay** *number* command for that particular port only. However, we recommend performing this action only under the circumstances described here, because partial removal of the SRP configuration can leave the router in an indeterminate state.

Enabling SRP on an OC-192/STM-64 SPA Port

To enable the use of SRP on an OC-192/STM-64 SPA port, you must perform this task.

Before proceeding with this task, be aware that an SRP interface is comprised of two OC-192/STM-64 POS/RPR SPAs, each of which is installed in a separate bay. Also, each SPA runs a separate process. Therefore, there are two SPAs and processes for a single SRP interface.

SUMMARY STEPS

1. **configure**
2. **hw-module port** *port-number-1* **srp location** *instance* **spa-bay** *number*

3. **hw-module port** *port-number-2* **srp location** *instance* **spa-bay** *number*
4. **commit**
5. **hw-module subslot** *subslot-id* **shutdown**
6. **hw-module subslot** *subslot-id* **shutdown**
7. **commit**
8. **no hw-module subslot** *subslot-id* **shutdown**
9. **no hw-module subslot** *subslot-id* **shutdown**
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module port** *port-number-1* **srp location** *instance* **spa-bay** *number* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0 spa-bay 0`	Enables SRP functionality on the SPA in the first bay.
Step 3	**hw-module port** *port-number-2* **srp location** *instance* **spa-bay** *number* **Example:** `RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0 spa-bay 1`	Enables SRP functionality on the SPA in the second bay.
Step 4	**commit** **Example:** `RP/0/RP0/CPU0:router(config)# commit`	Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**hw-module subslot** *subslot-id* **shutdown** **Example:** `RP/0/RP0/CPU0:router(config)# hw-module subslot 0/5/0 shutdown`	**Note** • You must reload each SPA to enable this configuration change and create the SRP interface. To do so, we recommend shutting down each SPA and then bringing it back up. We do not recommend using the **hw-module subslot** *subslot-id* **reload** command to reload each SPA, because doing so can cause synchronization problems with the two SPAs and processes that comprise an SRP interface. Shuts down the SPA in bay 0 of SRP location 0/5/cpu0.

	Command or Action	Purpose
Step 6	**hw-module subslot** *subslot-id* **shutdown** **Example:** `RP/0/RP0/CPU0:router(config)# hw-module subslot 0/5/1 shutdown`	**Note** • You must reload each SPA to enable this configuration change and create the SRP interface. To do so, we recommend shutting down each SPA and then bringing it back up. We do not recommend using the **hw-module subslot** *subslot-id* **reload** command to reload each SPA, because doing so can cause synchronization problems with the two SPAs and processes that comprise an SRP interface. Shuts down the SPA in bay 1 of SRP location 0/5/cpu0.
Step 7	**commit** **Example:** `RP/0/RP0/CPU0:router(config)# commit`	Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 8	**no hw-module subslot** *subslot-id* **shutdown** **Example:** `RP/0/RP0/CPU0:router(config)# no hw-module subslot 0/5/0 shutdown`	Returns the SPA in bay 0 of SRP location 0/5/cpu0 to the up state.
Step 9	**no hw-module subslot** *subslot-id* **shutdown** **Example:** `RP/0/RP0/CPU0:router(config)# no hw-module subslot 0/5/1 shutdown`	Returns the SPA in bay 1 of SRP location 0/5/cpu0 to the up state.
Step 10	**commit** **Example:** `RP/0/RP0/CPU0:router(config)# commit`	Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. After you complete this procedure, SRP interface 0/5, which is comprised of ports 0/5/0 and 0/5/1 are available to be configured.

Note To disable the use of SRP on the OC-192/STM-64 ports associated with a particular SRP interface, perform the same steps in this section except in Step 2 and Step 3, use the **no hw-module port** *port-number* **srp location** *instance* **spa-bay** *number* command.

If you have incorrectly configured an OC-192/STM-64 port and you want to remove that configuration, you can specify the **no hw-module port** *port-number* **srp location** *instance* **spa-bay** *number* command for that particular port only. However, we recommend performing this action only under the circumstances described above, because partial removal of the SRP configuration can leave the router in an indeterminate state.

Creating a Basic SRP Configuration

This task explains how to create a basic SRP configuration. There are many other possible parameters that can be set and only the most basic are illustrated in this task.

Note You must enable SRP on the interface before you can perform this task. See Enabling SRP on a PLIM Port.

SUMMARY STEPS

1. **show interfaces**
2. **configure**
3. **controller sonet** *interface-path-id* **clock source internal**
4. **interface srp** *interface-path-id*
5. **ipv4 address** *ip-address*
6. **srp topology-timer** *value*
7. **no shutdown**
8. **end** or **commit**
9. **show interfaces srp** *interface-path-id*
10. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show interfaces** **Example:** `RP/0/RP0/CPU0:router# show interfaces`	(Optional) Displays configured interfaces. • Also use this command to confirm that the router recognizes the PLIM card.
Step 2	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 3	**controller sonet** *interface-path-id* **clock source internal**
	Example:
	`RP/0/RP0/CPU0:router(config)# controller sonet 0/1/0/0 clock source internal`
	`RP/0/RP0/CPU0:router(config)# controller sonet 0/1/0/1 clock source internal`
		Configures the SONET port transmit clock source for each port comprising the SRP interface. The controller instance is in the notation *rack/slot/module/port*, and the **internal** keyword specifies internal clock.
		Note • Internal clocking is required for SRP interfaces.
		• Refer to Configuring Clear Channel SONET Controllers, on page 271 for more detailed information on the SONET controller configuration.

	Command or Action	Purpose
Step 4	**interface srp** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface srp 0/1/0/0`	Specifies the SRP interface name and notation *rack/slot/module/port,* and enters interface configuration mode.
Step 5	**ipv4 address** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.1 255.255.255.224`	Assigns an IP address and subnet mask to the interface.
Step 6	**srp topology-timer** *value* **Example:** `RP/0/RP0/CPU0:router(config-if)# srp topology-timer 1`	(Optional) Specifies how frequently topology discovery messages are sent around the ring to identify the current nodes on the SRP ring.
Step 7	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration. • The removal of the shutdown configuration removes the forced administrative down state on the interface, enabling it to move to an up or down state.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 9	**show interfaces srp** *interface-path-id* **Example:**	(Optional) Displays the SRP interface configuration.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show interfaces srp 0/1/0/0`	
Step 10	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	(Optional) Displays the configuration information currently running on the router.

Configuring Intelligent Protection Switching (IPS)

Perform this task to configure IPS on an SRP interface. This is an optional task.

Intelligent Protection Switching (IPS) provides IP self-healing and restoration, and performance monitoring after a link or node failure. There are two SRP IPS modes:

- Automatic SRP IPS mode takes effect when the SRP ring detects an event, a fiber cut, or a node failure, and remains in effect until the trigger condition is cleared. Once the trigger is cleared, the SRP IPS mode remains in effect until the wait-to-restore (WTR) value expires.
- User-configured SRP IPS mode takes effect as soon as you enter the command and remains in effect until it is removed by a user command or overridden by an SRP IPS command with higher priority. You can use the **no srp ips request forced-switch** global configuration command or the **srp remove manual-switch** EXEC command to negate a user-configured command.

A a user-configured, forced-switch adds a high-priority protection switch wrap on each end of a specified span by entering the user-configured **srp ips request forced-switch** command. For example, you can enter an **srp ips request forced-switch** command to force data traffic to one side of the ring before a DPT PLIM is removed from a router slot, or in response to an event.

This table describes the IPS requests in the order of priority, from higher to lower.

Table 10: Explanation of SRP IPS User Requests

SRP IPS Request	Description
Forced-switch	Adds a high-priority protection switch wrap on each end of a specified span by entering the user-configured srp ips request forced-switch command.
Manual-switch	Adds a low-priority protection switch wrap on each end of a specified span by entering the user-configured srp request manual-switch command.

Note Before removing the DPT PLIM, you can use the srp ips request forced-switch command on both sides of the interface that is to be removed.

If an automatic or user-configured protection switch is requested for a given span, the node that receives the protection request issues a protection request to the node on the other end of the span using both the short path over the failed span, because the failure may be unidirectional, and the long path around the ring.

As the protection requests travel around the ring, the protection hierarchy is applied. For example, if a high-priority Signal Fail (SF) request enters the ring, it overrides a preexisting lower-priority request. If an event or a user-configured command enters a low-priority request, it is not allowed if a high-priority request is present on the ring.

Note An exception is that multiple signal-fail and forced-switch requests can coexist on the SRP ring and will bisect the ring if they occur on separate fiber links.

All protection switches are performed bidirectionally and enter wraps at both ends of a span for transmit and receive directions, even if a failure is only unidirectional.

SUMMARY STEPS

1. **configure**
2. **interface srp** *interface-path-id*
3. **srp ips wtr-timer** *seconds*
4. **srp ips timer** *seconds*
5. **srp ips request forced-switch {a | b}**
6. **end** or **commit**
7. **srp {request | remove} manual-switch {a | b} interface srp** *interface-path-id*
8. **show srp ips interface srp** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface srp** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface srp 0/1/0/0`	Specifies the SRP interface name in the notation *rack/slot/module/port* and enters interface configuration mode.
Step 3	**srp ips wtr-timer** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-if)# srp ips wtr-timer 60`	(Optional) Configures the amount of time in seconds that a wrap remains in place after the cause of the wrap is removed.
Step 4	**srp ips timer** *seconds* **Example:**	(Optional) Specifies the frequency of the transmission of IPS requests. The default is 1 second.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# srp ips timer 60 a`	**Note** • We recommend that the IPS timer value be the same for all nodes on a ring. Therefore, if the IPS timer value is changed on one node, you should change it for all nodes on the ring using **srp ips timer** command.
Step 5	**srp ips request forced-switch {a | b}** **Example:** `RP/0/RP0/CPU0:router(config-if)# srp ips request forced-switch a`	(Optional) Adds a high-priority protection switch wrap on each end of a specified span. **Note** • Use this command only as necessary, as it disables the node.

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

Step 7	**srp {request | remove} manual-switch {a | b} interface srp** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# srp remove manual-switch a interface srp 0/1/0/0`	(Optional) Adds or removes a low-priority protection switch wrap on each end of a specified span. **Note** • Use this command only as necessary.
Step 8	**show srp ips interface srp** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show srp ips interface srp 0/1/0/0`	(Optional) Displays the IPS configuration on the SRP interface.

Configuring Modular Quality of Service CLI (MQC) with SRP

Perform this task to configure quality-of-service (QoS) classifications with SRP using the Modular QoS command-line interface (MQC) feature. This is an optional task.

Note For more information regarding MQC, refer to Configuring Modular Quality of Service Packet Classification on Cisco IOS XR Software and Cisco IOS XR Modular Quality of Service Command Reference.

SUMMARY STEPS

1. **configure**
2. **class-map match-any** *class-map-name*
3. **match mpls experimental topmost** *exp-value*
4. **exit**
5. **class-map match-any** *class-map-name*
6. **match precedence** *precedence-value*
7. **exit**
8. **policy-map** *policy-name*
9. **class** *class-name*
10. **police cir** *kbps*
11. **set cos** *cos-value*
12. **priority**
13. **exit**
14. **class** *class-name*
15. **priority**
16. **set cos** *cos-value*
17. **exit**
18. **exit**
19. **interface srp** *interface-path-id*
20. **service-policy output** *policy-map*
21. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**class-map match-any** *class-map-name* **Example:** `RP/0/RP0/CPU0:router(config)# class-map match-any voice`	Enters class map configuration mode. • Creates a class map to be used for matching packets to the class whose name you specify.

	Command or Action	Purpose
		• If you specify **match-any**, one of the match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class.
Step 3	**match mpls experimental topmost** *exp-value* **Example:** `RP/0/RP0/CPU0:router(config-cmap)# match mpls experimental topmost 4`	Configures a class map so that the three-bit experimental (EXP) field in the topmost Multiprotocol Label Switching (MPLS) labels are examined for EXP field values. • The EXP value argument is specified as the exact value from 0 to 7.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-cmap)# exit`	Exits the current submode.
Step 5	**class-map match-any** *class-map-name* **Example:** `RP/0/RP0/CPU0:router(config)# class-map match-any ctrl`	Enters class map configuration mode. • Creates a class map to be used for matching packets to the class whose name you specify. • If you specific **match-any**, one of the match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class.
Step 6	**match precedence** *precedence-value* **Example:** `RP/0/RP0/CPU0:router(config-cmap)# match precedence internet`	(Optional) Identifies IP precedence values as match criteria. • The range is from 0 to 63. • Reserved keywords can be specified instead of numeric values.
Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router(config-cmap)# exit`	Exits the current submode.
Step 8	**policy-map** *policy-name* **Example:** `RP/0/RP0/CPU0:router(config)# policy-map srp-policy`	Enters policy map configuration mode. • Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 9	**class** *class-name* **Example:** `RP/0/RP0/CPU0:router(config-pmap)# class voice`	Specifies the name of the class whose policy you want to create or change.
Step 10	**police cir** *kbps* **Example:**	Configures traffic policing.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-pmap-c)# police cir 2000000`	**Note** • 2000000 represents 10 percent of the interface line rate.
Step 11	**set cos** *cos-value* **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# set cos 4`	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
Step 12	**priority** **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# priority`	Gives priority to a class of traffic belonging to a policy map. **Note** • The **priority** command should only be used if the **set cos** command is also used and specifies a cos value greater then or equal to 2.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# exit`	Exits the current submode.
Step 14	**class** *class-name* **Example:** `RP/0/RP0/CPU0:router(config-pmap)# class ctrl`	Specifies the name of the class whose policy you want to create or change.
Step 15	**priority** **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# priority`	Gives priority to a class of traffic belonging to a policy map. **Note** • The **priority** command should only be used if the **set cos** command is also used and specifies a cos value greater than or equal to 2.
Step 16	**set cos** *cos-value* **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# set cos 6`	Sets the Layer 2 CoS value of an outgoing packet.
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# exit`	Exits the current submode.
Step 18	**exit** **Example:**	Exits the current submode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-pmap)# exit`	
Step 19	**interface srp** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface srp 0/1/0/0`	Specifies the SRP interface in the notation *rack/slot/module/port* and enters interface configuration mode.
Step 20	**service-policy output** *policy-map* **Example:** `RP/0/RP0/CPU0:router(config-if)# service-policy output srp-policy`	Attaches a policy map to an input or output interface to be used as the service policy for that interface. • The traffic policy evaluates all traffic leaving that interface.

Step 21	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Adding a Node to the Ring

This task describes how to add a node to an existing SRP ring, using Cisco IOS XR commands that insert forced-switch wraps away from the area on the fiber where the node is being added, to ensure a minimal loss of data traffic.

For the purpose of this example, a fifth node is added to a four-node ring. Node 5 is added between Node 1 and Node 4. The below figures show the physical configuration using a single DPT PLIM and logical configuration.

SUMMARY STEPS

1. **configure**
2. **interface srp** *interface-path-id*
3. **srp ips request forced-switch {a | b}**
4. **end** or **commit**
5. Disconnect the fiber-optic cables connecting Node 1 to Node 4.
6. Connect the cables to add the new node while observing the receive (RX) and transmit (TX) cabling relationship.
7. **interface srp** *interface-path-id*
8. **no srp ips request forced-switch** {**a** | **b**}
9. **end** or **commit**
10. **show srp ips**
11. **show srp errors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router1# configure`	Enters global configuration mode.
Step 2	**interface srp** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router1(config)# interface srp 0/1/0/0`	Specifies the SRP interface in the notation *rack/slot/module/port* for Node 1 and enters interface configuration mode.
Step 3	**srp ips request forced-switch {a | b}** **Example:** `RP/0/RP0/CPU0:router1(config-if)# srp ips request forced-switch a`	(Optional) Adds a high-priority protection switch wrap on each end of the specified span. This stops traffic flowing from Node 1 on the fiber that will be disconnected and creates a wrap next to Node 1 on Side A. **Note** • If you choose not to use the **srp ips request forced-switch** command, as soon as you perform Step 5, a signal failure is detected by Node 1 and Node 4, and they automatically insert two signal-fail wraps away from the failure between the nodes. We recommend that you use the **srp ips request forced-switch** command to minimize data loss.
Step 4	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router1(config-if)# end`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

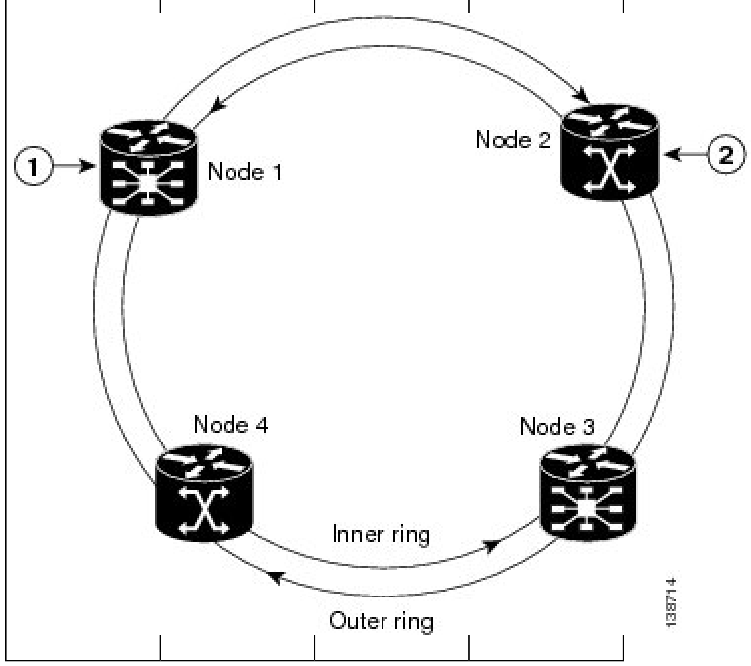
	Command or Action	Purpose
	or `RP/0/RP0/CPU0:router1(config-if)# commit`	`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC 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.
Step 5	Disconnect the fiber-optic cables connecting Node 1 to Node 4.	
Step 6	Connect the cables to add the new node while observing the receive (RX) and transmit (TX) cabling relationship.	See the below figure.
Step 7	**interface srp** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router1(config)# interface srp 0/1/0/0`	Specifies the SRP interface in the notation *rack/slot/module/port* for Node 1 and enters interface configuration mode.
Step 8	**no srp ips request forced-switch** {**a** | **b**} **Example:** `RP/0/RP0/CPU0:router1(config-if)# no srp ips request forced-switch a`	Removes the high-priority protection switch wrap on each end of the specified span. This allows traffic to flow again from Node 1. (See the below figure.) **Note** • If you performed Step 3, then you must use the **no srp ips request forced-switch** command to remove the wraps. If you did not perform Step 3, the wraps are removed automatically when the WTR timer has expired.

Step 9	**end** or **commit**
	
	Example:
	
	`RP/0/RP0/CPU0:router1(config-if)# end`
	
	or
	
	`RP/0/RP0/CPU0:router1(config-if)# 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]:`

	Command or Action	Purpose
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 10	**show srp ips** **Example:** `RP/0/RP0/CPU0:router5# show srp ips`	Confirms that the wraps have disappeared and the new node is part of the ring. (See the figure below.)
Step 11	**show srp errors** **Example:** `RP/0/RP0/CPU0:router5# show srp errors`	Confirms that there are no problems with the new ring configuration. If there are failures, note the status on the LEDs to determine what the problem might be.

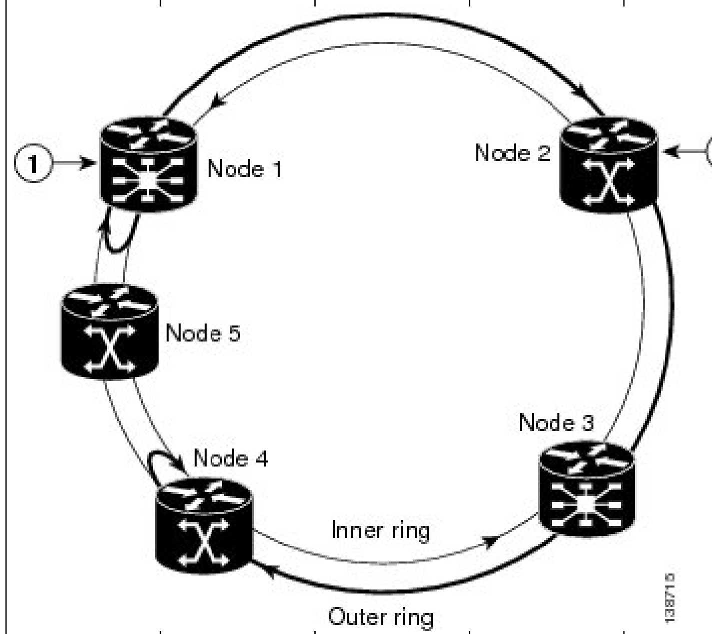
Figure 14: Four Routers on the SRP Ring (Each Router Using Two Physical Ports)

1	Cisco CRS-1 Router	2	Cisco 12000 Series Router



	Command or Action	Purpose
		Figure 15: SRP Ring Topology with a Fifth Node Added to a Wrapped Ring

1	Cisco CRS-1 Router	2	Cisco 12000 Series Router



Configuring Single Ring Recovery on SRP Interfaces

Perform this task to configure the Single Ring Recover (SRR) protocol. SRR allows SRP rings to operate over a single fiber in the event of multiple failures on one of the two counter-rotating SRP rings, thereby allowing the system to operate with full connectivity.

The following configuration is optional. SRR is enabled by default.

Note Cisco Systems recommends the use of the default bandwidth and timer values for optimal running of the SRR protocol.

Before you begin

SRR requires a fully SRR compatible ring to operate. In order for SRR to converge:

- all nodes on the SRP ring must support SRR
- all nodes must support the same SRR version

  If one or more of the nodes does not support SRR or has a different SRR version, SRR will have no effect and the ring will operate like a standard SRP ring without SRR.

SUMMARY STEPS

1. **configure**
2. **interface srp** *instance*
3. **srp srr bandwidth** *value*
4. **srp srr timer** *seconds*
5. **srp srr wtr-timer** *seconds*
6. **end** or **commit**
7. **show srp srr interface srp** *instance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface srp** *instance* **Example:** `RP/0/RP0/CPU0:router(config)# interface srp 0/4/0/1`	Enters the SRP interface configuration mode.
Step 3	**srp srr bandwidth** *value* **Example:** `RP/0/RP0/CPU0:router(config-if)# srp srr bandwidth 400`	Specifies the bandwidth for the interface in megabits per second (Mbps) when the interface operates on a single ring. The value is between 1 to the maximum possible bandwidth of the node.
Step 4	**srp srr timer** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-if)# srp srr timer 10`	Specifies the periodic timer, in seconds, for SRR messages in the Idle state. The default is 10 seconds.
Step 5	**srp srr wtr-timer** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-if)# srp srr wtr-timer 10`	Specifies the amount of time in seconds that the ring operates on one ringe after the cause of the failure is removed. The default is 60 seconds.

Step 6	**end** or **commit**
	Example:
	`RP/0/0/CPU0:router(config-if)# end`
	or
	`RP/0/0/CPU0:router(config-if)# 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]:`

	Command or Action	Purpose
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7	**show srp srr interface srp** *instance* **Example:** `RP/0/RP0/CPU0:router# show srp srr interface srp 0/1/0/0`	(Optional) Displays the SRR configuration on the SRP interface.

Configuration Examples for SRP Interfaces

This section provides the following configuration examples:

Enabling SRP: Examples

This example shows how to enable SRP on a PLIM port.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/3/CPU0
RP/0/RP0/CPU0:router(config)# hw-module port 1 srp location 0/3/CPU0
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# end
RP/0/RP0/CPU0:router# hw-module node 0/3/CPU0 reload

<Wait for LC to be reloaded, and interface created.  Or can use 'preconfigure'...>
```

This example shows how to enable SRP on four OC-48/STM-16 SPA ports.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/3/CPU0 spa-bay 2
RP/0/RP0/CPU0:router(config)# hw-module port 1 srp location 0/3/CPU0 spa-bay 2
RP/0/RP0/CPU0:router(config)# hw-module port 2 srp location 0/3/CPU0 spa-bay 2
RP/0/RP0/CPU0:router(config)# hw-module port 3 srp location 0/3/CPU0 spa-bay 2
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# end
RP/0/RP0/CPU0:router# hw-module subslot 0/3/CPU0 reload
```

```
<Wait for LC to be reloaded, and interface created.  Or can use 'preconfigure'...>
```

This example shows how to enable SRP on two OC-192/STM-64 SPA ports, which comprise an SRP interface.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/3/CPU0 spa-bay 0
RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/3/CPU0 spa-bay 1
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# hw-module subslot 0/3/0 shutdown
RP/0/RP0/CPU0:router(config)# hw-module subslot 0/3/1 shutdown
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# no hw-module subslot 0/3/0 shutdown
RP/0/RP0/CPU0:router(config)# no hw-module subslot 0/3/1 shutdown
RP/0/RP0/CPU0:router(config)# commit
```

Configuring Basic SRP: Example

This example shows how to configure the basic interface configuration for SRP.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# controller SONET 0/3/0/0 clock source internal
RP/0/RP0/CPU0:router(config)# controller SONET 0/3/0/1 clock source internal
RP/0/RP0/CPU0:router(config)# interface SRP 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config)# end
```

Configuring Modular QoS with SRP: Example

This example shows how to configure two quality-of-service (QoS) classes. One is for voice traffic and is identified by an MPLS experimental bit value of 4; the second is control traffic that is identified by an IP precedence value of 6. Both classes of traffic are sent to the SRP high priority queue and are marked with high SRP priority (4 and 6).

```
Last configuration change at 04:56:06 UTC Tue Sep 06 2005 by lab
!
hostname router
class-map match-any ctrl
 match precedence internet
!
class-map match-any voice
 match mpls experimental topmost 4
!
policy-map srp-policy
 class voice
  police cir 2000000
  set cos 4
  priority
 !
 class ctrl
  priority
  set cos 6
 !
!
interface SRP0/7/0/0
 description "Connected to 3-nodes ring"
```

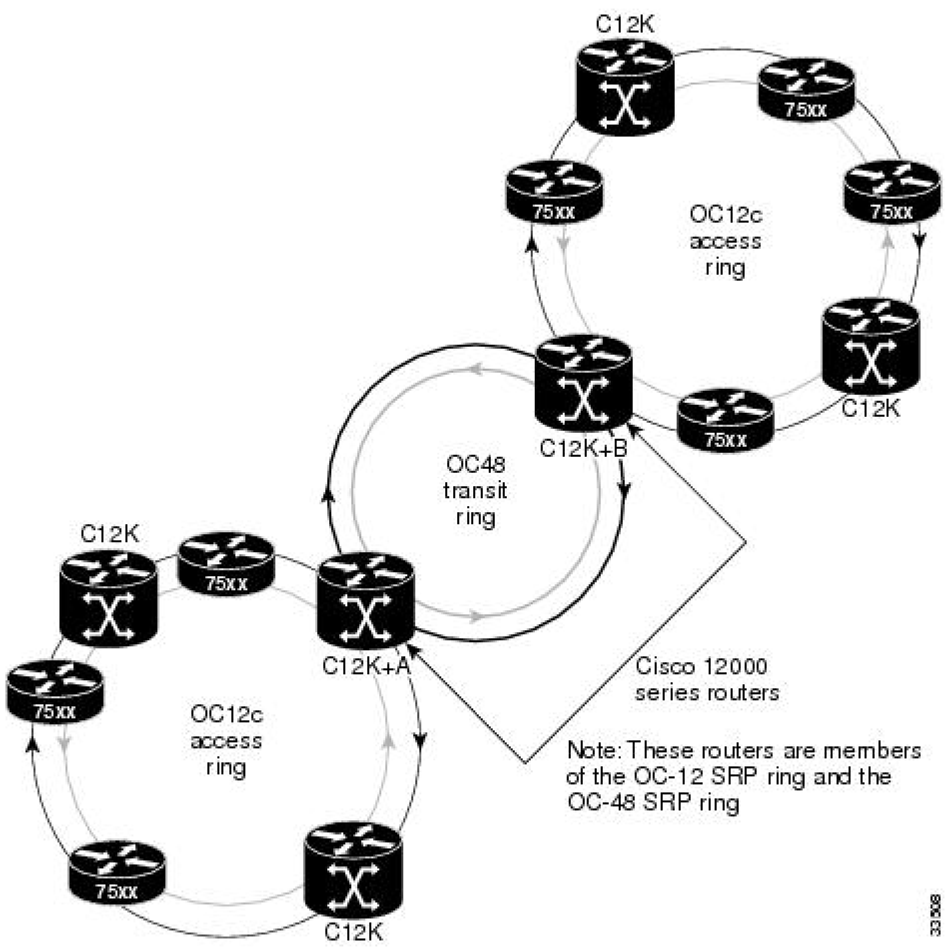
```
service-policy output srp-policy
ipv4 address 30.30.30.2 255.255.255.0
```

Creating a Metropolitan-Area Network with SRP Rings

In this example, an OC-48c/STM-16c SRP ring is used to interconnect two OC-12c/STM-4c access rings to form a larger hierarchical SRP ring topology by directly connecting two Cisco 12000 Series Internet Routers together using direct fiber connections without the use of SONET Add/Drop Multiplexers (ADMs).

Note Each SRP ring must be on a different subnet.

Figure 16: Two OC12 SRP Rings Connected to an OC48 SRP Ring



This configuration example shows the Cisco IOS commands used to configure SRP rings on the GSR+ A and GSR+ B routers in the above figure.

GSR+ A Configuration

```
GSR+A:
```

```
Building configuration...
Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname GSR+A
!
!
hw-module slot 4 srp
!
ip subnet-zero
no ip domain-lookup
ip multicast-routing distributed
 ip pim rp-address 10.8.1.20 1
!
interface Loopback0
ip address 10.0.0.1 255.255.255.252
no ip directed-broadcast
!
interface SRP1/0
ip address 10.10.10.1 255.255.255.192
no ip redirects
no ip directed-broadcast
ip pim sparse-mode
ip mroute-cache distributed
load-interval 30
!
interface Ethernet0
ip address 10.100.1.2 255.255.255.0
no ip directed-broadcast
no ip route-cache cef
!
interface SRP4/0
ip address 10.10.20.1 255.255.255.192
no ip redirects
no ip directed-broadcast
ip pim sparse-mode
ip mroute-cache distributed
load-interval 30
 srp topology-timer 1
 srp ips wtr-timer 10
!

router ospf 100
network 10.10.10.0 0.0.0.255 area 1
network 10.10.20.0 0.0.0.255 area 0
network 10.0.0.1 0.0.0.0 area 0

auto-cost reference-bandwidth 2488
!
ip classless
!
```

GSR B Configuration

```
GSR+B:
Building configuration...
Current configuration:
!
```

```
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname GSR+B
!
!
hw-module slot 4 srp
!
ip subnet-zero
no ip domain-lookup
ip multicast-routing distributed
 ip pim rp-address 10.8.1.20 1
!
interface Loopback0
ip address 10.0.0.2 255.255.255.252
no ip directed-broadcast
!
interface SRP1/0
ip address 10.10.30.1 255.255.255.192
no ip redirects
no ip directed-broadcast
ip pim sparse-mode
ip mroute-cache distributed
load-interval 30
!
interface Ethernet0
ip address 10.100.1.5 255.255.255.0
no ip directed-broadcast
no ip route-cache cef
!
interface SRP4/0
ip address 10.10.20.2 255.255.255.192
no ip redirects
no ip directed-broadcast
ip pim sparse-mode
ip mroute-cache distributed
load-interval 30
 srp topology-timer 1
 srp ips wtr-timer 10
!
router ospf 100

network 10.10.30.0 0.0.0.255 area 2
 network 10.10.20.0 0.0.0.255 area 0
 network 10.0.0.2 0.0.0.0 area 0
auto-cost reference-bandwidth 2488
!
ip classless
```

CHAPTER 15

Configuring Serial Interfaces

This module describes the configuration of serial interfaces.

Before you configure a serial interface, you must configure the clear channel T3/E3 controller or channelized T1/E1controller (DS0 channel) that is associated with that interface.

Feature History for Configuring Serial Controller Interfaces

Release	Modification
Release 3.4.1	This feature was introduced on the Cisco CRS-1 Router. Support was added on the Cisco CRS-1 Router for the following hardware: • Cisco CRS-1 SIP-800 • Cisco 2-Port and 4-Port Clear Channel T3/E3 SPA

Prerequisites for Configuring Serial Interfaces

Before configuring serial interfaces, ensure that the following tasks and conditions are met:

- You must be in a user group associated with a task group that includes the proper task IDs. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You have installed a 2-Port or 4-Port Clear Channel T3/E3 SPA.
- You should have configured the clear channel T3/E3 controller controller that is associated with the serial interface you want to configure, as described in the *Configuring Clear Channel T3/E3 Controllers* module in this manual.

Information About Configuring Serial Interfaces

To configure serial interfaces, study the following concepts:

High-Level Overview: Serial Interface Configuration on Clear-Channel SPAs

This table provides a high-level overview of the tasks required to configure a T3 serial interface on the Cisco 2-Port and 4-Port Clear Channel T3/E3 SPA.

Table 11: Overview: Configuring a T3 Serial Interface on a Clear Channel SPA

Step	Task	Module	Section
1.	Use the **hw-module subslot** command to set serial mode for the SPA to be T3, if necessary. **Note** By default, the 2-Port and 4-Port Clear Channel T3/E3 SPA is set to run in T3 mode.	*Configuring Clear Channel T3/E3 Controllers*	Setting the Card Type
2.	Configure the T3 controller.	*Configuring Clear Channel T3/E3 Controllers*	Setting the Card Type
3.	Configure the serial interface that is associated with the T3 controller you configured in Step 2.	*Configuring Serial Interfaces*	How to Configure Serial Interfaces

This table provides a high-level overview of the tasks required to configure an E3 serial interface on a 2-Port and 4-Port Clear Channel T3/E3 SPA.

Table 12: Overview: Configuring an E3 Serial Interface on a Clear Channel SPA

Step	Task	Module	Section
1.	Use the **hw-module subslot** command to set serial mode for the SPA to be E3.	*Configuring Clear Channel T3/E3 Controllers*	Setting the Card Type
2.	Configure the E3 controller.	*Configuring Clear Channel T3/E3 Controllers*	Setting the Card Type

Step	Task	Module	Section
3.	Configure the serial interface that is associated with the E3 controller you configured in Step 2.	*Configuring Serial Interfaces*	How to Configure Serial Interfaces

Cisco HDLC Encapsulation

Cisco High-Level Data Link Controller (HDLC) is the Cisco proprietary protocol for sending data over synchronous serial links using HDLC. Cisco HDLC also provides a simple control protocol called Serial Line Address Resolution Protocol (SLARP) to maintain serial link keepalives. HDLC is the default encapsulation type for serial interfaces under Cisco IOS XR software. Cisco HDLC is the default for data encapsulation at Layer 2 (data link) of the Open System Interconnection (OSI) stack for efficient packet delineation and error control.

Note Cisco HDLC is the default encapsulation type for the serial interfaces.

Cisco HDLC uses keepalives to monitor the link state, as described in the Keepalive Timer.

Note Use the **debug chdlc slarp packet** command to display information about the Serial Line Address Resolution Protocol (SLARP) packets that are sent to the peer after the keepalive timer has been configured.

PPP Encapsulation

PPP is a standard protocol used to send data over synchronous serial links. PPP also provides a Link Control Protocol (LCP) for negotiating properties of the link. LCP uses echo requests and responses to monitor the continuing availability of the link.

Note When an interface is configured with PPP encapsulation, a link is declared down, and full LCP negotiation is re-initiated after five ECHOREQ packets are sent without receiving an ECHOREP response.

PPP provides the following Network Control Protocols (NCPs) for negotiating properties of data protocols that will run on the link:

- IP Control Protocol (IPCP) to negotiate IP properties
- Multiprotocol Label Switching control processor (MPLSCP) to negotiate MPLS properties
- Cisco Discovery Protocol control processor (CDPCP) to negotiate CDP properties
- IPv6CP to negotiate IP Version 6 (IPv6) properties
- Open Systems Interconnection control processor (OSICP) to negotiate OSI properties

PPP uses keepalives to monitor the link state, as described in the Keepalive Timer.

PPP supports the following authentication protocols, which require a remote device to prove its identity before allowing data traffic to flow over a connection:

- Challenge Handshake Authentication Protocol (CHAP)—CHAP authentication sends a challenge message to the remote device. The remote device encrypts the challenge value with a shared secret and returns the encrypted value and its name to the local router in a response message. The local router attempts to match the name of the remote device with an associated secret stored in the local username or remote security server database; it uses the stored secret to encrypt the original challenge and verify that the encrypted values match.
- Microsoft Challenge Handshake Authentication Protocol (MS-CHAP)—MS-CHAP is the Microsoft version of CHAP. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a personal computer using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server.
- Password Authentication Protocol (PAP)—PAP authentication requires the remote device to send a name and a password, which are checked against a matching entry in the local username database or in the remote security server database.

Note For more information on enabling and configuring PPP authentication protocols, see the *Configuring PPP* module in this manual.

Use the **ppp authentication** command in interface configuration mode to enable CHAP, MS-CHAP, and PAP on a serial interface.

Note Enabling or disabling PPP authentication does not effect the local router's willingness to authenticate itself to the remote device.

Multilink PPP

Multilink Point-to-Point Protocol (MLPPP) is supported on these SPAs:

MLPPP provides a method for combining multiple physical links into one logical link. The implementation of MLPPP combines multiple PPP serial interfaces into one multilink interface. MLPPP performs the fragmenting, reassembling, and sequencing of datagrams across multiple PPP links.

MLPPP provides the same features that are supported on PPP Serial interfaces with the exception of QoS. It also provides the following additional features:

- Fragment sizes of 128, 256, and 512 bytes
- Long sequence numbers (24-bit)
- Lost fragment detection timeout period of 80 ms
- Minimum-active-links configuration option
- LCP echo request/reply support over multilink interface
- Full T1 and E1 framed and unframed links

For more information about configuring MLPPP on a serial interface, see the *Configuring PPP* module in this document.

Keepalive Timer

Cisco keepalives are useful for monitoring the link state. Periodic keepalives are sent to and received from the peer at a frequency determined by the value of the keepalive timer. If an acceptable keepalive response is not received from the peer, the link makes the transition to the down state. As soon as an acceptable keepalive response is obtained from the peer or if keepalives are disabled, the link makes the transition to the up state.

Note The **keepalive** command applies to serial interfaces using HDLC or PPP encapsulation. It does not apply to serial interfaces using Frame Relay encapsulation.

For each encapsulation type, a certain number of keepalives ignored by a peer triggers the serial interface to transition to the down state. For HDLC encapsulation, three ignored keepalives causes the interface to be brought down. For PPP encapsulation, five ignored keepalives causes the interface to be brought down. ECHOREQ packets are sent out only when LCP negotiation is complete (for example, when LCP is open).

Note Use the **keepalive** command in interface configuration mode to set the frequency at which LCP sends ECHOREQ packets to its peer. To restore the system to the default keepalive interval of 10 seconds, use the **keepalive** command with **no** argument. To disable keepalives, use the **keepalive** disable command. For both PPP and Cisco HDLC, a keepalive of 0 disables keepalives and is reported in the **show running-config** command output as **keepalive disable**.

Note Before performing a Minimal Disruptive Restart (MDR) upgrade, we recommend configuring a keepalive interval of 10 seconds or more on a Cisco CRS-1 Router.

When LCP is running on the peer and receives an ECHOREQ packet, it responds with an echo reply (ECHOREP) packet, regardless of whether keepalives are enabled on the peer.

Keepalives are independent between the two peers. One peer end can have keepalives enabled; the other end can have them disabled. Even if keepalives are disabled locally, LCP still responds with ECHOREP packets to the ECHOREQ packets it receives. Similarly, LCP also works if the period of keepalives at each end is different.

Note Use the **debug chdlc slarp packet** command and other Cisco HDLC **debug** commands to display information about the Serial Line Address Resolution Protocol (SLARP) packets that are sent to the peer after the keepalive timer has been configured.

Frame Relay Encapsulation

When Frame Relay encapsulation is enabled on a serial interface, the interface configuration is hierarchical and comprises the following elements:

1. The serial main interface comprises the physical interface and port. If you are not using the serial interface to support Cisco HDLC and PPP encapsulated connections, then you must configure subinterfaces with permanent virtual circuits (PVCs) under the serial main interface. Frame Relay connections are supported on PVCs only.
2. Serial subinterfaces are configured under the serial main interface. A serial subinterface does not actively carry traffic until you configure a PVC under the serial subinterface. Layer 3 configuration typically takes place on the subinterface.
3. Point-to-point PVCs are configured under a serial subinterface. You cannot configure a PVC directly under a main interface. A single point-to-point PVC is allowed per subinterface. PVCs use a predefined circuit path and fail if the path is interrupted. PVCs remain active until the circuit is removed from either configuration. Connections on the serial PVC support Frame Relay encapsulation only.

Note The administrative state of a parent interface drives the state of the subinterface and its PVC. When the administrative state of a parent interface or subinterface changes, so does the administrative state of any child PVC configured under that parent interface or subinterface.

To configure Frame Relay encapsulation on serial interfaces, use the **encapsulation frame-relay** command.

Frame Relay interfaces support two types of encapsulated frames:

- Cisco (default)
- IETF

Use the **encap** command in PVC configuration mode to configure Cisco or IETF encapsulation on a PVC. If the encapsulation type is not configured explicitly for a PVC, then that PVC inherits the encapsulation type from the main serial interface.

Note Cisco encapsulation is required on serial main interfaces that are configured for MPLS. IETF encapsulation is not supported for MPLS.

Before you configure Frame Relay encapsulation on an interface, you must verify that all prior Layer 3 configuration is removed from that interface. For example, you must ensure that there is no IP address configured directly under the main interface; otherwise, any Frame Relay configuration done under the main interface will not be viable.

LMI on Frame Relay Interfaces

The Local Management Interface (LMI) protocol monitors the addition, deletion, and status of PVCs. LMI also verifies the integrity of the link that forms a Frame Relay UNI interface. By default, **cisco** LMI is enabled on all PVCs. However, you can modify the default LMI type to be ANSI or Q.933.

If the LMI type is **cisco** (the default LMI type), the maximum number of PVCs that can be supported under a single interface is related to the MTU size of the main interface. Use the following formula to calculate the maximum number of PVCs supported on a card or SPA:

```
(MTU - 13)/8 = maximum number of PVCs
```

Note The default setting of the **mtu** command for a serial interface is 1504 bytes. Therefore, the default numbers of PVCs supported on a serial interface configured with **cisco** LMI is 186.

Layer 2 Tunnel Protocol Version 3-Based Layer 2 VPN on Frame Relay

The Layer 2 Tunnel Protocol Version 3 (L2TPv3) feature defines the L2TP protocol for tunneling Layer 2 payloads over an IP core network using Layer 2 virtual private networks (VPNs).

L2TPv3 is a tunneling protocol used for transporting Layer 2 protocols. It can operate in a number of different configurations and tunnel a number of different Layer 2 protocols and connections over a packet-switched network.

Before you can configure L2TPv3, you need to configure a connection between the two attachment circuits (ACs) that will host the L2TPv3 psuedowire. Cisco IOS XR software supports a point-to-point, end-to-end service, where two ACs are connected together.

This module describes how to configure a Layer 2 AC on a Frame Relay encapsulated serial interface.

Note Serial interfaces support DLCI mode layer 2 ACs only; layer 2 port mode ACs are not supported on serial interfaces.

For detailed information about configuring L2TPv3 in your network, see the "*Implementing Layer 2 Tunnel Protocol Version 3"* module of the *Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router*. For detailed information about configuring L2VPNs, see the "*Implementing MPLS Layer 2 VPNs"* module of the *Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router*.

Default Settings for Serial Interface Configurations

When an interface is enabled on a T3/E3 SPA, and no additional configuration commands are applied, the default interface settings shown in this table are present. These default settings can be changed by configuration.

Table 13: Serial Interface Default Settings

Parameter	Configuration File Entry	Default Settings
Keepalive **Note** The **keepalive** command applies to serial interfaces using HDLC or PPP encapsulation. It does not apply to serial interfaces using Frame Relay encapsulation.	**keepalive** [**disable**] **no keepalive**	keepalive 10 seconds
Encapsulation	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**ietf**]]	hdlc
Maximum transmission unit (MTU)	**mtu** *bytes*	1504 bytes
Cyclic redundancy check (CRC)	**crc** [**16** | **32]**	16
Data stream inversion on a serial interface	**invert**	Data stream is not inverted
Payload scrambling (encryption)	**scramble**	Scrambling is disabled.
Number of High-Level Data Link Control (HDLC) flag sequences to be inserted between the packets	**transmit-delay**	Default is 0 (disabled).

Note Default settings do not appear in the output of the show running-config command.

Serial Interface Naming Notation

The naming notation for serial interfaces on a clear channel SPA is *rack*/*slot*/*module*/*port*, as shown in this example:

```
interface serial 0/0/1/2
```

The naming notation for T1, E1, and DS0 interfaces on a channelized SPA is *rack*/*slot*/*module*/*port*/*channel-num:channel-group-number*, as shown in the following example:

```
interface serial 0/0/1/2/4:3
```

If a subinterface and PVC are configured under the serial interface, then the router includes the subinterface number at the end of the serial interface address. In this case, the naming notation is *rack*/*slot*/*module*/*port*[/*channel-num:channel-group-number*].*subinterface,* as shown in the following examples:

```
interface serial 0/0/1/2.1
interface serial 0/0/1/2/4:3.1
```

Note A slash between values is required as part of the notation.

The naming notation syntax for serial interfaces is as follows:

- *rack*: Chassis number of the rack.
- *slot*: Physical slot number of the modular services card or line card.
- *module*: Module number. Shared port adapters (SPAs) are referenced by their subslot number.
- *port*: Physical port number of the controller.
- *channel-num*: T1 or E1 channel number. T1 channels range from 0 to 23; E1 channels range from 0 to 31.
- *channel-group-number*: Time slot number. T1 time slots range from 1 to 24; E1 time slots range from 1 to 31. The *channel-group-number* is preceded by a colon and not a slash.
- *subinterface*: Subinterface number.

Use the question mark (**?**) online help function following the **serial** keyword to view a list of all valid interface choices.

How to Configure Serial Interfaces

After you have configured a channelized or clear channel T3/E3 controller, as described in the *Configuring Clear Channel T3/E3 Controllers* module in this document, you can configure the serial interfaces associated with that controller.

Bringing Up a Serial Interface

This task describes the commands used to bring up a serial interface.

Before you begin

- The Cisco CRS-1 Router must have the following SIP and SPA installed and running Cisco IOS XR software:
- Cisco CRS-1 SIP-800
- 2-Port and 4-Port T3/E3 Serial SPA

Restrictions

The configuration on both ends of the serial connection must match for the interface to be active.

SUMMARY STEPS

1. **show interfaces**
2. **configure**
3. **interface serial** *interface-path-id*
4. **ipv4 address ip-address**
5. **no shutdown**
6. **end** or **commit**
7. **exit**
8. **exit**
9. Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.
10. **show ipv4 interface brief**
11. **show interfaces serial** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show interfaces** **Example:** `RP/0/RP0/CPU0:router# show interfaces`	(Optional) Displays configured interfaces. • Use this command to also confirm that the router recognizes the PLIM card.
Step 2	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 3	**interface serial** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0`	Specifies the serial interface name and notation *rack/slot/module/port,* and enters interface configuration mode.
Step 4	**ipv4 address ip-address** **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.1 255.255.255.224`	Assigns an IP address and subnet mask to the interface. **Note** • Skip this step if you are configuring Frame Relay encapsulation on this interface. For Frame Relay, the IP address and subnet mask are configured under the subinterface.
Step 5	**no shutdown** **Example:** `RP/0/RP0/CPU0:router (config-if)# no shutdown`	Removes the shutdown configuration. **Note** • Removal of the shutdown configuration eliminates the forced administrative down on the interface, enabling it to move to an up or down state (assuming the parent SONET layer is not configured administratively down).

	Command or Action	Purpose

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

Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode.

Step 9	Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.
	Example:
	`RP/0/RP0/CPU0:router# show interfaces`
	
	`RP/0/RP0/CPU0:router# configure`
	`RP/0/RP0/CPU0:router (config)# interface serial 0/1/0/1`
	`RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.2 255.255.255.224`
	`RP/0/RP0/CPU0:router (config-if)# no shutdown`
	`RP/0/RP0/CPU0:router (config-if)# commit`
	`RP/0/RP0/CPU0:router (config-if)# exit`
	`RP/0/RP0/CPU0:router (config)# exit`
		Note • The configuration on both ends of the serial connection must match.

Step 10	**show ipv4 interface brief**	Verifies that the interface is active and properly configured.

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router # show ipv4 interface brief`	If you have brought up a serial interface properly, the "Status" field for that interface in the **show ipv4 interface brief** command output displays "Up."
Step 11	**show interfaces serial** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces serial 0/1/0/0`	(Optional) Displays the interface configuration.

What to do next

To modify the default configuration of the serial interface you just brought up, see the "Configuring Optional Serial Interface Parameters" section on page 564.

Configuring Optional Serial Interface Parameters

This task describes the commands used to modify the default configuration on a serial interface.

Before you begin

Before you modify the default serial interface configuration, you must bring up the serial interface and remove the shutdown configuration, as described in the Bringing Up a Serial Interface.

Restrictions

The configuration on both ends of the serial connection must match for the interface to be active.

SUMMARY STEPS

1. **configure**
2. **interface serial** *interface-path-id*
3. **encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]
4. **serial**
5. **crc** *length*
6. **invert**
7. **scramble**
8. **transmit-delay** *hdlc-flags*
9. **end** or **commit**
10. **exit**
11. **exit**
12. **exit**
13. **show interfaces serial** [*interface-path-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface serial** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0`	Specifies the serial interface name and notation *rack/slot/module/port,* and enters interface configuration mode.
Step 3	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**] **Example:** `RP/0/RP0/CPU0:router(config-if)# encapsulation hdlc`	(Optional) Configures the interface encapsulation parameters and details such as HDLC, PPP or Frame Relay. **Note** • The default encapsulation is **hdlc**.
Step 4	**serial** **Example:** `RP/0/RP0/CPU0:router(config-if)# serial`	(Optional) Enters serial submode to configure the serial parameters.
Step 5	**crc** *length* **Example:** `RP/0/RP0/CPU0:ios(config-if-serial)# crc 32`	(Optional) Specifies the length of the cyclic redundancy check (CRC) for the interface. Enter the **16** keyword to specify 16-bit CRC mode, or enter the **32** keyword to specify 32-bit CRC mode. **Note** • The default is CRC length is 16.
Step 6	**invert** **Example:** `RP/0/RP0/CPU0:ios(config-if-serial)# inverts`	(Optional) Inverts the data stream.
Step 7	**scramble** **Example:** `RP/0/RP0/CPU0:ios(config-if-serial)# scramble`	(Optional) Enables payload scrambling on the interface. **Note** • Payload scrambling is disabled on the interface.
Step 8	**transmit-delay** *hdlc-flags* **Example:** `RP/0/RP0/CPU0:ios(config-if-serial)# transmit-delay 10`	(Optional) Specifies a transmit delay on the interface. Values can be from 0 to 128. **Note** • Transmit delay is disabled by default (the transmit delay is set to **0**).
Step 9	**end** or **commit**	Saves configuration changes.

	Command or Action	Purpose

	Example:
	`RP/0/RP0/CPU0:router (config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 10	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if-serial)# exit`	Exits serial configuration mode.
Step 11	**exit** **Example:** `RP/0/RP0/CPU0:router (config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 12	**exit** **Example:** `RP/0/RP0/CPU0:router (config)# exit`	Exits global configuration mode and enters EXEC mode.
Step 13	**show interfaces serial** [*interface-path-id*] **Example:** `RP/0/RP0/CPU0:router# show interface serial 0/1/0/0`	(Optional) Displays general information for the specified serial interface.

What to do next

- To create a point-to-point Frame Relay subinterface with a PVC on the serial interface you just brought up, see the "Creating a Point-to-Point Serial Subinterface with a PVC" section on page 567.
- To configure PPP authentication on serial interfaces with PPP encapsulation, see the "Configuring PPP on the Cisco ASR 9000 Series Router" module later in this manual.

- To modify the default keepalive configuration, see the “Modifying the Keepalive Interval on Serial Interfaces” section on page 572.

Creating a Point-to-Point Serial Subinterface with a PVC

The procedure in this section creates a point-to-point serial subinterface and configures a permanent virtual circuit (PVC) on that serial subinterface.

Note Subinterface and PVC creation is supported on interfaces with Frame Relay encapsulation only.

Before you begin

Before you can create a subinterface on a serial interface, you must bring up the main serial interface with Frame Relay encapsulation, as described in the Bringing Up a Serial Interface.

Restrictions

Only one PVC can be configured for each point-to-point serial subinterface.

SUMMARY STEPS

1. **configure**
2. **interface serial** *interface-path-id.subinterface* **point-to-point**
3. **ipv4 address** *ipv4_address/prefix*
4. **pvc** *dlci*
5. **end** or **commit**
6. Repeat Step 1 through Step 5 to bring up the serial subinterface and any associated PVC at the other end of the connection.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface serial** *interface-path-id.subinterface* **point-to-point** **Example:** `RP/0/RP0/CPU0:router (config)# interface serial 0/1/0/0.1`	Enters serial subinterface configuration mode.
Step 3	**ipv4 address** *ipv4_address/prefix* **Example:**	Assigns an IP address and subnet mask to the subinterface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router (config-subif)#ipv4 address 10.46.8.6/24`	
Step 4	**pvc** *dlci* **Example:** `RP/0/RP0/CPU0:router (config-subif)# pvc 20`	Creates a serial permanent virtual circuit (PVC) and enters Frame Relay PVC configuration submode. Replace *dlci* with a PVC identifier, in the range from 16 to 1007. **Note** • Only one PVC is allowed per subinterface.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# 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.

Step 6	Repeat Step 1 through Step 5 to bring up the serial subinterface and any associated PVC at the other end of the connection.
	Example:
	`RP/0/RP0/CPU0:router# configure`
	`RP/0/RP0/CPU0:router (config)# interface serial 0/1/0/1.1`
	`RP/0/RP0/CPU0:router (config-subif)#ipv4 address 10.46.8.5/24`
	`RP/0/RP0/CPU0:router (config-subif)# pvc 20`
	`RP/0/RP0/CPU0:router (config-fr-vc)# commit`
		Note • The DLCI (or PVC identifier) must match on both ends of the subinterface connection.
		• When assigning an IP address and subnet mask to the subinterface at the other end of the connection, keep in mind that the addresses at both ends of the connection must be in the same subnet.

What to do next

- To configure optional PVC parameters, see the "Configuring Optional Serial Interface Parameters" section on page 564.
- To attach a Layer 3 QOS service policy to the PVC under the PVC submode, refer to the appropriate Cisco IOS XR software configuration guide.

Configuring Optional PVC Parameters

This task describes the commands you can use to modify the default configuration on a serial PVC.

Before you begin

Before you can modify the default PVC configuration, you must create the PVC on a serial subinterface, as described in the Creating a Point-to-Point Serial Subinterface with a PVC.

Restrictions

- The DLCI (or PVI identifier) must match on both ends of the PVC for the connection to be active.
- To change the PVC DLCI, you must delete the PVC and then add it back with the new DLCI.

SUMMARY STEPS

1. Configuring Optional PVC Parameters
2. **interface serial** *interface-path-id.subinterface*
3. **pvc** *dlci*
4. **encap** [**cisco** | **ietf**]
5. **service-policy** {**input** | **output**} *policy-map*
6. **end** or **commit**
7. Repeat Step 1 through Step 6 to bring up the serial subinterface and any associated PVC at the other end of the connection.
8. **show frame-relay pvc** *dlci-number*
9. **show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**} or **show policy-map type qos interface serial** *interface-path-id.subinterface* {**input** | **output**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	Configuring Optional PVC Parameters **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface serial** *interface-path-id.subinterface* **Example:** `RP/0/RP0/CPU0:router (config)# interface serial 0/1/0/0.1`	Enters serial subinterface configuration mode.

	Command or Action	Purpose
Step 3	**pvc** *dlci* **Example:** `RP/0/RP0/CPU0:router (config-subif)# pvc 20`	Enters subinterface configuration mode for the PVC.
Step 4	**encap** [**cisco** | **ietf**] **Example:** `RP/0/RP0/CPU0:router (config-fr-vc)# encap ietf`	(Optional) Configures the encapsulation for a Frame Relay PVC. **Note** • If the encapsulation type is not configured explicitly for a PVC, then that PVC inherits the encapsulation type from the main serial interface.
Step 5	**service-policy** {**input** | **output**} *policy-map* **Example:** `RP/0/RP0/CPU0:router (config-fr-vc)# service-policy output policy1`	Attaches a policy map to an input subinterface or output subinterface. Once attached, the policy map is used as the service policy for the subinterface.

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

Step 7	Repeat Step 1 through Step 6 to bring up the serial subinterface and any associated PVC at the other end of the connection. **Example:** `RP/0/RP0/CPU0:router# configure`	**Note** • The configuration on both ends of the subinterface connection must match.

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router (config)# interface serial 0/1/0/1.1`
	
	`RP/0/RP0/CPU0:router (config-subif)# pvc 20`
	
	`RP/0/RP0/CPU0:router (config-fr-vc)# encap cisco`
	`RP/0/RP0/CPU0:router (config-fr-vc)# commit`	

Step 8	**show frame-relay pvc** *dlci-number* **Example:** `RP/0/RP0/CPU0:router# show frame-relay pvc 20`	(Optional) Verifies the configuration of specified serial interface.

Step 9	**show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**} or **show policy-map type qos interface serial** *interface-path-id.subinterface* {**input** | **output**}
	
	Example:
	
	`RP/0/RP0/CPU0:router# show policy-map interface serial 0/1/0/0.1 output`
	
	or
	
	`RP/0/RP0/CPU0:router# show policy-map type qos interface serial 0/1/0/0.1 output`
		(Optional) Displays the statistics and the configurations of the input and output policies that are attached to a subinterface.

Modifying the Keepalive Interval on Serial Interfaces

Perform this task to modify the keepalive interval on serial interfaces that have Cisco HDLC or PPP encapsulation enabled.

Note When you enable Cisco HDLC or PPP encapsulation on a serial interface, the default keepalive interval is 10 seconds. Use this procedure to modify that default keepalive interval.

Cisco HDLC is enabled by default on serial interfaces.

Before you begin

Before modifying the keepalive timer configuration, ensure that Cisco HDLC or PPP encapsulation is enabled on the interface. Use the **encapsulation** command to enable Cisco HDLC or PPP encapsulation on the interface, as described in the Configuring Optional Serial Interface Parameters.

Restrictions

- Before performing a Minimal Disruptive Restart (MDR) upgrade, we recommend configuring a keepalive interval of 10 seconds or more on a Cisco CRS-1 Router.

SUMMARY STEPS

1. **configure**
2. **interface serial** *interface-path-id*
3. **keepalive** {*seconds* | **disable**} or **no keepalive**
4. **end** or **commit**
5. **show interfaces serial** *interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface serial** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0`	Specifies the serial interface name and notation *rack/slot/module/port* and enters interface configuration mode.

Step 3	**keepalive** {*seconds* | **disable**} or **no keepalive**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# keepalive 3`
	or
	`RP/0/RP0/CPU0:router(config-if)# no keepalive`
		Specifies the number of seconds between keepalive messages.
		• Use the **keepalive disable** command, the **no keepalive**, or the **keepalive** command with an argument of 0 to disable the keepalive feature.
		• The range is from 1 to 30 seconds. The default is 10 seconds.
		• If keepalives are configured on an interface, use the **no keepalive** command to disable the keepalive feature before configuring Frame Relay encapsulation on that interface.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

	Command or Action	Purpose
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show interfaces serial** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces serial 0/1/0/0`	(Optional) Verifies the interface configuration.

How to Configure a Layer 2 Attachment Circuit

The Layer 2 AC configuration tasks are described in the following procedures:

- Creating a Serial Layer 2 Subinterface with a PVC
- Configuring Optional Serial Layer 2 PVC Parameters

Note After you configure an interface for Layer 2 switching, no routing commands such as **ipv4 address** are permissible. If any routing commands are configured on the interface, then the **l2transport** command is rejected.

Creating a Serial Layer 2 Subinterface with a PVC

The procedure in this section creates a Layer 2 subinterface with a PVC.

Before you begin

Before you can create a subinterface on a serial interface, you must bring up a serial interface, as described in the Bringing Up a Serial Interface.

Restrictions

Only one PVC can be configured for each serial subinterface.

SUMMARY STEPS

1. **configure**
2. **interface serial** *interface-path-id.subinterface* **l2transport**
3. **pvc** *vpi/vci*
4. **end** or **commit**
5. Repeat Step 1 through Step 4 to bring up the serial subinterface and any associated PVC at the other end of the AC.

DETAILED STEPS

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 1</td><td>configure
Example:
<code>RP/0/RP0/CPU0:router# configure</code></td><td>Enters global configuration mode.</td></tr>
<tr><td>Step 2</td><td>interface serial interface-path-id.subinterface l2transport
Example:
<code>RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0.1 l2transport</code></td><td>Creates a subinterface and enters serial subinterface configuration mode for that subinterface.</td></tr>
<tr><td>Step 3</td><td>pvc vpi/vci
Example:
<code>RP/0/RP0/CPU0:router(config-if)# pvc 5/20</code></td><td>Creates a serial permanent virtual circuit (PVC) and enters serial Layer 2 transport PVC configuration mode.
Note • Only one PVC is allowed per subinterface.</td></tr>
<tr><td>Step 4</td><td>end or commit
Example:
<code>RP/0/RP0/CPU0:router(config-fr-vc)# end</code>
or
<code>RP/0/RP0/CPU0:router(config-fr-vc)# commit</code></td><td>Saves configuration changes.
• When you issue the end command, the system prompts you to commit changes:
<code>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</code>
- 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.</td></tr>
<tr><td>Step 5</td><td>Repeat Step 1 through Step 4 to bring up the serial subinterface and any associated PVC at the other end of the AC.</td><td>Brings up the AC.
Note • The configuration on both ends of the AC must match.</td></tr>
</table>

What to do next

- To configure optional PVC parameters, see the "Configuring Optional Serial Layer 2 PVC Parameters" section on page 577.
- For detailed information about configuring L2TPv3 in your network, see the "Implementing Layer 2 Tunnel Protocol Version 3" module of the Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router. For detailed information about configuring L2VPNs, see the "Implementing MPLS Layer 2 VPNs" module of the Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router.

Configuring Optional Serial Layer 2 PVC Parameters

This task describes the commands you can use to modify the default configuration on a serial Layer 2 PVC.

Before you begin

Before you can modify the default PVC configuration, you must create the PVC on a Layer 2 subinterface, as described in the Creating a Serial Layer 2 Subinterface with a PVC.

Restrictions

The configuration on both ends of the PVC must match for the connection to be active.

SUMMARY STEPS

1. **configure**
2. **interface serial** *interface-path-id.subinterface* **l2transport**
3. **pvc** *dlci*
4. **encap** {**cisco** | **ietf**}
5. **service-policy** {**input** | **output**} *policy-map*
6. **end** or **commit**
7. Repeat Step 1 through Step 7 to configure the PVC at the other end of the AC.
8. **show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**} or **show policy-map type qos interface serial** *interface-path-id.subinterface* {**input** | **output**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface serial** *interface-path-id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0.1 l2transport`	Enters serial subinterface configuration mode for a Layer 2 serial subinterface.

	Command or Action	Purpose
Step 3	**pvc** *dlci* **Example:** `RP/0/RP0/CPU0:router(config-if)# pvc 100`	Enters serial Frame Relay PVC configuration mode for the specified PVC.
Step 4	**encap** {**cisco** | **ietf**} **Example:** `RP/0/RP0/CPU0:router(config-fr-vc)# encapsulation aal5`	Configures the encapsulation for a Frame Relay PVC.
Step 5	**service-policy** {**input** | **output**} *policy-map* **Example:** `RP/0/RP0/CPU0:router (config-subif)# service-policy output policy1`	Attaches a policy map to an input subinterface or output subinterface. Once attached, the policy map is used as the service policy for the subinterface.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-serial-l2transport-pvc)# end`
	or
	`RP/0/RP0/CPU0:router(config-serial-l2transport-pvc)# 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.
		Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 7	Repeat Step 1 through Step 7 to configure the PVC at the other end of the AC.	Brings up the AC. **Note** • The configuration on both ends of the connection must match.

	Command or Action	Purpose

Step 8	**show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**} or **show policy-map type qos interface serial** *interface-path-id.subinterface* {**input** | **output**}
	Example:
	`RP/0/RP0/CPU0:router# show policy-map interface pos 0/1/0/0.1 output`
	or
	`RP/0/RP0/CPU0:router# show policy-map type qos interface pos 0/1/0/0.1 output`
		(Optional) Displays the statistics and the configurations of the input and output policies that are attached to a subinterface.

What to do next

- To configure a point-to-point pseudowire XConnect on the AC you just created, see the "Implementing Layer 2 Tunnel Protocol Version 3" module of the Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router.
- To configure an L2VPN, see the "Implementing MPLS Layer 2 VPNs" module of the Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router.

Configuration Examples for Serial Interfaces

This section provides the following configuration examples:

Bringing Up and Configuring a Serial Interface with Cisco HDLC Encapsulation: Example

The following example shows how to bring up a basic serial interface with Cisco HDLC encapsulation:

```
RP/0/RP0/CPU0:Router#config
RP/0/RP0/CPU0:Router(config)# interface serial 0/3/0/0/0:0
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 192.0.2.2 255.255.255.252
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

The following example shows how to configure the interval between keepalive messages to be 10 seconds:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RP0/CPU0:router(config-if)# keepalive 10
RP/0/RP0/CPU0:router(config-if)# commit
```

The following example shows how to modify the optional serial interface parameters:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RP0/CPU0:Router(config-if)# serial
RP/0/RP0/CPU0:Router(config-if-serial)# crc 16
RP/0/RP0/CPU0:Router(config-if-serial)# invert
RP/0/RP0/CPU0:Router(config-if-serial)# scramble
RP/0/RP0/CPU0:Router(config-if-serial)# transmit-delay 3
RP/0/RP0/CPU0:Router(config-if-serial)# commit
```

The following is sample output from the **show interfaces serial** command:

```
RP/0/RP0/CPU0:Router# show interfaces serial 0/0/3/0/5:23
Serial0/0/3/0/5:23 is down, line protocol is down
  Hardware is Serial network interface(s)
  Internet address is Unknown
  MTU 1504 bytes, BW 64 Kbit
     reliability 143/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16,  loopback not set, keepalive set (10 sec)
  Last clearing of "show interface" counters 18:11:15
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2764 packets input, 2816 bytes, 3046 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     3046 input errors, 1 CRC, 0 frame, 0 overrun, 2764 ignored, 281 abort
     2764 packets output, 60804 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Configuring a Serial Interface with Frame Relay Encapsulation: Example

The following example shows how to create a serial interface on a SPA with Frame Relay encapsulation and a serial subinterface with a PVC on router 1:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0
RP/0/RP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/RP0/CPU0:router(config-if)#frame-relay intf-type dce
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0.1 point-to-point
RP/0/RP0/CPU0:router (config-subif)#ipv4 address 10.20.3.1/24

RP/0/RP0/CPU0:router (config-subif)# pvc 16

RP/0/RP0/CPU0:router (config-fr-vc)# encapsulation ietf

RP/0/RP0/CPU0:router (config-fr-vc)# commit
```

```
RP/0/RP0/CPU0:router(config-fr-vc)# exit

RP/0/RP0/CPU0:router(config-subif)# exit

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

RP/0/RP0/CPU0:router# show interface serial 0/1/0/0
Wed Oct  8 04:14:39.946 PST DST
Serial0/1/0/0 is up, line protocol is up
  Interface state transitions: 5
  Hardware is Serial network interface(s)
  Internet address is 10.20.3.1/24
  MTU 4474 bytes, BW 44210 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 16,
  Scrambling is disabled, Invert data is disabled
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
  LMI enq recvd 880, LMI stat sent  880, LMI upd sent  0 , DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
  Last clearing of "show interface" counters 02:23:04
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     858 packets input, 11154 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     858 packets output, 12226 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

The following example shows how to create a serial interface on a SPA with Frame Relay encapsulation and a serial subinterface with a PVC on router 2, which is connected to router 1:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/1
RP/0/RP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/1.1 point-to-point
RP/0/RP0/CPU0:router (config-subif)#ipv4 address 10.20.3.2/24

RP/0/RP0/CPU0:router (config-subif)# pvc 16

RP/0/RP0/CPU0:router (config-fr-vc)# encapsulation ietf

RP/0/RP0/CPU0:router (config-fr-vc)# commit

RP/0/RP0/CPU0:router(config-fr-vc)# exit

RP/0/RP0/CPU0:router(config-subif)# exit
```

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

RP/0/RP0/CPU0:router# show interface serial 0/1/0/1
Wed Oct  8 04:13:45.046 PST DST
Serial0/1/0/1 is up, line protocol is up
  Interface state transitions: 7
  Hardware is Serial network interface(s)
  Internet address is Unknown
  MTU 4474 bytes, BW 44210 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 16,
  Scrambling is disabled, Invert data is disabled
  LMI enq sent  1110, LMI stat recvd 875, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 1023  LMI type is CISCO  frame relay DTE
  Last clearing of "show interface" counters 02:22:09
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     853 packets input, 12153 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     853 packets output, 11089 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

Configuring a Serial Interface with PPP Encapsulation: Example

The following example shows how to create and configure a serial interface with PPP encapsulation:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# ppp authentication chap MIS-access
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

The following example shows how to configure serial interface 0/3/0/0/0:0 to allow two additional retries after an initial authentication failure (for a total of three failed authentication attempts):

```
RP/0/RP0/CPU0:router# configuration
RP/0/RP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# ppp authentication chap
RP/0/RP0/CPU0:router(config-if)# ppp max-bad-auth 3
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

CHAPTER 16

Configuring PPP

This module describes the configuration of Point-to-Point Protocol (PPP) on POS and serial interfaces on the Cisco CRS-1 Router.

Feature History for Configuring PPP Interfaces

Release	Modification
Release 2.0	PPP authentication was introduced on theCisco CRS-1 Router.

Prerequisites for Configuring PPP

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 you can configure PPP authentication on a POS or serial interface, be sure that the following tasks and conditions are met:

- Your hardware must support POS or serial interfaces.
- You have enabled PPP encapsulation on your interface with the **encap ppp** command, as described in the appropriate module:
  - To enable PPP encapsulation on a POS interface, see the Configuring POS Interfaces, on page 327 module in this manual.
  - To enable PPP encapsulation on a serial interface, see the Configuring Serial Interfaces, on page 371 module in this manual.

Information About PPP

To configure PPP and related features, you should understand the information in this section:

PPP Authentication

When PPP authentication is configured on an interface, a host requires that the other host uniquely identify itself with a secure password before establishing a PPP connection. The password is unique and is known to both hosts.

PPP supports the following authentication protocols:

- Challenge-Handshake Authentication Protocol (CHAP)
- Microsoft extension to the CHAP protocol (MS-CHAP)
- Password Authentication Protocol (PAP).

When you first enable PPP on a POS or serial interface, no authentication is enabled on the interface until you configure a CHAP, MS-CHAP, or PAP secret password under that interface. Keep the following information in mind when configuring PPP on an interface:

- CHAP, MS-CHAP, and PAP can be configured on a single interface; however, only one authentication method is used at any one time. The order in which the authentication protocols are used is determined by the peer during the LCP negotiations. The first authentication method used is the one that is also supported by the peer.
- PAP is the least secure authentication protocol available on POS and serial interfaces. To ensure higher security for information that is sent over POS and serial interfaces, we recommend configuring CHAP or MS-CHAP authentication in addition to PAP authentication.
- Enabling or disabling PPP authentication does not effect the local router's willingness to authenticate itself to the remote device.
- The **ppp authentication** command is also used to specify the order in which CHAP, MS-CHAP, and PAP authentication is selected on the interface. You can enable CHAP, MS-CHAP, or PAP in any order. If you enable all three methods, the first method specified is requested during link negotiation. If the peer suggests using the second method, or refuses the first method, the second method is tried. Some remote devices support only one method. Base the order in which you specify methods on the remote device's ability to correctly negotiate the appropriate method and on the level of data line security you require. PAP usernames and passwords are sent as clear text strings, which can be intercepted and reused.

Caution If you use a *list-name* value that was not configured with the **aaa authentication ppp** command, your interface cannot authenticate the peer. For details on implementing the **aaa authentication** command with the **ppp** keyword, see the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module of *Cisco IOS XR System Security Command Referenc*e and *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

PAP Authentication

PAP provides a simple method for a remote node to establish its identity using a two-way handshake. After a PPP link is established between two hosts, a username and password pair is repeatedly sent by the remote node across the link (in clear text) until authentication is acknowledged, or until the connection is terminated.

PAP is not a secure authentication protocol. Passwords are sent across the link in clear text and there is no protection from playback or trial-and-error attacks. The remote node is in control of the frequency and timing of the login attempts.

CHAP Authentication

CHAP is defined in RFC 1994, and it verifies the identity of the peer by means of a three-way handshake. The steps that follow provide a general overview of the CHAP process:

SUMMARY STEPS

1. The CHAP authenticator sends a challenge message to the peer.
2. The peer responds with a value calculated through a one-way hash function.
3. The authenticator checks the response against its own calculation of the expected hash value. If the values match, then the authentication is successful. If the values do not match, then the connection is terminated.

DETAILED STEPS

Step 1 The CHAP authenticator sends a challenge message to the peer.

Step 2 The peer responds with a value calculated through a one-way hash function.

Step 3 The authenticator checks the response against its own calculation of the expected hash value. If the values match, then the authentication is successful. If the values do not match, then the connection is terminated.

This authentication method depends on a CHAP password known only to the authenticator and the peer. The CHAP password is not sent over the link. Although the authentication is only one-way, you can negotiate CHAP in both directions, with the help of the same CHAP password set for mutual authentication.

Note For CHAP authentication to be valid, the CHAP password must be identical on both hosts.

MS-CHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension to RFC 1994. MS-CHAP follows the same authentication process used by CHAP. In this case, however, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server (NAS).

Note For MS-CHAP authentication to be valid, the MS-CHAP password must be identical on both hosts.

How to Configure PPP

This section includes the following procedures:

Modifying the Default PPP Configuration

When you first enable PPP on an interface, the following default configuration applies:

- The interface resets itself immediately after an authentication failure.
- The maximum number of configuration requests without response permitted before all requests are stopped is 10.
- The maximum number of consecutive Configure Negative Acknowledgments (CONFNAKs) permitted before terminating a negotiation is 5.
- The maximum number of terminate requests (TermReqs) without response permitted before the Link Control Protocol (LCP) or Network Control Protocol (NCP) is closed is 2.
- Maximum time to wait for a response to an authentication packet is 10 seconds.
- Maximum time to wait for a response during PPP negotiation is 3 seconds.

This task explains how to modify the basic PPP configuration on serial and POS interfaces that have PPP encapsulation enabled. The commands in this task apply to all authentication types supported by PPP (CHAP, MS-CHAP, and PAP).

Before you begin

You must enable PPP encapsulation on the interface with the **encapsulation ppp** command.

- To enable PPP encapsulation on a POS interface, see the Configuring POS Interfaces, on page 327 module in this manual.
- To enable PPP encapsulation on an interface, see the Configuring Serial Interfaces, on page 371 module in this manual.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp max-bad-auth** *retries*
4. **ppp max-configure** *retries*
5. **ppp max-failure** *retries*
6. **ppp max-terminate** *number*
7. **ppp timeout authentication** *seconds*
8. **ppp timeout retry** *seconds*
9. **end** or **commit**
10. **show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp max-bad-auth** *retries* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp max-bad-auth 3`	(Optional) Configures the number of authentication retries allowed on an interface after a PPP authentication failure. • If you do not specify the number of authentication retries allowed, the router resets itself immediately after an authentication failure. • Replace the *retries* argument with number of retries after which the interface is to reset itself, in the range from 0 through 10. • The default is 0 retries. • The **ppp max-bad-auth** command applies to any interface on which PPP encapsulation is enabled.
Step 4	**ppp max-configure** *retries* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp max-configure 4`	(Optional) Specifies the maximum number of configure requests to attempt (without response) before the requests are stopped. • Replace the *retries* argument with the maximum number of configure requests retries, in the range from 4 through 20. • The default maximum number of configure requests is 10. • If a configure request message receives a reply before the maximum number of configure requests are sent, further configure requests are abandoned.
Step 5	**ppp max-failure** *retries* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp max-failure 3`	(Optional) Configures the maximum number of consecutive Configure Negative Acknowledgments (CONFNAKs) permitted before a negotiation is terminated. • Replace the *retries* argument with the maximum number of CONFNAKs to permit before terminating a negotiation, in the range from 2 through 10. • The default maximum number of CONFNAKs is 5.

	Command or Action	Purpose
Step 6	**ppp max-terminate** *number* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp max-terminate 5`	(Optional) Configures the maximum number of terminate requests (TermReqs) to send without reply before the Link Control Protocol (LCP) or Network Control Protocol (NCP) is closed. • Replace the *number* argument with the maximum number of TermReqs to send without reply before closing down the LCP or NCP. Range is from 2 to 10. • The default maximum number of TermReqs is 2.
Step 7	**ppp timeout authentication** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp timeout authentication 20`	(Optional) Sets PPP authentication timeout parameters. • Replace the *seconds* argument with the maximum time, in seconds, to wait for a response to an authentication packet. Range is from 3 to 30 seconds. • The default authentication time is 10 seconds, which should allow time for a remote router to authenticate and authorize the connection and provide a response. However, it is also possible that it will take much less time than 10 seconds. In such cases, use the **ppp timeout authentication** command to lower the timeout period to improve connection times in the event that an authentication response is lost.
Step 8	**ppp timeout retry** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp timeout retry 8`	(Optional) Sets PPP timeout retry parameters. • Replace the *seconds* argument with the maximum time, in seconds, to wait for a response during PPP negotiation. Range is from 1 to 10 seconds. • The default is 3 seconds.

Step 9	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

	Command or Action	Purpose
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 10	**show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*} **Example:** `RP/0/RP0/CPU0:router# show ppp interfaces serial 0/2/0/0`	Verifies the PPP configuration for an interface or for all interfaces that have PPP encapsulation enabled.

Configuring PPP Authentication

This section contains the following procedures:

Enabling PAP, CHAP, and MS-CHAP Authentication

This task explains how to enable PAP, CHAP, and MS-CHAP authentication on a serial or POS interface.

Before you begin

You must enable PPP encapsulation on the interface with the **encapsulation ppp** command, as described in the following modules:

- To enable PPP encapsulation on a POS interface, see the Configuring POS Interfaces, on page 327 module in this manual.
- To enable PPP encapsulation on an interface, see the Configuring Serial Interfaces, on page 371 module in this manual.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp authentication** *protocol* [*protocol* [*protocol*]] [*list-name* | **default**]
4. **end** or **commit**
5. **show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp authentication** *protocol* [*protocol* [*protocol*]] [*list-name* | **default**] **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp authentication chap pap MIS-access`	Enables CHAP, MS-CHAP, or PAP on an interface, and specifies the order in which CHAP, MS-CHAP, and PAP authentication is selected on the interface. • Replace the *protocol* argument with **pap**, **chap**, **or ms-chap**. • Replace the *list name* argument with the name of a list of methods of authentication to use. To create a list, use the **aaa authentication ppp** command, as described in the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Command Reference*. • If no list name is specified, the system uses the default. The default list is designated with the **aaa authentication ppp** command, as described in the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Command Reference*.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

	Command or Action	Purpose
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*} **Example:** `RP/0/RP0/CPU0:router# show ppp interfaces serial 0/2/0/0`	Displays PPP state information for an interface. • Enter the *type interface-path-id* argument to display PPP information for a specific interface. • Enter the **brief** keyword to display brief output for all interfaces on the router, for a specific interface instance, or for all interfaces on a specific node. • Enter the **all** keyword to display detailed PPP information for all nodes installed in the router. • Enter the **location** *node-id* keyword argument to display detailed PPP information for the designated node. There are seven possible PPP states applicable for either the Link Control Protocol (LCP) or the Network Control Protocol (NCP).

What to do next

Configure a PAP, CHAP, or MS-CHAP authentication password, as described in the appropriate section:

- If you enabled PAP on an interface, configure a PAP authentication username and password, as described in the “Configuring a PAP Authentication Password” section on page 641.
- If you enabled CHAP on an interface, configure a CHAP authentication password, as described in the “Configuring a CHAP Authentication Password” section on page 643
- If you enabled MS-CHAP on an interface, configure an MS-CHAP authentication password, as described in the “Configuring an MS-CHAP Authentication Password” section on page 645

Configuring a PAP Authentication Password

This task explains how to enable and configure PAP authentication on a serial or POS interface.

Note PAP is the least secure authentication protocol available on POS and interfaces. To ensure higher security for information that is sent over POS and interfaces, we recommend configuring CHAP or MS-CHAP authentication in addition to PAP authentication.

Before you begin

You must enable PAP authentication on the interface with the **ppp authentication** command, as described in the Enabling PAP, CHAP, and MS-CHAP Authentication.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp pap sent-username** *username* **password** [**clear** | **encrypted**] *password*
4. **end** or **commit**
5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp pap sent-username** *username* **password** [**clear** | **encrypted**] *password* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp pap sent-username xxxx password notified`	Enables remote Password Authentication Protocol (PAP) support for an interface, and includes the **sent-username** and **password** commands in the PAP authentication request packet to the peer. • Replace the *username* argument with the username sent in the PAP authentication request. • Enter **password clear** to select cleartext encryption for the password, or enter **password encrypted** if the password is already encrypted. • The **ppp pap sent-username** command allows you to replace several username and password configuration commands with a single copy of this command on interfaces. • You must configure the **ppp pap sent-username** command for each interface. • Remote PAP support is disabled by default.
Step 4	**end** or **commit** **Example:**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		`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.

Step 5	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

Configuring a CHAP Authentication Password

This task explains how to enable CHAP authentication and configure a CHAP password on a serial or POS interface.

Before you begin

You must enable CHAP authentication on the interface with the **ppp authentication** command, as described in the Enabling PAP, CHAP, and MS-CHAP Authentication.

Restrictions

The same CHAP password must be configured on both host endpoints.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp chap password** [**clear** | **encrypted**] *password*
4. **end** or **commit**
5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp chap password** [**clear** | **encrypted**] *password* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp chap password clear xxxx`	Enables CHAP authentication on the specified interface, and defines an interface-specific CHAP password. • Enter **clear** to select cleartext encryption, or **encrypted** if the password is already encrypted. • Replace the *password* argument with a cleartext or already-encrypted password. This password is used to authenticate secure communications among a collection of routers. • The **ppp chap password** command is used for remote CHAP authentication only (when routers authenticate to the peer) and does not effect local CHAP authentication.This command is useful when you are trying to authenticate a peer that does not support this command (such as a router running an older Cisco IOS XR software image). • The CHAP secret password is used by the routers in response to challenges from an unknown peer.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

	Command or Action	Purpose
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

Configuring an MS-CHAP Authentication Password

This task explains how to enable MS-CHAP authentication and configure an MS-CHAP password on a serial or POS interface.

Before you begin

You must enable MS-CHAP authentication on the interface with the **ppp authentication** command, as described in the Enabling PAP, CHAP, and MS-CHAP Authentication.

Restrictions

The same MS-CHAP password must be configured on both host endpoints.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp ms-chap password** [**clear** | **encrypted**] *password*
4. **end** or **commit**
5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.

	Command or Action	Purpose
Step 3	**ppp ms-chap password** [**clear** | **encrypted**] *password* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp ms-chap password clear xxxx`	Enables a router calling a collection of routers to configure a common Microsoft Challenge Handshake Authentication (MS-CHAP) secret password. The MS-CHAP secret password is used by the routers in response to challenges from an unknown peer.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

Disabling an Authentication Protocol

This section contains the following procedures:

Disabling PAP Authentication on an Interface

This task explains how to disable PAP authentication on a serial or POS interface.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp pap refuse**
4. **end** or **commit**

5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp pap refuse** **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp pap refuse`	Refuses Password Authentication Protocol (PAP) authentication from peers requesting it. • If outbound Challenge Handshake Authentication Protocol (CHAP) has been configured (using the **ppp authentication** command), CHAP will be suggested as the authentication method in the refusal packet. • PAP authentication is disabled by default.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**show running-config** **Example:**	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show running-config`	

Disabling CHAP Authentication on an Interface

This task explains how to disable CHAP authentication on a serial or POS interface.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp chap refuse**
4. **end** or **commit**
5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp chap refuse** **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp chap refuse`	Refuses CHAP authentication from peers requesting it. After you enter the **ppp chap refuse** command under the specified interface, all attempts by the peer to force the user to authenticate with the help of CHAP are refused. • CHAP authentication is disabled by default. • If outbound Password Authentication Protocol (PAP) has been configured (using the **ppp authentication** command), PAP will be suggested as the authentication method in the refusal packet.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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]:`

	Command or Action	Purpose
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

Disabling MS-CHAP Authentication on an Interface

This task explains how to disable MS-CHAP authentication on a serial or POS interface.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp ms-chap refuse**
4. **end** or **commit**
5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp ms-chap refuse** **Example:**	Refuses MS-CHAP authentication from peers requesting it. After you enter the **ppp ms-chap refuse** command under

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# ppp ms-chap refuse`	the specified interface, all attempts by the peer to force the user to authenticate with the help of MS-CHAP are refused. • MS-CHAP authentication is disabled by default. • If outbound Password Authentication Protocol (PAP) has been configured (using the **ppp authentication** command), PAP will be suggested as the authentication method in the refusal packet.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

Configuration Examples for PPP

This section provides the following configuration examples:

Configuring a POS Interface with PPP Encapsulation: Example

The following example shows how to create and configure a POS interface with PPP encapsulation:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
```

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# ppp pap sent-username P1_TEST-8 password xxxx
RP/0/RP0/CPU0:router(config-if)# ppp authentication chap pap MIS-access
RP/0/RP0/CPU0:router(config-if)# ppp chap password encrypted xxxx
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

The following example shows how to configure POS interface 0/3/0/1 to allow two additional retries after an initial authentication failure (for a total of three failed authentication attempts):

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/1
RP/0/RP0/CPU0:router(config-if)# ppp max-bad-auth 3
```

Configuring a Serial Interface with PPP Encapsulation: Example

The following example shows how to create and configure a serial interface with PPP MS-CHAP encapsulation:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# ppp authentication ms-chap MIS-access
RP/0/RP0/CPU0:router(config-if)# ppp ms-chap password encrypted xxxx
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

CHAPTER 17

Configuring 802.1Q VLAN Interfaces

This module describes the configuration and management of 802.1Q VLAN interfaces.

The IEEE 802.1Q specification establishes a standard method for tagging Ethernet frames with VLAN membership information, and defines the operation of VLAN bridges that permit the definition, operation, and administration of VLAN topologies within a bridged LAN infrastructure.

The 802.1Q standard is intended to address the problem of how to divide large networks into smaller parts so broadcast and multicast traffic does not use more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

Feature History for Configuring 802.1Q VLAN Interfaces

Release	Modification
Release 3.2	This feature was introduced on the Cisco CRS-1 Router.
Release 3.3.0	• Support was added for VLAN commands on bundled Ethernet interfaces. • Support was added for the **dot1q native vlan** command on Cisco CRS-1 Router shared port adapters (SPAs).
Release 3.4.0	• The Layer 2 Virtual Private Network (L2VPN) feature was first supported on Ethernet interfaces on the Cisco CRS-1 Router. • Support was added on for the 8-Port 1-Gigabit Ethernet SPA.

Prerequisites for Configuring 802.1Q VLAN Interfaces

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 802.1Q VLAN interfaces, be sure that the following conditions are met:

- You must have configured a Gigabit Ethernet interface, a 10-Gigabit Ethernet interface, a Fast Ethernet interface, or an Ethernet Bundle.

Information About Configuring 802.1Q VLAN Interfaces

To configure 802.1Q VLAN interfaces, you must understand the following concepts:

802.1Q VLAN Overview

A VLAN is a group of devices on one or more LANs that are configured so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are very flexible for user and host management, bandwidth allocation, and resource optimization.

The IEEE 802.1Q protocol standard addresses the problem of dividing large networks into smaller parts so broadcast and multicast traffic does not consume more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.

Cisco IOS XR software supports VLAN subinterface configuration on Gigabit Ethernet, 10-Gigabit Ethernet, and Fast Ethernet interfaces.

802.1Q Tagged Frames

The IEEE 802.1Q tag-based VLAN uses an extra tag in the MAC header to identify the VLAN membership of a frame across bridges. This tag is used for VLAN and quality of service (QoS) priority identification. The VLANs can be created statically by manual entry or dynamically through Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP). The VLAN ID associates a frame with a specific VLAN and provides the information that switches must process the frame across the network. A tagged frame is four bytes longer than an untagged frame and contains two bytes of Tag Protocol Identifier (TPID) residing within the type and length field of the Ethernet frame and two bytes of Tag Control Information (TCI) which starts after the source address field of the Ethernet frame.

CFM on 802.1Q VLAN Interfaces

Configuring Connectivity Fault Management (CFM) for monitoring 802.1Q VLAN interfaces is identical to configuring CFM for monitoring Ethernet interfaces.

For information on configuring CFM for Ethernet interfaces, refer to the following sections in the Configuring Ethernet OAM, on page 119 module:

Subinterfaces

Subinterfaces are logical interfaces created on a hardware interface. These software-defined interfaces allow for segregation of traffic into separate logical channels on a single hardware interface as well as allowing for better utilization of the available bandwidth on the physical interface.

Subinterfaces are distinguished from one another by adding an extension on the end of the interface name and designation. For instance, the Ethernet subinterface 23 on the physical interface designated TenGigE 0/1/0/0 would be indicated by TenGigE 0/1/0/0.23.

Before a subinterface is allowed to pass traffic it must have a valid tagging protocol encapsulation and VLAN identifier assigned. All Ethernet subinterfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined.

Subinterface MTU

The subinterface maximum transmission unit (MTU) is inherited from the physical interface with an additional four bytes allowed for the 802.1Q VLAN tag.

Native VLAN

Each physical port may have a native VLAN assigned. All untagged frames are assigned to the LAN specified in the PVID parameter. When received packet is tagged with the PVID, that packet is treated as if it was untagged. Therefore, the configuration associated with the native VLAN must be placed on the main interface. The native VLAN allows the coexistence of VLAN-aware bridge or stations with VLAN-unaware bridges or stations.

VLAN Subinterfaces on Ethernet Bundles

An Ethernet bundle is a group of one or more Ethernet ports that are aggregated together and treated as a single link. Multiple VLAN subinterfaces can be added to a single Ethernet bundle.

The procedure for creating VLAN subinterfaces on an Ethernet bundle is exactly the same as the procedure for creating VLAN subinterfaces on a physical Ethernet interface.

To create a VLAN subinterface on an Ethernet bundle, see the How to Configure 802.1Q VLAN Interfaces section later in this module.

Layer 2 VPN on VLANs

The Layer 2 Virtual Private Network (L2VPN) feature enables Service Providers (SPs) to provide layer 2 services to geographically disparate customer sites.

The configuration model for configuring VLAN attachment circuits (ACs) is similar to the model used for configuring basic VLANs, where the user first creates a VLAN subinterface, and then configures that VLAN in subinterface configuration mode. To create an AC, you need to include the **l2transport** keyword in the **interface** command string to specify that the interface is a Layer 2 interface.

VLAN ACs support three modes of L2VPN operation:

- Basic Dot1Q AC—The AC covers all frames that are received and sent with a specific VLAN tag.

- Q-in-Q AC—The AC covers all frames received and sent with a specific outer VLAN tag and a specific inner VLAN tag. Q-in-Q is an extension to Dot1Q that uses a stack of two tags.
- Q-in-Any AC—The AC covers all frames received and sent with a specific outer VLAN tag and any inner VLAN tag, as long as that inner VLAN tag is not L3 terminated. Q-in-Any is an extension to Q-in-Q that uses wildcarding to match any second tag.

Note Cisco CRS Router does not support Q-in-Q and Dot1Q with the same outer VLAN ID on the same interface/port.

Note The Q-in-Any mode is a variation of the basic Dot1Q mode. In Q-in-Any mode, the frames have a basic Q-in-Q encapsulation; however, in Q-in-Any mode the inner tag is not relevant, except for the fact that a few specific inner VLAN tags are siphoned for specific services. For example, a tag may be used to provide L3 services for general internet access.

Each VLAN on a CE-to-PE link can be configured as a separate L2VPN connection (using either VC type 4 or VC type 5). To configure L2VPN on VLANs, see the Configuring an Attachment Circuit on a VLAN section.

Keep the following in mind when configuring L2VPN on a VLAN:

- Cisco IOS XR software supports 4k ACs per LC.
- In a point-to-point connection, the two ACs do not have to be of the same type. For example, a port mode Ethernet AC can be connected to a Dot1Q Ethernet AC.
- Pseudo-wires can run in VLAN mode or in port mode. A pseudo-wire running in VLAN mode has a single Dot1Q tag, while a pseudo-wire running in port mode has no tags. Some interworking is required to connect these different types of circuits together. This interworking takes the form of popping, pushing and rewriting tags. The advantage of Layer 2 VPN is that is simplifies the interworking required to connect completely different media types together.
- The ACs on either side of an MPLS pseudo-wire can be different types. In this case, the appropriate conversion is carried out at one or both ends of the AC to pseudo-wire connection.

Use the **show interfaces** command to display AC and pseudo-wire information.

Note For detailed information about configuring an L2VPN network, see the module of the *Multiprotocol Label Switching Configuration Guide.*

How to Configure 802.1Q VLAN Interfaces

This section contains the following procedures:

Configuring 802.1Q VLAN Subinterfaces

This task explains how to configure 802.1Q VLAN subinterfaces. To remove these subinterfaces, see the "Removing an 802.1Q VLAN Subinterface" section.

SUMMARY STEPS

1. **configure**
2. **interface {GigabitEthernet | TenGigE | Bundle-Ether}** *interface-path-id.subinterface*
3. **encapsulation dot1q**
4. **ipv4 address** *ip-address mask*
5. **exit**
6. Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.
7. **end** or **commit**
8. **show vlan interface** [*type interface-path-id*][**location** *instance*]
9. **show vlan trunks** [**brief**] [**location** *instance*] [{**GigabitEthernet | TenGigE | Bundle-Ether |** } *interface-path-id*] [**summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface {GigabitEthernet | TenGigE | Bundle-Ether}** *interface-path-id.subinterface* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10`	Enters subinterface configuration mode and specifies the interface type, location, and subinterface number. • Replace the *interface-path-id* argument with one of the following instances: • Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is *rack/slot/module/port*, and a slash between values is required as part of the notation. • Ethernet bundle instance. Range is from 1 through 65535. • Replace the *subinterface* argument with the subinterface value. Range is from 0 through 4095. • Naming notation is *interface-path-id.subinterface*, and a period between arguments is required as part of the notation.
Step 3	**encapsulation dot1q** **Example:**	Sets the Layer 2 encapsulation of an interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged`	**Note** • The **dot1q vlan** command is replaced by the **encapsulation dot1q** command on the Cisco ASR 9000 Series Router. It is still available for backward-compatibility, but only for Layer 3 interfaces.
Step 4	**ipv4 address** *ip-address mask* **Example:** `RP/0/RP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24`	Assigns an IP address and subnet mask to the subinterface. • Replace *ip-address* with the primary IPv4 address for an interface. • Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways: • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	(Optional) Exits the subinterface configuration mode. • The **exit** command is not explicitly required.
Step 6	Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.	—

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 8	**show vlan interface** [*type interface-path-id*][**location** *instance*] **Example:** `RP/0/RP0/CPU0:router# show vlan interface 5`	(Optional) Displays the interface configuration. • To display the configuration for a particular port, use the **location** keyword. • To display the configuration for the specified interface or subinterface, use the **interface** keyword.
Step 9	**show vlan trunks** [**brief**] [**location** *instance*] [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether** | } *interface-path-id*] [**summary**] **Example:** `RP/0/RP0/CPU0:router# show vlan trunk summary`	(Optional) Displays summary information about each of the VLAN trunk interfaces. • The keywords have the following meanings: • **brief**—Displays a brief summary. • **summary**—Displays a full summary. • **location**—Displays information about the VLAN trunk interface on the given port. • **interface**—Displays information about the specified interface or subinterface.

Configuring Native VLAN

This task explains how to configure the native, or default, VLAN on an interface.

SUMMARY STEPS

1. **configure**
2. **interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id*
3. **dot1q native vlan** *number*
4. **end** or **commit**
5. **show ethernet trunk bundle-ether** *instance* **show vlan trunks** [**brief**] [**location** *instance*] [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id*] [**summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4`	Enters interface configuration mode and specifies the Ethernet interface name and designation. Replace the *interface-path-id* argument with one of the following instances: • Physical Ethernet interface instance or with an Ethernet bundle instance. Naming notation is *rack/slot/module/port*, and a slash between values is required as part of the notation. • Ethernet bundle instance. Range is from 1 through 65535.
Step 3	**dot1q native vlan** *number* **Example:** `RP/0/RP0/CPU0:router(config-if)# dot1q native vlan 1`	Defines the default, or Native VLAN, associated with an 802.1Q trunk interface. • The number argument is the ID of the trunk interface. • Range is from 1 through 4094 inclusive (0 and 4095 are reserved).

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# 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.

Step 5	**show ethernet trunk bundle-ether** *instance* **show vlan trunks** [**brief**] [**location** *instance*] [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id*] [**summary**] **Example:**	(Optional) Displays summary information about each of the interface configuration.VLAN trunk interfaces. The Ethernet bundle instance range is from 1 through 65535. • The keywords have the following meanings:

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show ethernet vlan trunk bundle-ether 5summary`	• **brief**—Displays a brief summary. • **summary**—Displays a full summary. • **location**— Displays information about the VLAN trunk interface on the given port. • **interface**—Displays information about the specified interface or subinterface.

Configuring an Attachment Circuit on a VLAN

Use the following procedure to configure an attachment circuit on a VLAN.

SUMMARY STEPS

1. **configure**
2. **interface [GigabitEthernet | TenGigE | Bundle-Ether | TenGigE]** *interface-path*] *id.subinterface* **l2transport**
3. **dot1q vlan** *vlan-id* **encapsulation dot1q**
4. **l2protocol** {**cdp** | **pvst** | **stp** | **vtp**} {[**forward** | **tunnel**][**experimental** *bits*]|**drop**}
5. **end** or **commit**
6. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 2	**interface [GigabitEthernet | TenGigE | Bundle-Ether | TenGigE]** *interface-path*] *id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0.1 l2transport`	Enters subinterface configuration and specifies the interface type, location, and subinterface number. • Replace the *interface-path-id* argument with one of the following instances: • Physical Ethernet interface instance or Ethernet bundle instance. Naming notation is *rack*/*slot*/*module*/*port*, and a slash between values is required as part of the notation. • Ethernet bundle instance. Range is from 1 through 65535. • Replace the *subinterface* argument with the subinterface value. Range is from 0 through 4095.

	Command or Action	Purpose
		• Naming notation is *instance.subinterface*, and a period between arguments is required as part of the notation. • You must include the **l2transport** keyword in the command string; otherwise, the configuration creates a Layer 3 subinterface rather that an AC.
Step 3	**dot1q vlan** *vlan-id* **encapsulation dot1q** **Example:** `RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 10 vlan any`	Assigns a VLAN AC to the subinterface. • Replace the vlan-id argument with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved). To configure a basic Dot1Q AC, use the following syntax: **dot1q vlan** *vlan-id* • To configure a Q-in-Q AC, use the following syntax: **dot1q vlan** *vlan-id* **vlan** *vlan-id* • To configure a Q-in-Any AC, use the following syntax: **dot1q vlan** *vlan-id* **vlan any**
Step 4	**l2protocol** {**cdp** | **pvst** | **stp** | **vtp**}{[**forward** | **tunnel**][**experimental** *bits*]|**drop**} **Example:** `RP/0/RP0/CPU0:router(config-if-l2)# l2protocol stp tunnel`	Configures Layer 2 protocol tunneling and protocol data unit (PDU) filtering on an interface. Possible protocols and options are: • **cdp**—Cisco Discovery Protocol (CDP) tunneling and data unit parameters. • **pvst**—Configures VLAN spanning tree protocol tunneling and data unit parameters. • **stp**—spanning tree protocol tunneling and data unit parameters. • **vtp**—VLAN trunk protocol tunneling and data unit parameters. • **tunnel**—(Optional) Tunnels the packets associated with the specified protocol. • **experimental** *bits*—(Optional) Modifies the MPLS experimental bits for the specified protocol. • **drop**—(Optional) Drop packets associated with the specified protocol.
Step 5	**end** or **commit** **Example:**	Saves configuration changes.

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-if-l2)# end`
	or
	`RP/0/RP0/CPU0:router(config-if-l2)# 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.

Step 6	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface* **Example:** `RP/0/RP0/CPU0:router# show interfaces TenGigE 0/3/0/0.1`	(Optional) Displays statistics for interfaces on the router.

What to do next

- To configure a point-to-point pseudowire cross connect on the AC, see the "Implementing MPLS Layer 2 VPNs" VPNs module of the Cisco IOS XR Multiprotocol Label Switching Configuration Guide.
- To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or Quality of Service (QoS), to the VLAN, refer to the appropriate Cisco IOS XR software configuration guide.

Removing an 802.1Q VLAN Subinterface

This task explains how to remove 802.1Q VLAN subinterfaces that have been previously configured using the Configuring 802.1Q VLAN subinterfaces section in this module.

SUMMARY STEPS

1. **configure**
2. **no interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *interface-path-id.subinterface*
3. Repeat Step 2 to remove other VLAN subinterfaces.
4. **end** or **commit**

5. **show vlan trunks** [**brief**] [**location** *instance*] [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether** | **fastethernet**} *interface-path-id*] [**summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `/CPU0:router# configure`	Enters global configuration mode.
Step 2	**no interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *interface-path-id.subinterface* **Example:** `RP/0/RP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10`	Removes the subinterface, which also automatically deletes all the configuration applied to the subinterface. • Replace the *instance* argument with one of the following instances: • Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is *rack/slot/module/port*, and a slash between values is required as part of the notation. • Ethernet bundle instance. Range is from 1 through 65535. • Replace the *subinterface* argument with the subinterface value. Range is from 0 through 4095. Naming notation is *instance.subinterface*, and a period between arguments is required as part of the notation.
Step 3	Repeat Step 2 to remove other VLAN subinterfaces.	—

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show vlan trunks** [**brief**] [**location** *instance*] [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether** | **fastethernet**} *interface-path-id*] [**summary**] **Example:** `RP/0/RP0/CPU0:router# show vlan trunk summary`	(Optional) Displays summary information about each of the VLAN trunk interfaces. • The keywords have the following meanings: • **brief**—Displays a brief summary. • **summary**—Displays a full summary. • **location**—Displays information about the VLAN trunk interface on the given port. • **interface**—Displays information about the specified interface or subinterface.

Configuration Examples for VLAN Interfaces

This section contains the following example:

VLAN Subinterfaces: Example

The following example shows how to create three VLAN subinterfaces at one time:

```
RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface TenGigE 0/2/0/4.1
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 10
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 10.0.10.1/24
RP/0/RP00/CPU0:router(config-subif)# interface TenGigE0/2/0/4.2
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 101
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 10.0.20.1/24
RP/0/RP00/CPU0:router(config-subif)# interface TenGigE0/2/0/4.3
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 30
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 102
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 10.0.30.1/24
RP/0/RP00/CPU0:router(config-subif)# commit
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# exit

RP/0/RP00/CPU0:router# show vlan trunks summary
Trunk                                 Sub types        Sub states
VLAN trunks: 1,
  1 are 802.1Q (Ether)
Sub-interfaces: 3,
```

```
  3 are up.
802.1Q VLANs: 3,
  3 have VLAN Ids,

RP/0/RP0/CPU0:router# show vlan interfaceinterface          encapsulation  vlan-id
intf-state
Te0/2/0/4.1         802.1Q             10  up
Te0/2/0/4.2         802.1Q             20  up
Te0/2/0/4.3         802.1Q             30  up
RP/0/RP0/CPU0:router# show vlan trunks briefinterface           encapsulations
  intf-state

Summary                       1000        0    1000     1000       0       0

Te0/2/0/4           802.1Q (Ether)        up
```

The following example shows how to create two VLAN subinterfaces on an Ethernet bundle:

```
RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface bundle-ether 2
RP/0/RP00/CPU0:router(config-if)# ipv4 address 192.168.2.1/24
RP/0/RP00/CPU0:router(config-if)# exit
RP/0/RP00/CPU0:router(config)# interface bundle-ether 2.1
RP/0/RP0/CPU0:router(config-subif)#  dot1q vlan 10
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 192.168.100.1/24
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# interface bundle-ether 2.2
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 200
RP/0/RP00/CPU0:router(config-subif)# ipv4 address 192.168.200.1/24
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# commit
```

The following example shows how to create a basic dot1Q AC:

```
RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.1
RP/0/RP00/CPU0:router(config-subif)# l2transport
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP00/CPU0:router(config-subif)# commit
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# exit
```

The following example shows how to create a Q-in-Q AC:

```
RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.2
RP/0/RP00/CPU0:router(config-subif)# l2transport
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20 vlan 10
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 200 second-dot1q 201
```

```
RP/0/RP00/CPU0:router(config-subif)# commit
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# exit
```

The following example shows how to create a Q-in-Any AC:

```
RP/0/RP00/CPU0:router# configure
RP/0/RP00/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.3
RP/0/RP00/CPU0:router(config-subif)# l2transport
RP/0/0/CPU0:router(config-subif)# dot1q vlan 30 vlan any
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 300 second-dot1q any
RP/0/RP00/CPU0:router(config-subif)# commit
RP/0/RP00/CPU0:router(config-subif)# exit
RP/0/RP00/CPU0:router(config)# exit
```

CHAPTER 18

Configuring Tunnel Interfaces

This module describes the configuration of Tunnel-IPSec interfaces on the Cisco CRS Router.

Tunnel interfaces are virtual interfaces that provide encapsulation of arbitrary packets within another transport protocol. The Tunnel-IPSec interface provides secure communications over otherwise unprotected public routes.

A virtual interface represents a logical packet switching entity within the router. Virtual Interfaces have a global scope and do not have an associated location. The Cisco IOS XR Software uses the rack/slot/module/port notation for identifying physical interfaces, but uses a globally unique numerical ID after the interface name to identify virtual interfaces. Examples of this numerical ID are Loopback 0, Loopback 1, and Null99999. The ID is unique for each virtual interface type so you may simultaneously have a Loopback 0 and a Null 0.

Virtual interfaces have their control plane presence on the active route processor (RP). The configuration and control plane are mirrored onto the standby RP and, in the event of a switchover, the virtual interfaces will move to the standby, which then becomes the newly active RP.

Note Subinterfaces can be physical or virtual, depending on their parent interface.

Virtual tunnels are configured on any RP or distributed RP (DRP), but they are created and operate only from the RP.

Note Tunnels do not have a one-to-one modular services card association.

Feature History for Configuring Tunnel Interfaces on Cisco IOS XR Software

Release	Modification
Release 2.0	This feature was introduced on the Cisco CRS Router.

Prerequisites for Configuring Tunnel Interfaces

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.

Information About Configuring Tunnel Interfaces

To implement tunnel interfaces, you must understand the following concepts:

Tunnel Interfaces Overview

Tunneling provides a way to encapsulate arbitrary packets inside of a transport protocol. This feature is implemented as a virtual interface to provide a simple interface for configuration. The tunnel interfaces are not tied to specific “passenger” or “transport” protocols, but, rather, they represent an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. Because supported tunnels are point-to-point links, you must configure a separate tunnel for each link.

There are three necessary steps in configuring a tunnel interface:

1. Specify the tunnel interface—**interface tunnel-ipsec***identifier*.
2. Configure the tunnel source—**tunnel source** {*ip-address* | *interface-id*}.
3. Configure the tunnel destination—**tunnel destination** {*ip-address* | *tunnel-id*}.

Virtual Interface Naming Convention

Virtual interface names never use the physical interface naming notation *rack/slot/module/port* for identifying an interface’s rack, slot, module, and port, because they are not tied to any physical interface or subinterface.

Virtual interfaces use a globally unique numerical identifier (per virtual interface type).

Examples of naming notation for virtual interfaces:

```
Interface                      IP-Address      Status  Protocol
Loopback0                      10.9.0.0         Up           Up
Loopback10                     10.7.0.0         Up           Up
Tunnel-TE5000                  172.18.189.38    Down         Down
Null10                         10.8.0.0          Up           Up
```

Tunnel-IPSec Overview

IPSec (IP security) is a framework of open standards for ensuring secure private communications over the Internet. It can be used to support Virtual Private Network (VPN), firewalls, and other applications that must transfer data across a public or insecure network. The router IPSec protocol suite provides a set of standards that are used to provide privacy, integrity, and authentication service at the IP layer. The IPSec protocol suite also includes cryptographic techniques to support the key management requirements of the network-layer security.

When IPSec is used, there is no need to use Secure Shell (SSH) or Secure Socket Layer (SSL). Their use causes the same data to be encrypted or decrypted twice, which creates unnecessary overhead. The IPSec daemon is running on both the RPs and the DRPs. IPSec is an optional feature on the router. IPSec is a good choice for a user who has multiple applications that require secure transport. On the client side, customers can use "Cisco VPN 3000 Client" or any other third-party IPSec client software to build IPSec VPN.

Note IPSec tunnel exists in the control plane, so you do not have to bring up or bring down the tunnel. Entry into the IPSec tunnel is only for locally sourced traffic from the RP or DRP, and is dictated by the access control lists (ACL) configured as a part of the profile that is applied to the Tunnel-IPSec.

Tunnel-IPSec Naming Convention

A profile is entered from interface configuration submode for interface tunnel-ipsec. For example:

```
interface tunnel-ipsec 30
     profile <profile name>
```

Crypto Profile Sets

Crypto profile sets must be configured and applied to tunnel interfaces (or to the crypto IPSec transport). For details on using the crypto IPSec transport, refer to the link provided in the Where to Go Next. For IPSec to succeed between two IPSec peers, the crypto profile entries of both peers must contain compatible configuration statements.

Two peers that try to establish a security association must each have at least one crypto profile entry that is compatible with one of the other peer's crypto profile entries. For two crypto profile entries to be compatible, they must at least meet the following criteria:

- They must contain compatible crypto access lists. In the case where the responding peer is using dynamic crypto profiles, the entries in the local crypto access list must be "permitted" by the peer's crypto access list.
- They must each identify the other peer (unless the responding peer is using dynamic crypto profiles).
- They must have at least one transform set in common.

Note Crypto profiles cannot be shared; that is, the same profile cannot be attached to multiple interfaces.

How to Configure Tunnel Interfaces

This section contains the following procedures:

Configuring Tunnel-IPSec Interfaces

This task explains how to configure Tunnel-IPSec interfaces.

Before you begin

To use the profile command, you must be in a user group associated with a task group that includes the proper task IDs for crypto commands. To use the **tunnel destination** command, you must be in a user group associated with a task group that includes the proper task IDs for interface commands.

For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of *Cisco IOS XR System Security Configuration Guide*.

The following tasks are required for creating Tunnel-IPSec interfaces:

- Setting Global Lifetimes for IPSec Security Associations
- Configuring Checkpointing
- Configuring Crypto Profiles

  For detailed information on configuring the prerequisite checkpointing and crypto profiles, and setting the global lifetimes for IPSec security associations, refer to the *Implementing IPSec Network Security on Cisco IOS XR Software* module in .

  After configuring crypto profiles, you must apply a crypto profile to each tunnel interface through which IPSec traffic will flow. Applying the crypto profile set to a tunnel interface instructs the router to evaluate all the interface's traffic against the crypto profile set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

SUMMARY STEPS

1. **configure**
2. **end** or **commit**
3. **configure**
4. **interface tunnel-ipsec** *identifier*
5. **profile** *profile-name*
6. **tunnel source** (*ip-address* | *interface-id*) RP/0/RP0/CPU0:router(config-if)# tunnel source Ethernet0/1/1/2
7. **tunnel destination** {*ip-address* | *tunnel-id*}
8. **end** or **commit**
9. **show ip route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# 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.

	Command or Action	Purpose
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 3	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 4	**interface tunnel-ipsec** *identifier* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-ipsec 30`	Identifies the IPSec interface to which the crypto profile will be attached and enters interface configuration mode.
Step 5	**profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# profile user1`	Assigns the crypto profile name to be applied to the tunnel for IPSec processing. • The same crypto profile cannot be shared in different IPSec modes.
Step 6	**tunnel source** (*ip-address* | *interface-id*) RP/0/RP0/CPU0:router(config-if)# tunnel source Ethernet0/1/1/2	Specifies the tunnel source IP address or interface ID. • This command is required for both static and dynamic profiles.
Step 7	**tunnel destination** {*ip-address* | *tunnel-id*} **Example:** `RP/0/RP0/CPU0:router(config-if)# tunnel destination 192.168.164.19`	(Optional) Specifies the tunnel destination IP address. • This command is not required if the profile is dynamic.

Step 8	**end** or **commit**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config)# end`
	
	or
	
	`RP/0/RP0/CPU0:router(config)# 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.

	Command or Action	Purpose
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 9	**show ip route** **Example:** `RP/0/RP0/CPU0:router# show ip route`	Displays forwarding information for the tunnel. • The command **show ip route** displays what was advertised and shows the routes for static and autoroute.

Configuration Examples for Tunnel Interfaces

This section contains the following example:

Tunnel-IPSec: Example

This example shows the process of creating and applying a profile to an IPSec tunnel. The necessary preliminary steps are also shown. You must first define a transform set and then create a profile before configuring the IPSec tunnel.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ipsec transform-set tset1
RP/0/RP0/CPU0:router(config-transform-set tset1 tset1)# transform esp-sha-hmac
RP/0/RP0/CPU0:router(config-transform-set tset1)# end


Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes


RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ipsec profile user1
RP/0/RP0/CPU0:router(config-user1)# match sampleac1 transform-set tset1
RP/0/RP0/CPU0:router(config-user1)# set pfs group5
RP/0/RP0/CPU0:router(config-user1)# set type dynamic
RP/0/RP0/CPU0:router(config-user1)# exit


RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-ipsec 30
RP/0/RP0/CPU0:router(config-if)# profile user1
RP/0/RP0/CPU0:router(config-if)# tunnel source MgmtEth 0/RPO/CPU0/0
RP/0/RP0/CPU0:router(config-if)# tunnel destination 192.168.164.19
RP/0/RP0/CPU0:router(config-if)# end
```

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
```

Where to Go Next

You now must apply a crypto profile to each transport. Applying the crypto profile set to a transport instructs the router to evaluate all the interface's traffic against the crypto profile set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

For information on applying a crypto profile to each transport, see the Implementing IPSec Network Security on Cisco IOS XR Software module of the Cisco IOS XR System Security Configuration Guide.

CHAPTER 19

Configuring LAN/WAN-PHY Controllers

This module describes the configuration of LAN/WAN-PHY controllers on the Cisco CRS-1 Router.

Feature History for Configuring LAN/WAN-PHY Controller Interfaces

Release	Modification
Release 3.5.2	This feature was introduced on the Cisco CRS-1 Router. Support was added for the 1-Port 10-Gigabit Ethernet LAN/WAN-PHY Shared Port Adapter (SPA-1X10GE-WL-V2).
Release 3.6.0	No modification.
Release 3.7.0	No modification.
Release 3.8.0	No modification.
Release 3.9.0	Support was added for user configuration of WAN-PHY Signal Failure (SF) and Signal Degrade (SD) Bit Error Rate (BER) reporting and thresholds.
Release 4.0.0	Support was added for the following physical layer interface modules (PLIMs): • 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (14X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (20X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)

Release 4.0.1	Support was added for the following PLIMs: • 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP) • 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)

Prerequisites for Configuring LAN/WAN-PHY Controller Interfaces

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 a LAN/WAN-PHY controller, be sure that the following tasks and conditions are met:

- You have installed one of the following cards that supports the LAN/WAN-PHY controller:
  - 1-Port 10-Gigabit Ethernet LAN/WAN-PHY Shared Port Adapter
  - 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)
  - 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)
  - 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
  - 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM

Restrictions

The LAN/WAN-PHY controller has the following restrictions:

- LAN-PHY mode is configurable using the **lanmode on** command only on the 1-Port 10-Gigabit Ethernet LAN/WAN-PHY SPA. The default mode of the LAN/WAN-PHY controller is WAN mode for the 1-Port 10-Gigabit Ethernet SPA.
- WAN-PHY mode is configurable using the **wanmode on** command only on the 10-Gigabit Ethernet LAN/WAN-PHY PLIMs. The default mode of the LAN/WAN-PHY controller is LAN mode for the following SPAs:
  - 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)
  - 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)
  - 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
  - 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM

Information About the LAN/WAN-PHY Controllers

LAN/WAN-PHY support in Cisco IOS XR software is based on the IEEE 802.3ae standard.

WAN-PHY controllers can only be used as Path Terminating Equipment (PTE). When deploying Ethernet WAN interfaces as endpoints or PTE between routers, the other endpoint must be an Ethernet WAN interface. WAN-PHY will not interoperate and terminate on a PoS (Packet over Sonet) or EoS (Ethernet over Sonet) port. Between devices (LTE - Line Terminating Equipment or STE - Section Terminating Equipment), the endpoints can be an Add-Drop Multiplexer (ADM) or Dense Wavelength Division Multiplexing (DWDM) OC-192c POS interfaces.

The purpose of WAN-PHY is to render 10 Gigabit Ethernet compatible with the SONET STS-192c format and data rate, as defined by ANSI, as well as the SDH VC-4-64c container specified by ITU. To achieve this compatibility, a WAN Interface Sublayer (WIS) is inserted between the 10 Gigabit Ethernet Physical Coding Sublayer (PCS) and the serial Physical Medium Attachment sublayer/Physical Medium Dependent sublayer (PMA/PMD). When the controller is in WAN-PHY mode the WIS sublayer transports 10 Gigabit Ethernet frames in an OC-192c SONET payload which can interoperate with SONET section or line level repeaters. This effectively bridges the asynchronous world of Ethernet data with synchronous SONET/SDH transport allowing 10 Gigabit Ethernet to be transparently carried over current DWDM networks without having to directly map the Ethernet frames into SONET/SDH.

At a high-level, the WIS has the following characteristics and functions:

- The WIS allows WAN-PHY equipment to generate an Ethernet data stream to be mapped to an OC-192c or VC-4-64c concatenated payload at the PHY level without any MAC or higher layer processing.
- In theory, a 10GBASE-W interface in not intended to interoperate directly with SONET/SDH equipment because WAN-PHY is not fully compliant with SONET/SDH optical and electrical specifications. In practice, SONET/SDH and 10GBASE-W interfaces can interoperate.
- From a MAC perspective, WAN-PHY does not appear any different from LAN-PHY (no WIS) with the exception of the sustained data rate. In the case of LAN-PHY the data rate is 10.3125 Gbps, while at WAN-PHY it is 9.95328 Gbps (as required by SONET/SDH).

Note For information on the data rates for the SPA-1X10GE-WL-V2, refer to Table 4, *Feature and Application Comparison Between 10 Gigabit Ethernet Interfaces*, in the Cisco 1-Port 10 Gigabit Ethernet LAN/WAN-PHY Shared Port Adapter data sheet:

http://www.cisco.com/c/en/us/products/collateral/interfaces-modules/shared-port-adapters-spa-interface-processors/product_data_sheet0900aecd80715dc5.html

- The WIS implements a subset of the SONET functions including creating the Section, Line, Path Overhead headers, calculating the Bit Interleaved Parity (BIP) bytes for error monitoring and managing a variety of alarms and defect indications.

Beside the frame format and data rate 10GBASE-W and Packet over SONET (POS) have very little in common:

- POS and 10GBASE-W cannot in fact interoperate on the same link since the protocol architecture is completely different. POS is based on a serial protocol like PPP whose frames are logically and physical different from Ethernet MAC frames.
- From a Service Provider point of view, POS is a L3 point-to-point service while WAN-PHY is a L2 Ethernet hand-off. WAN-PHY should be compared more properly to an Ethernet over SONET (EoS) encapsulation technology, such as ITU-T X.86 or GFP (ITU-T G.7041) where the Ethernet frame is encapsulated respectively in an HDLC-like or GFP frame.
- POS is optically and electrically compatible with SONET/SDH protocols whereas WAN-PHY is not.
- Cisco POS supports linear Automatic Protection Switching (APS) to restore link failures in 50 msec, while WAN-PHY is not designed to support APS.
- The synchronous nature of POS requires clocking to be configured either "internal" or "line" (Internal clocking is used when the POS interface is connected to another POS in back-to-back or through DWDM, while the line clocking is required when the POS is connected to a SONET/SDH add/drop multiplexer). WAH-PHY has no requirement to support line clocking.

How to Configure LAN/WAN-PHY Controllers

The LAN/WAN-PHY controllers are configured in the physical layer control element of the Cisco IOS XR software configuration space. By default the 1-Port 10GE LAN/WAN-PHY Shared Port Adapter boots up in WAN-PHY mode. LAN-PHY mode configuration is done using the **controller wanphy** command.

Configuration of LAN/WAN-PHY controllers is described in the following tasks.

Note All interface configuration tasks for the POS or GE interfaces still must be performed in interface configuration mode. Refer to *Configuring POS Interfaces on Cisco IOS XR Software* and *Configuring Ethernet Interfaces on Cisco IOS XR Software* modules for more information.

Configuring LAN-PHY Mode

This task describes how to configure LAN-PHY mode on the 1-Port 10-Gigabit Ethernet LAN/WAN-PHY SPA, which by default operates in WAN-PHY mode.

Note After configuring LAN-PHY mode and reloading the SPA, all links are in the UP state.

Before you begin

You have the 1-Port 10-Gigabit Ethernet LAN/WAN-PHY SPA installed.

SUMMARY STEPS

1. **show controllers wanphy** *interface-path-id* [**alarms** | **all** | **registers**]
2. **configure**

3. **controller wanphy** *interface-path-id*
4. **lanmode on**
5. **end** or **commit**
6. **hw-module subslot** *interface-path-id* **reload**
7. **show controllers wanphy** *interface-path-id* [**alarms | all | registers**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show controllers wanphy** *interface-path-id* [**alarms | all | registers**]
	Example:
	`RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all`
	`Tue Jan  5 23:01:18.641 PST`
	`Interface: wanphy0_6_1_0`
	`Configuration Mode: WAN Mode`
		Displays the configuration mode of the LAN/WAN-PHY controller. By default, prior to configuration of LAN-PHY mode the controller will be in WAN-PHY mode.

Step 2	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 3	**controller wanphy** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0`	Specifies the LAN/WAN-PHY controller name in the notation *rack/slot/module/port* and enters wanphy configuration mode.

Step 4	**lanmode on**
	Example:
	`RP/0/RP0/CPU0:router(config-wanphy)# lanmode on`
	`Tue Jan  5 23:08:09.024 PST`
	`To complete the mode change the SPA must be power-cycled.`
		Configures LAN-PHY mode.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-wanphy)# end`
	or
	`RP/0/RP0/CPU0:router(config-wanphy)# 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.

	Command or Action	Purpose
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 6	**hw-module subslot** *interface-path-id* **reload** **Example:** `RP/0/RP0/CPU0:router# hw-module subslot 0/6/1 reload`	Reloads the SPA in the notation *rack/slot/module.* To complete the mode change from WAN-PHY to LAN-PHY the SPA must be power-cycled.

Step 7	**show controllers wanphy** *interface-path-id* [**alarms | all | registers**]
	Example:
	`RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all`
	`Tue Jan  5 23:28:37.738 PST`
	`Interface: wanphy0_6_1_0`
	`Configuration Mode: LAN Mode`
		Displays the configuration mode of the LAN/WAN-PHY controller.

Troubleshooting Tips

The SPA must be power-cycled to complete the controller mode change.

Examples

The following example shows how to configure LAN-PHY mode from a controller in default WAN-PHY mode for the 1-Port 10-Gigabit Ethernet LAN/WAN-PHY SPA:

```
RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all
Tue Jan 12 20:53:20.945 PST
Interface: wanphy0_6_1_0
Configuration Mode: WAN Mode
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0
RP/0/RP0/CPU0:router(config-wanphy)# lanmode on
Tue Jan 12 20:55:49.610 PST
To complete the mode change the SPA must be power-cycled.
RP/0/RP0/CPU0:router(config-wanphy)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
RP/0/RP0/CPU0:router# hw-module subslot 0/6/1 reload
RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all
Tue Jan 12 20:57:28.779 PST
Interface: wanphy0_6_1_0
```

```
Configuration Mode: LAN Mode
```

The following example shows how to configure WAN-PHY mode from a controller configured in LAN-PHY mode for the 1-Port 10-Gigabit Ethernet LAN/WAN-PHY SPA:

```
RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all
Tue Jan 12 20:03:46.483 PST
Interface: wanphy0_6_1_0
Configuration Mode: LAN Mode
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0
RP/0/RP0/CPU0:router(config-wanphy)# no lanmode on
Tue Jan 12 20:35:06.523 PST
To complete the mode change the SPA must be power-cycled.
RP/0/RP0/CPU0:router(config-wanphy)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
RP/0/RP0/CPU0:router# hw-module subslot 0/6/1 reload
RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all
Tue Jan 12 20:39:32.570 PST
Interface: wanphy0_6_1_0
Configuration Mode: WAN Mode
```

What to Do Next

All interface configuration tasks for the POS or GE interfaces still must be performed in interface configuration mode. Refer to *Configuring POS Interfaces on Cisco IOS XR Software* and *Configuring Ethernet Interfaces on Cisco IOS XR Software* modules for more information.

Configuring WAN-PHY Mode

This task describes how to configure WAN-PHY mode on the 10-Gigabit Ethernet LAN/WAN-PHY PLIMs.

Before you begin

You have one of the following 10-Gigabit Ethernet LAN/WAN-PHY PLIMs installed:

- 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)
- 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)
- 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
- 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM

SUMMARY STEPS

1. **configure**
2. **controller wanphy** *interface-path-id*
3. **wanmode on**
4. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller wanphy** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0`	Specifies the LAN/WAN-PHY controller name in the notation *rack/slot/module/port* and enters wanphy configuration mode.
Step 3	**wanmode on** **Example:** `RP/0/RP0/CPU0:router(config-wanphy)# wanmode on`	Configures WAN-PHY mode.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-wanphy)# end`
	or
	`RP/0/RP0/CPU0:router(config-wanphy)# 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.

What to do next

All interface configuration tasks for the POS or GE interfaces still must be performed in interface configuration mode. Refer to Configuring POS Interfaces on Cisco IOS XR Software and Configuring Ethernet Interfaces on Cisco IOS XR Software modules for more information.

Configuring WAN-PHY Signal Failure and Signal Degrade Bit Error Rates

This task describes how to configure WAN-PHY Signal Failure (SF) and Signal Degrade (SD) Bit Error Rate (BER) reporting and thresholds.

A Signal Failure (SF) alarm is declared if the line bit error (B2) rate exceeds a user provisioned threshold (over the range of 10e-3 to 10e-9). If the B2 errors cross the SF threshold then the link is considered unreliable and the interface changes the state to down.

A Signal Degrade (SD) alarm is declared if the line bit error (B2) rate exceeds a user provisioned threshold (over the range of 10e-3 to 10e-9). If the B2 errors cross the SD threshold then a warning of link quality degradation will occur.

The SF or SD alarm is cleared when the B2 error count remains below the configured threshold for a period of time called the Hold on Time. This table lists the configured SF and SD threshold values and the corresponding Hold on Times.

Table 14: SF and SD Configured Threshold Value Hold on Times

Configured Threshold Value	Hold on Time (max) in seconds
10e-3	13
10e-4	13
10e-5	13
10e-6	13
10e-7	20
10e-8	110
10e-9	1010

These WAN-PHY alarms are required for some users who are upgrading their Layer 2 core network from a sonet ring to a 10 Gigabit Ethernet ring.

Before you begin

The controller must be in WAN-PHY mode prior to configuring Signal Failure (SF) and Signal Degrade (SD) Bit Error Rate (BER) reporting and thresholds.

Restrictions

SF and SD BER is not supported on the following cards:

- 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)
- 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (XFP optics)

SUMMARY STEPS

1. **show controllers wanphy** *interface-path-id* [**alarms | all | registers**]
2. **configure**
3. **controller wanphy** *interface-path-id*

4. **report sd-ber**
5. **report sf-ber disable**
6. **threshold sd-ber** *range*
7. **threshold sf-ber** *range*
8. **end** or **commit**
9. **show controllers wanphy** *interface-path-id* [**alarms | all | registers**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show controllers wanphy** *interface-path-id* [**alarms | all | registers**]
	Example:
	`RP/0/RP0/CPU0:router#show controllers wanphy 0/6/1/0 all`
	`Tue Jan 19 22:32:50.591 PST`
	`Interface: wanphy0_6_1_0`
	`Configuration Mode: WAN Mode`
		Displays the configuration mode of the LAN/WAN-PHY controller.

Step 2	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 3	**controller wanphy** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0`	Specifies the LAN/WAN-PHY controller name in the notation *rack/slot/module/port* and enters wanphy configuration mode.
Step 4	**report sd-ber** **Example:** `RP/0/RP0/CPU0:router(config-wanphy)#report sd-ber`	Enables signal degrade (sd) bit error rate (ber) reporting. **Note** • By default sd-ber reporting is disabled.
Step 5	**report sf-ber disable** **Example:** `RP/0/RP0/CPU0:router(config-wanphy)#report sf-ber disable`	Disables signal fault (sf) bit error rate (ber) reporting. **Note** • By default sf-ber reporting is enabled.
Step 6	**threshold sd-ber** *range* **Example:** `RP/0/RP0/CPU0:router(config-wanphy)#threshold sd-ber 7`	Specifies the signal degrade (sd) bit error rate (ber) threshold. • Range is 3 to 9. • Range value is expressed exponentially as 10e-n. **Note** • The default sd-ber value is 6 (10e-6).

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 7</td><td>threshold sf-ber range
Example:
<code>RP/0/RP0/CPU0:router(config-wanphy)#threshold sf-ber 4</code></td><td>Specifies the signal fault (sf) bit error rate (ber) threshold.
• Range is 3 to 9.
• Range value is expressed exponentially as 10e-n.
Note • The default sf-ber value is 3 (10e-3).</td></tr>
<tr><td>Step 8</td><td>end or commit
Example:
<code>RP/0/RP0/CPU0:router(config-dwdm)# end</code>
or
<code>RP/0/RP0/CPU0:router(config-dwdm)# commit</code></td><td>Saves configuration changes.
• When you issue the end command, the system prompts you to commit changes:
<code>Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:</code>
- 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.</td></tr>
<tr><td>Step 9</td><td>show controllers wanphy interface-path-id [alarms | all | registers]
Example:
<code>RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 alarms
Wed Jan 20 19:25:51.462 PST
Interface: wanphy0_6_1_0
Configuration Mode: WAN Mode
BER thresholds: SF = 10e-4 SD = 10e-7
Alarm reporting enabled for: sd ber,</code></td><td>Displays the configuration of the Signal Failure (SF) and Signal Degrade (SD) Bit Error Rate (BER) reporting and thresholds.</td></tr>
</table>

Examples

The following example shows how to configure WAN-PHY Signal Failure (SF) and Signal Degrade (SD) Bit Error Rate (BER) reporting and thresholds and how to display the configuration and current statistics:

```
RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all
Wed Jan 20 19:15:44.751 PST
Interface: wanphy0_6_1_0
```

```
Configuration Mode: WAN Mode
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0
RP/0/RP0/CPU0:router(config-wanphy)# report sd-ber
RP/0/RP0/CPU0:router(config-wanphy)# threshold sd-ber 7
RP/0/RP0/CPU0:router(config-wanphy)# threshold sf-ber 4
RP/0/RP0/CPU0:router(config-wanphy)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 alarms
Wed Jan 20 19:25:51.462 PST
Interface: wanphy0_6_1_0
Configuration Mode: WAN Mode
SECTION
  LOF = 1, LOS = 1, BIP(B1) = 0
LINE
  AIS = 1, RDI = 0, FEBE = 0, BIP(B2) = 0
PATH
  AIS = 1, RDI = 0, FEBE = 0, BIP(B3) = 0
  LOP = 0, NEWPTR = 0, PSE = 0, NSE = 0
WIS ALARMS
  SER = 1, FELCDP = 0, FEAISP = 0
  WLOS = 1, PLCD = 0
  LFEBIP = 0, PBEC = 0
Active Alarms[All defects]: lof, path ais, line ais, sef,
Active Alarms[Highest Alarms]: lof
  Rx(K1/K2): N/A, Tx(K1/K2): N/A
  S1S0 = N/A, C2 = N/A
PATH TRACE BUFFER
Remote IP addr:
BER thresholds: SF = 10e-4 SD = 10e-7
TCA thresholds: N/A
Alarm reporting enabled for: sf ber, sd ber,
```