Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router

Cisco IOS XR Software Release 5.1.x

Creation Date: February 17, 2014
Revision Date: September 11, 2014

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.
# Preface

Changes to This Document  
Obtaining Documentation and Submitting a Service Request

# New and Changed Information

## Preconfiguring Physical Interfaces on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites for Preconfiguring Physical Interfaces</td>
<td>HC-6</td>
</tr>
<tr>
<td>Information About Preconfiguring Physical Interfaces</td>
<td>HC-6</td>
</tr>
<tr>
<td>Physical Interface Preconfiguration Overview</td>
<td>HC-6</td>
</tr>
<tr>
<td>Benefits of Interface Preconfiguration</td>
<td>HC-7</td>
</tr>
<tr>
<td>Use of the Interface Preconfigure Command</td>
<td>HC-7</td>
</tr>
<tr>
<td>Active and Standby RPs and Virtual Interface Configuration</td>
<td>HC-8</td>
</tr>
<tr>
<td>How to Preconfigure Physical Interfaces</td>
<td>HC-8</td>
</tr>
<tr>
<td>Configuration Examples for Preconfiguring Physical Interfaces</td>
<td>HC-10</td>
</tr>
<tr>
<td>Preconfiguring an Interface: Example</td>
<td>HC-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Additional References</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Documents</td>
<td>HC-11</td>
</tr>
<tr>
<td>Standards</td>
<td>HC-11</td>
</tr>
<tr>
<td>MIBs</td>
<td>HC-11</td>
</tr>
<tr>
<td>RFCs</td>
<td>HC-11</td>
</tr>
<tr>
<td>Technical Assistance</td>
<td>HC-12</td>
</tr>
</tbody>
</table>

## Configuring ATM Interfaces on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites for Implementing ATM</td>
<td>HC-14</td>
</tr>
<tr>
<td>Information About ATM</td>
<td>HC-15</td>
</tr>
<tr>
<td>VC-Class Mapping</td>
<td>HC-16</td>
</tr>
<tr>
<td>F5 OAM on ATM Interfaces</td>
<td>HC-16</td>
</tr>
<tr>
<td>ILMI on ATM Interfaces</td>
<td>HC-17</td>
</tr>
<tr>
<td>Layer 2 VPN on ATM Interfaces</td>
<td>HC-17</td>
</tr>
<tr>
<td>ATM Layer 2 QoS</td>
<td>HC-19</td>
</tr>
<tr>
<td>Configuring ATM Interfaces</td>
<td>HC-22</td>
</tr>
</tbody>
</table>
Bringing Up an ATM Interface
Configuring Optional ATM Interface Parameters
How to Create and Configure a Point-to-Point ATM Subinterface with a PVC
Creating a Point-to-Point ATM Subinterface with a PVC
Configuring Optional Point-to-Point ATM PVC Parameters
How to Configure a Layer 2 Attachment Circuit
Creating a Layer 2 Port Mode AC
Configuring Optional Parameters on a Layer 2 Port Mode AC
Creating an ATM Layer 2 Subinterface with a PVC
Configuring Optional ATM Layer 2 PVC Parameters
Creating an ATM Layer 2 Subinterface with a PVP
Configuring Optional ATM Layer 2 PVP Parameters
How to Create and Configure a VC-Class
Creating and Configuring a VC-Class
Attaching a VC-Class to a Point-to-Point ATM Main Interface
Attaching a VC-Class to a Point-to-Point ATM Subinterface
Attaching a VC-Class to a PVC on an ATM Subinterface
How to Configure ILMI on ATM Interfaces
Enabling ILMI on an ATM Interface
Disabling ILMI on an ATM Interface
Attaching a Service-Policy to an Attachment Circuit
ATM Configuration: Examples
ATM Interface Bring Up and Configuration: Example
Point-To-Point ATM Subinterface Configuration: Example
Layer 2 AC Creation and Configuration: Example
VC-Class Creation and Configuration: Example
ATM Layer 2 QoS Configuration: Examples
Verifying ATM Layer 2 QoS Configuration: Examples
Additional References
Related Documents
Standards
MIBs
RFCs
Technical Assistance
Advanced Configuration and Modification of the Management Ethernet Interface on Cisco IOS XR Software
Contents
Prerequisites for Configuring Management Ethernet Interfaces
Information About Configuring Management Ethernet Interfaces
Configuring Link Bundling on Cisco IOS XR Software  HC-209

Contents  HC-211

Prerequisites for Configuring Link Bundling  HC-212

Prerequisites for Configuring Link Bundling on a Cisco CRS-1 Router  HC-212

Information About Configuring Link Bundling  HC-214

Link Bundling Overview  HC-214

Features and Compatible Characteristics of Link Bundles  HC-215

Link Aggregation Through LACP  HC-216

Multichassis Link Aggregation  HC-217

LACP Short Period Time Intervals  HC-225

Load Balancing  HC-226

QoS and Link Bundling  HC-227

VLANs on an Ethernet Link Bundle  HC-227

Link Bundle Configuration Overview  HC-227

Nonstop Forwarding During RP Switchover  HC-228

Link Switchover  HC-228

How to Configure Link Bundling  HC-228

Configuring Ethernet Link Bundles  HC-229

Configuring EFP Load Balancing on an Ethernet Link Bundle  HC-233

Configuring VLAN Bundles  HC-235

Configuring POS Link Bundles  HC-242

Configuring the Default LACP Short Period Time Interval  HC-246

Configuring Custom LACP Short Period Time Intervals  HC-248

Configuring Multichassis Link Aggregation  HC-258

Configuration Examples for Link Bundling  HC-276

Example: Configuring an Ethernet Link Bundle  HC-277

Example: Configuring a VLAN Link Bundle  HC-277

Example: Configuring a POS Link Bundle  HC-277

Example: Configuring EFP Load Balancing on an Ethernet Link Bundle  HC-278

Examples: Configuring LACP Short Periods  HC-279

Additional References  HC-280

Related Documents  HC-280

Standards  HC-280

MIBs  HC-280

RFCs  HC-280

Technical Assistance  HC-281

Configuring Traffic Mirroring on the Cisco IOS XR Software  HR-283

Contents  HR-283
## Configuring Clear Channel SONET Controllers on Cisco IOS XR Software

### Contents
- Prerequisites for Configuring Clear Channel SONET Controllers
- Information About Configuring SONET Controllers
  - SONET Controller Overview
  - Default Configuration Values for SONET Controllers
- SONET APS
- How to Configure Clear Channel SONET Controllers
  - Configuring a Clear Channel SONET Controller
  - Configuring SONET APS
  - Configuring a Hold-off Timer to Prevent Fast Reroute from Being Triggered
- Configuration Examples for SONET Controllers
- Additional References

### Prerequisites for Configuring Clear Channel SONET Controllers

### Information About Configuring SONET Controllers
- SONET Controller Overview
- Default Configuration Values for SONET Controllers
- SONET APS

### How to Configure Clear Channel SONET Controllers
- Configuring a Clear Channel SONET Controller
- Configuring SONET APS
- Configuring a Hold-off Timer to Prevent Fast Reroute from Being Triggered

### Configuration Examples for SONET Controllers
- SONET Controller Configuration: Example
- SONET APS Group Configuration: Example

### Additional References
- Related Documents
- Standards
- MIBs
- RFCs
- Technical Assistance

## Configuring Clear Channel T3/E3 Controllers on Cisco IOS XR Software

### Contents
- Prerequisites for Configuring T3/E3 Controllers
- Information About T3/E3 Controllers and Serial Interfaces
  - Configuration Overview
  - Default Configuration Values for T3 and E3 Controllers
- How to Configure Clear Channel T3/E3 Controllers
  - Setting the Card Type
  - Configuring a Clear Channel E3 Controller
  - Modifying the Default E3 Controller Configuration
  - Configuring a Clear Channel T3 Controller
  - Modifying the Default T3 Controller Configuration
  - Configuring Serial Interfaces on Cisco IOS XR Software
    - Configuring Serial Interfaces on Cisco IOS XR Software
    - Configuring BERT
- Configuration Examples
  - Configuring a Clear Channel T3 Controller: Example
  - Configuring BERT on a T3 Controller: Example

### Prerequisites for Configuring T3/E3 Controllers

### Information About T3/E3 Controllers and Serial Interfaces
- Configuration Overview
- Default Configuration Values for T3 and E3 Controllers

### How to Configure Clear Channel T3/E3 Controllers
- Setting the Card Type
- Configuring a Clear Channel E3 Controller
- Modifying the Default E3 Controller Configuration
- Configuring a Clear Channel T3 Controller
- Modifying the Default T3 Controller Configuration
- Configuring Serial Interfaces on Cisco IOS XR Software
- Configuring Serial Interfaces on Cisco IOS XR Software
- Configuring BERT

### Configuration Examples
- Configuring a Clear Channel T3 Controller: Example
- Configuring BERT on a T3 Controller: Example

### Additional References
Related Documents HC-348
Standards HC-348
MIBs HC-348
RFCs HC-349
Technical Assistance HC-349

Configuring Dense Wavelength Division Multiplexing Controllers on Cisco IOS XR Software HC-351

Contents HC-351

Prerequisites for Configuring DWDM Controller Interfaces HC-352
New DWDM Configuration Requirement HC-352
Information About the DWDM Controllers HC-352
Information about IPoDWDM HC-353
How to Configure DWDM Controllers HC-354
Configuring the Optical Parameters HC-355
Configuring G.709 Parameters HC-357
How to Perform Performance Monitoring on DWDM Controllers HC-360
Configuring DWDM Controller Performance Monitoring HC-360
Configuring IPoDWDM HC-364
Configuring the SRLG and Optical Layer DWDM Ports HC-364
Configuring the Administrative State of DWDM Optical Ports HC-366
Configuring Proactive FEC-FRR Triggering HC-368

Configuration Examples HC-370
Turning On the Laser: Example HC-370
Turning Off the Laser: Example HC-371
DWDM Controller Configuration: Examples HC-371
DWDM Performance Monitoring: Examples HC-371
IPoDWDM Configuration: Examples HC-372

Additional References HC-373
Related Documents HC-373
Standards HC-374
MIBs HC-374
RFCs HC-374
Technical Assistance HC-374

Configuring POS Interfaces on Cisco IOS XR Software HC-375

Contents HC-375

Prerequisites for Configuring POS Interfaces HC-376
Information About Configuring POS Interfaces HC-376
Configuring Serial Interfaces on Cisco IOS XR Software  HC-425

Contents  HC-425

Prerequisites for Configuring Serial Interfaces  HC-425

Information About Configuring Serial Interfaces  HC-426

High-Level Overview: Serial Interface Configuration on Clear-Channel SPAs  HC-426
Cisco HDLC Encapsulation  HC-427
PPP Encapsulation  HC-427
Keepalive Timer  HC-429
Frame Relay Encapsulation  HC-430
Layer 2 Tunnel Protocol Version 3-Based Layer 2 VPN on Frame Relay  HC-431
Default Settings for Serial Interface Configurations  HC-431
Serial Interface Naming Notation  HC-432

How to Configure Serial Interfaces  HC-433

Bringing Up a Serial Interface  HC-433
Configuring Optional Serial Interface Parameters  HC-436
Creating a Point-to-Point Serial Subinterface with a PVC  HC-439
Configuring Optional PVC Parameters  HC-441
Modifying the Keepalive Interval on Serial Interfaces  HC-443
How to Configure a Layer 2 Attachment Circuit  HC-445
Creating a Serial Layer 2 Subinterface with a PVC  HC-445
Configuring Optional Serial Layer 2 PVC Parameters  HC-447

Configuration Examples for Serial Interfaces  HC-450

Bringing Up and Configuring a Serial Interface with Cisco HDLC Encapsulation: Example  HC-450
Configuring a Serial Interface with Frame Relay Encapsulation: Example  HC-451
Configuring a Serial Interface with PPP Encapsulation: Example  HC-452

Additional References  HC-453
Related Documents  HC-453
Standards  HC-453
MIBs  HC-453
RFCs  HC-453
Technical Assistance  HC-454

Configuring PPP on Cisco IOS XR Software  HC-455

Contents  HC-455
Prerequisites for Configuring PPP  HC-456
Information About PPP  HC-456
  PPP Authentication  HC-456
  Multilink PPP  HC-458
  Supported Cards  HC-459
How to Configure PPP  HC-459
  Modifying the Default PPP Configuration  HC-459
  Configuring PPP Authentication  HC-463
  Disabling an Authentication Protocol  HC-471
  Configuring Multilink PPP  HC-476
Configuration Examples for PPP  HC-486
  Configuring a POS Interface with PPP Encapsulation: Example  HC-486
  Configuring a Serial Interface with PPP Encapsulation: Example  HC-487
  Configuring MLPPP: Example  HC-487
  Verifying Multilink PPP Configurations  HC-488
Additional References  HC-492
  Related Documents  HC-492
  Standards  HC-492
  MIBs  HC-492
  RFCs  HC-492
  Technical Assistance  HC-493

Configuring 802.1Q VLAN Interfaces on Cisco IOS XR Software  HC-495
Contents  HC-495
Prerequisites for Configuring 802.1Q VLAN Interfaces  HC-496
Information About Configuring 802.1Q VLAN Interfaces  HC-496
  802.1Q VLAN Overview  HC-496
  802.1Q Tagged Frames  HC-496
  CFM on 802.1Q VLAN Interfaces  HC-497
  Subinterfaces  HC-497
  Subinterface MTU  HC-497
  Native VLAN  HC-497
  VLAN Subinterfaces on Ethernet Bundles  HC-498
  Layer 2 VPN on VLANS  HC-498
How to Configure 802.1Q VLAN Interfaces  HC-499
  Configuring 802.1Q VLAN Subinterfaces  HC-499
  Configuring Native VLAN  HC-502
  Configuring an Attachment Circuit on a VLAN  HC-503
  Removing an 802.1Q VLAN Subinterface  HC-506
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
- Obtaining Documentation and Submitting a Service Request

Changes to This Document

Table 1 lists the technical changes made to this document since it was first printed.

<table>
<thead>
<tr>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2014</td>
<td>Republished with documentation updates for Cisco IOS XR Release 5.1.3.</td>
</tr>
<tr>
<td>May 2014</td>
<td>Republished with documentation updates for Cisco IOS XR Release 5.1.2.</td>
</tr>
<tr>
<td>February 2014</td>
<td>Initial release of this document.</td>
</tr>
</tbody>
</table>

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:


Subscribe to the What’s New in Cisco Product Documentation as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.
New and Changed Information in Release 5.1.x

This table summarizes the new and changed feature information for the Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router, and tells you where they are documented.

For a complete list of new and changed features in Cisco IOS XR Software, Release 5.1.x, see the New and Changed Features in Cisco IOS XR Software, Release 5.1.x for Cisco CRS Router document.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFM down MEP on VRF interfaces</td>
<td>This feature was introduced.</td>
<td>Release 5.1.2</td>
<td>Configuring Ethernet OAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Configuring CFM MEPs</td>
</tr>
<tr>
<td>CFM over Bundles</td>
<td>This feature was introduced.</td>
<td>Release 5.1.3</td>
<td>CFM over Bundles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• CFM over Bundles</td>
</tr>
</tbody>
</table>
Preconfiguring Physical Interfaces on Cisco IOS XR Software

This module describes the preconfiguration of physical interfaces on the Cisco IOS XR Software. 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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>POS preconfiguration was introduced.</td>
</tr>
<tr>
<td>Release 3.0</td>
<td>Ethernet preconfiguration was introduced.</td>
</tr>
</tbody>
</table>
Preconfiguring Physical Interfaces on Cisco IOS XR Software

Contents

- Prerequisites for Preconfiguring Physical Interfaces, page 6
- Information About Preconfiguring Physical Interfaces, page 6
- How to Preconfigure Physical Interfaces, page 8
- Configuration Examples for Preconfiguring Physical Interfaces, page 10
- Additional References, page 11

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 conditions are 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, page 6
- Benefits of Interface Preconfiguration, page 7
- Use of the Interface Preconfigure Command, page 7
- Active and Standby RPs and Virtual Interface Configuration, page 8

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.

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

Interaces 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.

Note 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.
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. ipv4 address ip-address subnet-mask
4. Configure additional interface parameters.
5. end
   or
   commit
6. exit
7. exit
8. show running-config
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
</tbody>
</table>

**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**
- ipv4 address *ip-address* *subnet-mask*
- or
- 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.
Preconfiguring Physical Interfaces on Cisco IOS XR Software

Configuration Examples for Preconfiguring Physical Interfaces

This section contains the following example:

Preconfiguring an Interface: Example, page 10

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
```

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>end or commit best-effort</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if-pre)# end or commit</td>
<td></td>
</tr>
</tbody>
</table>

### Step 5

**Example:**

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

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

- Use the `commit 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

**Example:**

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

(Optional) Displays the configuration information currently running on the router.

Configuration Examples for Preconfiguring Physical Interfaces

Preconfiguring an Interface: Example, page 10
Additional References

The sections that follow provide references related to the preconfiguration of physical interfaces.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master command reference</td>
<td><em>Cisco CRS-1 Router Master Command Listing</em></td>
</tr>
<tr>
<td>Interface configuration commands</td>
<td><em>Cisco CRS-1 Router Interface and Hardware Component Command Reference</em></td>
</tr>
<tr>
<td>Initial system bootup and configuration information</td>
<td><em>Cisco IOS XR Getting Started Guide for the Cisco CRS-1 Router</em></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td><em>Cisco IOS XR Task ID Reference Guide</em></td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no applicable MIBs for this module.</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>able technical content, including links to products, technologies, solutions,</td>
<td></td>
</tr>
<tr>
<td>technical tips, and tools. Registered Cisco.com users can log in from this</td>
<td></td>
</tr>
<tr>
<td>page to access even more content.</td>
<td></td>
</tr>
</tbody>
</table>
Configuring ATM Interfaces on Cisco IOS XR Software

This module describes how to configure ATM on the Cisco CRS-1 Router using Cisco IOS XR software. 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 on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.0</td>
<td>ATM Layer 2 VPN (Port Mode) and QoS was introduced on the following SPAs:</td>
</tr>
<tr>
<td></td>
<td>- 3-Port Clear Channel OC-3 ATM SPA</td>
</tr>
<tr>
<td></td>
<td>- 1-Port Clear Channel OC-12 ATM SPA</td>
</tr>
</tbody>
</table>
### 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:

- VC-Class Mapping, page 16
- F5 OAM on ATM Interfaces, page 16
- ILMI on ATM Interfaces, page 17
- Layer 2 VPN on ATM Interfaces, page 17
- ATM Layer 2 QoS, page 19

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 the following 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 PVC has unspecified bit rate (UBR) traffic shaping configured, but it is attached to a class map that is configured 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 F5 OAM 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.

**Note**
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.

**Note**
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 – policing, marking, and queueing 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 AAL5NLPIIID 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.
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)

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.

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.

Note
Configuring ATM Interfaces

The ATM interface configuration tasks are described in the following procedures:

- **Bringing Up an ATM Interface, page 22**
- **Configuring Optional ATM Interface Parameters, page 24**
- **How to Create and Configure a Point-to-Point ATM Subinterface with a PVC, page 27**
  - Creating a Point-to-Point ATM Subinterface with a PVC, page 27
  - Configuring Optional Point-to-Point ATM PVC Parameters, page 29
- **How to Configure a Layer 2 Attachment Circuit, page 32**
  - Creating a Layer 2 Port Mode AC, page 32
  - Configuring Optional Parameters on a Layer 2 Port Mode AC, page 34
  - Creating an ATM Layer 2 Subinterface with a PVC, page 35
  - Configuring Optional ATM Layer 2 PVC Parameters, page 37
  - Creating an ATM Layer 2 Subinterface with a PVP, page 40
  - Configuring Optional ATM Layer 2 PVP Parameters, page 41
- **How to Create and Configure a VC-Class, page 43**
  - Creating and Configuring a VC-Class, page 44
  - Attaching a VC-Class to a Point-to-Point ATM Main Interface, page 47
  - Attaching a VC-Class to a Point-to-Point ATM Subinterface, page 48
  - Attaching a VC-Class to a PVC on an ATM Subinterface, page 49
- **How to Configure ILMI on ATM Interfaces, page 51**
  - Enabling ILMI on an ATM Interface, page 51
  - Disabling ILMI on an ATM Interface, page 53
- **Attaching a Service-Policy to an Attachment Circuit, page 56**

Bringing Up an ATM Interface

This task describes the commands used to bring 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. `no shutdown`
4. `end`
   or
   `commit`
5. `exit`
6. `exit`
7. Repeat Step 1 through Step 6 to bring up the interface at the other end of the connection.
8. `show interfaces atm interface-path-id brief`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface atm interface-path-id</td>
<td>Enters ATM interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> no shutdown</td>
<td>Removes the shutdown configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router (config-if)# no shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router (config-if)# end</code></td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router (config-if)# exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Removal of the shutdown configuration eliminates the forced administrative down on the interface, enabling it to move to an up or down state.</td>
</tr>
</tbody>
</table>

• When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)?
  [cancel]:

  – Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

  – Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

  – Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

• Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
### Configuring ATM Interfaces

#### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><code>exit</code></td>
<td>Exits global configuration mode and enters EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router (config)# exit</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Repeat Step 1 through Step 6 to bring up the interface at the other end of the connection.</td>
<td>Brings up the connection. <strong>Note</strong> The configuration on both ends of the ATM connection must match.</td>
</tr>
<tr>
<td>8</td>
<td><code>show interfaces atm interface-path-id brief</code></td>
<td>(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 <code>show interfaces atm</code> command output shows “up.”</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router# show interfaces atm 0/6/0/1 brief</code></td>
<td></td>
</tr>
</tbody>
</table>

#### 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 24.
- 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 27.
- To create a vp-tunnel on the ATM interface you just brought up, see the “How to Configure a Layer 2 Attachment Circuit” section on page 32.
- To use the interface as a Layer 2 post mode AC, see the “How to Configure a Layer 2 Attachment Circuit” section on page 32.
- 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 43.
- To enable ILMI on the ATM interface you just brought up, see the “How to Configure ILMI on ATM Interfaces” section on page 51.

### Configuring Optional ATM Interface Parameters

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

#### Prerequisites

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” section on page 22.

#### 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 brief

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

<table>
<thead>
<tr>
<th>Step 6</th>
<th>end or commit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router (config-if)# end</code> or <code>RP/0/RP0/CPU0:router(config-if)# commit</code></td>
</tr>
</tbody>
</table>

**Purpose**
- Saves configuration changes.
- When you issue the `end` command, the system prompts you to commit changes:
  - 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.

<table>
<thead>
<tr>
<th>Step 7</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router (config-if)# exit</code></td>
</tr>
</tbody>
</table>

**Purpose**
- Exits interface configuration mode and enters global configuration mode.

<table>
<thead>
<tr>
<th>Step 8</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router (config)# exit</code></td>
</tr>
</tbody>
</table>

**Purpose**
- Exits global configuration mode and enters EXEC mode.

<table>
<thead>
<tr>
<th>Step 9</th>
<th>show atm interface atm [interface-path-id]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router# show atm interface atm 0/6/0/1</code></td>
</tr>
</tbody>
</table>

**Purpose**
- (Optional) Displays ATM-specific data for the specified ATM interface.

<table>
<thead>
<tr>
<th>Step 10</th>
<th>show interfaces atm interface-path-id</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router# show interfaces atm 0/6/0/1</code></td>
</tr>
</tbody>
</table>

**Purpose**
- (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 27.
- To create a vp-tunnel on the ATM interface you just brought up, see the “How to Configure a Layer 2 Attachment Circuit” section on page 32.
- To use the interface as a Layer 2 ATM AC, see the “How to Configure a Layer 2 Attachment Circuit” section on page 32.
- 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 43.
- To enable ILMI on the ATM interface you just brought up, see the “How to Configure ILMI on ATM Interfaces” section on page 51.

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, page 27
- Configuring Optional Point-to-Point ATM PVC Parameters, page 29

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.

Prerequisites

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” section on page 22.

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
6. Repeat Step 1 through Step 5 to bring up the ATM subinterface and any associated PVC at the other end of the connection.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>configure</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>interface atm interface-path-id.subinterface point-to-point</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>interface atm 0/6/0/1.10</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>ipv4 address ipv4_address/prefix</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>ipv4 address 10.46.8.6/24</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>pvc vpi/vci</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>pvc 5/10</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>end</code> or <code>commit</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>end</code> or <code>commit</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Repeat Step 1 through Step 5 to bring up the ATM subinterface and any associated PVC at the other end of the connection.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The configuration on both ends of the subinterface connection must match.</td>
</tr>
</tbody>
</table>
What to Do Next

- To configure optional PVC parameters, see the “Configuring Optional Point-to-Point ATM PVC Parameters” section on page 29.
- 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 on page 44.

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.

Prerequisites

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” section on page 27.

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-aisdi-failure]]`
6. `oam ais-rdi [down-count [up-count]]`
7. `oam retry`
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.
## Configuring ATM Interfaces

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Configure</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>configure</code></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router# configure</code></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

| **Step 2** | **Interface atm interface-path-id.subinterface point-to-point** |
| **Example:** | `interface atm interface-path-id.subinterface point-to-point` |
| | `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:** | `pvc vpi/vci` |
| | `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:** | `encapsulation {aal5mux ipv4 | aal5nlpid | aal5snap}` |
| | `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:** | `oam-pvc manage [frequency] [disable] [keep-vc-up [seg-aisrdi-failure]]` |
| | `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 will not be brought down because of end CC failure or loopback failure. |

| **Step 6** | **OAM AIS RDI [down-count [up-count]]** |
| **Example:** | `oam ais-rdi [down-count [up-count]]` |
| | `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:** | `oam retry [up-count [down-count [retry-frequency]]]` |
| | `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. |
### Command or Action

**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
```

**Step 9**

```
service-policy [input | output] policy_name
```

*Example:*

```
RP/0/RP0/CPU0:router (config-atm-vc)# service-policy input policyA
```

**Step 10**

```
end
or
commit
```

*Example:*

```
RP/0/RP0/CPU0:router (config-subif)# end
or
RP/0/RP0/CPU0:router(config-subif)# commit
```

**Step 11**

Repeat Step 1 through Step 10 to configure the PVC at the other end of the connection.

### Purpose

- 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.

- 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*.

- 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.

- 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 on page 44.
How to Configure a Layer 2 Attachment Circuit

The Layer 2 AC configuration tasks are described in the following procedures:

- Creating a Layer 2 Port Mode AC
- Configuring Optional Parameters on a Layer 2 Port Mode AC
- Creating an ATM Layer 2 Subinterface with a PVC
- Configuring Optional ATM Layer 2 PVC Parameters
- Creating an ATM Layer 2 Subinterface with a PVP
- Configuring Optional ATM Layer 2 PVP 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 Layer 2 Port Mode AC

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

Prerequisites

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” section on page 22.

Restrictions

ILMI configuration is not supported on Layer 2 port mode ACs.

Restrictions

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`
5. Repeat Step 1 through Step 4 to bring up the ATM AC at the other end of the connection.
DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>interface atm interface-path-id</td>
<td>Enters interface configuration mode for an ATM interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>l2transport</td>
<td>Enters ATM Layer 2 transport configuration mode, and enables Layer 2 port mode on this ATM interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router (config-if)# l2transport</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>or commit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router (config-if-l2)# end</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if-l2)# commit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>When you issue the end command, the system prompts you to commit changes:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[cancel]:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Enter yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Repeat Step 1 through Step 4 to bring up the Layer 2 port mode AC at the other end of the connection.</td>
<td>Brings up the Layer 2 port mode AC.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>The configuration on both ends of the connection must match.</td>
</tr>
</tbody>
</table>

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 34.
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.

Prerequisites

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” section on page 32.

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
6. end
   or
   commit
7. Repeat Step 1 through Step 6 to configure the Layer 2 port mode AC at the other end of the connection.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface atm interface-path-id</td>
<td>Enters interface configuration mode for an ATM interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> atm mcpt-timers timer-1 timer-2 timer-3</td>
<td>Specifies the maximum cell packing timeout values for each of the three per-interface MCPT timers, in microseconds.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config-if)# atm mcpt-timers 50 100 200</td>
<td><strong>Note</strong> The default value for each timer is 50 microseconds.</td>
</tr>
<tr>
<td><strong>Step 4</strong> l2transport</td>
<td>Enters ATM Layer 2 transport configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config-if)# l2transport</td>
<td></td>
</tr>
</tbody>
</table>
Create an ATM Layer 2 Subinterface with a PVC

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

Prerequisites

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” section on page 22.

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
5. Repeat Step 1 through Step 4 to bring up the ATM subinterface and any associated PVC at the other end of the AC.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface atm interface-path-id.subinterface l2transport</td>
<td>Creates a subinterface and enters ATM subinterface configuration mode for that subinterface.</td>
</tr>
<tr>
<td><strong>Step 3</strong> pvc vpi/vci</td>
<td>Creates an ATM permanent virtual circuit (PVC) and enters ATM Layer 2 transport PVC configuration mode.</td>
</tr>
<tr>
<td><strong>Note</strong> Only one PVC is allowed per subinterface.</td>
<td></td>
</tr>
</tbody>
</table>
### 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.

### Prerequisites

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” section on page 35.
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 vpivc1
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
8. Repeat Step 1 through Step 7 to configure the PVC at the other end of the AC.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters ATM subinterface configuration mode for a Layer 2 ATM subinterface.</td>
</tr>
<tr>
<td>interface atm</td>
<td>Example: RP/0/RP0/CPU0:router(config-if)# interface atm 0/6/0/1.10 l2transport</td>
</tr>
<tr>
<td>interface-path-id</td>
<td>subinterface l2transport</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters ATM Layer 2 transport PVC configuration mode for the specified PVC.</td>
</tr>
<tr>
<td>pvc</td>
<td>Example: RP/0/RP0/CPU0:router(config-atm-l2transport-pvc # pvc 5/20</td>
</tr>
<tr>
<td>vpivc1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures the ATM adaptation layer (AAL) and encapsulation type for a PVC.</td>
</tr>
<tr>
<td>encapsulation</td>
<td>Example: RP/0/RP0/CPU0:router(config-atm-l2transport-pvc # encapsulation aal5</td>
</tr>
<tr>
<td>{aal0</td>
<td>aal5}</td>
</tr>
</tbody>
</table>
## Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th><strong>cell-packing</strong> <em>cells timer</em></th>
</tr>
</thead>
</table>

**Example:**
```
RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# 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.

| Step 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)* |
|--------|-------------------------------------------------------------|

**Example:**
```
RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# 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.

<table>
<thead>
<tr>
<th>Step 7</th>
<th><strong>end</strong> or <strong>commit</strong></th>
</tr>
</thead>
</table>

**Example:**
```
RP/0/RP0/CPU0:router(config-atm-l2transport-pvc)# end
```

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

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

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

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Repeat Step 1 through Step 7 to configure the PVC at the other end of the AC.</th>
</tr>
</thead>
</table>

Brings up the AC.

**Note**  The configuration on both ends of the connection must match.
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.

Prerequisites

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” section on page 22.

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
5. Repeat Step 1 through Step 4 to bring up the ATM subinterface and any associated PVP at the other end of the AC.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2 interface atm interface-path-id.subinterface l2transport</td>
<td>Creates an ATM subinterface and enters ATM layer2 transport configuration mode for that subinterface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface atm 0/6/0/1.10 l2transport</td>
</tr>
</tbody>
</table>
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.

Prerequisites

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” section on page 40.
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
7.  Repeat Step 1 through Step 6 to configure the PVP at the other end of the connection.

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface atm interface-path-id.subinterface l2transport</td>
<td>Enters ATM subinterface configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface atm 0/6/0/1.10 l2transport</td>
<td></td>
</tr>
<tr>
<td>Step 3 pvp vpi</td>
<td>Enters subinterface configuration mode for the PVP.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# pvp 10</td>
<td></td>
</tr>
<tr>
<td>Step 4 cell-packing cells timer</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.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-atm-12transport-pvp)# cell-packing 5 2</td>
<td></td>
</tr>
</tbody>
</table>

- 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 5: Shape ATM traffic for the PVC.

```
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
```

Step 6: End or commit configuration.

```
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
```

Step 7: Repeat Step 1 through Step 6 to configure the PVP at the other end of the AC.

```
Repeat Step 1 through Step 6 to configure the PVP at the other end of the AC.
```

Step 7: Repeat 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.

What to Do Next

- 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 the following procedures:

- Creating and Configuring a VC-Class, page 44
- Attaching a VC-Class to a Point-to-Point ATM Main Interface, page 47
Configuring ATM Interfaces

• Attaching a VC-Class to a Point-to-Point ATM Subinterface, page 48
• Attaching a VC-Class to a PVC on an ATM Subinterface, page 49

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
</tr>
</tbody>
</table>

Example:

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

| **Step 2** | `vc-class atm name` | Creates a vc-class for an ATM interface and enters vc-class configuration mode. |

Example:

```
RP/0/RP0/CPU0:router (config)# vc-class atm class1
```
### Configuring ATM Interfaces on Cisco IOS XR Software

#### Configuring ATM Interfaces

**Step 3**

**encapsulation** `{aal5mux ipv4 | aal5nlpid | aal5snap}

**Example:**

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

- `encapsulation aal5snap`

**Purpose:**

- 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.

**Step 4**

**oam ais-rdi** `[down-count [up-count]]

**Example:**

RP/0/RP0/CPU0:router (config-vc-class-atm)# oam ais-rdi 25 5

**Purpose:**

- 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

**Purpose:**

- 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

**Purpose:**

- 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.
Configuring ATM Interfaces

Configuring ATM Interfaces on Cisco IOS XR Software

Step 7

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`shape [cbr peak_output_rate</td>
<td>ubr peak_output_rate</td>
</tr>
<tr>
<td><code>shape vbr-nrt 100000 100000 8000</code></td>
<td></td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router (config-vc-class-atm)# shape vbr-nrt 100000 100000 8000
```

Step 8

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router (config-if)# end</code> or <code>RP/0/RP0/CPU0:router(config-if)# commit</code></td>
<td></td>
</tr>
</tbody>
</table>

Example:

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

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 -47.
- 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 -48.
- 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 -49.
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface atm</td>
<td>Enters ATM interface configuration mode.</td>
</tr>
<tr>
<td>interface-path-id</td>
<td></td>
</tr>
<tr>
<td>point-to-point</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config)#</td>
<td></td>
</tr>
<tr>
<td>interface atm 0/6/0/1 point-to-point</td>
<td></td>
</tr>
</tbody>
</table>
### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3</td>
<td><strong>class-int</strong> vc-class-name</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router (config-if)# class-int classA</td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>end</strong> or <strong>commit</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router (config-if)# end or commit</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>interface atm interface-path-id.subinterface point-to-point</code></td>
<td>Enters ATM subinterface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1.10 point-to-point</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>class-int vc-class-name</code></td>
<td>Assigns the vc-class to an ATM subinterface. Replace the <code>vc-class-name</code> argument with the name of the vc-class you configured in the “Creating and Configuring a VC-Class” section on page -44.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config-subif)# class-int classA</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config-subif)# end or RP/0/RP0/CPU0:router(config-subif)# commit</td>
<td></td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface atm interface-path-id[.subinterface] [point-to-point</td>
<td>l2transport]</td>
</tr>
<tr>
<td>Example:</td>
<td>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.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1.10</td>
<td>Note For more information on creating and configuring ATM subinterfaces, see the “Creating a Point-to-Point ATM Subinterface with a PVC” section on page 27.</td>
</tr>
<tr>
<td>Step 3 pvc vpi/vci</td>
<td>Enters ATM PVC configuration mode and creates the PVC if it does not already exist.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note For more information on creating and configuring PVCs on ATM subinterfaces, see the “Creating a Point-to-Point ATM Subinterface with a PVC” section on page 27.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config-if)# pvc 5/50</td>
<td></td>
</tr>
</tbody>
</table>
Configuring ATM Interfaces on Cisco IOS XR Software

Configuring ATM Interfaces

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, page 51
- Disabling ILMI on an ATM Interface, page 53

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.

Prerequisites

Bring up the ATM interface and remove the shutdown configuration, as described in the “Bringing Up an ATM Interface” section on page 22.

---

### Command or Action | Purpose
--- | ---
**Step 4** | 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.

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

**Step 5** | 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.

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

or

```
RP/0/RP0/CPU0:router (config-if)# commit
```
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
   or
   commit
7. exit
8. exit
9. show atm ilmi-status [atm interface-path-id]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface atm interface-path-id</td>
<td>Enters ATM interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1</td>
<td></td>
</tr>
<tr>
<td>Step 3 atm address-registration</td>
<td>(Optional) Enables the router to engage in address registration and callback functions with the Interim Local Management Interface (ILMI).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config-if)# atm address-registration</td>
<td></td>
</tr>
<tr>
<td>Step 4 atm ilmi-keepalive [act-poll-freq frequency]</td>
<td>(Optional) Enables ILMI keepalives on an ATM interface.</td>
</tr>
<tr>
<td>[retries count] [inact-poll-freq frequency]</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config-if)# atm ilmi-keepalive</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface atm interface-path-id</td>
<td>Enters ATM interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config)# interface atm 0/6/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> atm ilmi-config disable</td>
<td>(Optional) Disables ILMI on the ATM interface. To re-enable ILMI on an ATM interface, use the <code>no atm ilmi-config disable</code> form of this command.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config-if)# atm ilmi-config disable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>or</td>
<td>• When you issue the <code>end</code> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td>commit</td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering <code>yes</code> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering <code>no</code> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering <code>cancel</code> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config-if)# end or RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router (config-if)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router (config)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits global configuration mode and enters EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>show atm ilmi-status [atm interface-path-id]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router (config)# show atm ilmi-status atm 0/6/0/1</td>
</tr>
<tr>
<td></td>
<td>(Optional) Verifies the ILMI configuration for the specified interface.</td>
</tr>
</tbody>
</table>
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.

**SUMMARY STEPS**

**PVC Mode**
1. `config`
2. `interface atm interface-path-id.subinterface l2transport`
3. `pvc vpi/vci`
4. `service-policy input | output policy_name`
5. `commit`

**PVP Mode**
1. `config`
2. `interface atm interface-path-id.subinterface l2transport`
3. `pvp vpi`
4. `service-policy input | output policy_name`
5. `commit`

**Port Mode**
1. `config`
2. `interface atm interface-path-id`
3. `l2transport`
4. `service-policy input | output policy_name`
5. `commit`

**Main Interface (non-port mode)**
1. `config`
2. `interface atm interface-path-id`
3. `service-policy input | output policy_name`
4. `commit`
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PVC Mode</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>config</strong> Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# config terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>interface atm interface-path-id.subinterface l2transport</strong> Creates a subinterface and enters ATM subinterface configuration mode for that subinterface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface atm 0/1/0/0.2 l2transport</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>interface atm interface-path-id</strong> Enters interface configuration mode for an ATM interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router (config)# interface atm 0/1/0/1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**service-policy input</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0(config-atm-l2transport-pvc)#service-policy input</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>commit</strong> Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td><strong>PVP Mode</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>config</strong> Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# config terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>interface atm interface-path-id.subinterface l2transport</strong> Creates a subinterface and enters ATM subinterface configuration mode for that subinterface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface atm 0/1/0/0.2 l2transport</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>pvp vpi</strong> (Optional) Creates an ATM PVP and enters ATM PVP configuration submode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-subif)# pvp 30</td>
</tr>
</tbody>
</table>
### Configuring ATM Interfaces

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> service-policy input</td>
<td>output policy_name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0(config-atm-l2transport-pvp)#service-policy input</td>
<td>output atm_policy_2</td>
</tr>
<tr>
<td><strong>Step 5</strong> commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
</tbody>
</table>

### Port Mode

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> config</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# config terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface atm interface-path-id</td>
<td>Enters interface configuration mode for an ATM interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config)# interface atm 0/1/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> l2transport</td>
<td>Enters ATM Layer 2 transport configuration mode, and enables Layer 2 port mode on this ATM interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config-if)# l2transport</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service-policy input</td>
<td>output policy_name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0(config-if-l2)#service-policy input</td>
<td>output atm_policy_3</td>
</tr>
<tr>
<td><strong>Step 5</strong> commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
</tbody>
</table>

### Main Interface (non-port mode)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> config</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# config terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface atm interface-path-id</td>
<td>Enters interface configuration mode for an ATM interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config)# interface atm 0/1/0/1</td>
<td></td>
</tr>
</tbody>
</table>
ATM Configuration: Examples

This section provides the following configuration examples:

- ATM Interface Bring Up and Configuration: Example, page 59
- Point-To-Point ATM Subinterface Configuration: Example, page 59
- Layer 2 AC Creation and Configuration: Example, page 61
- VC-Class Creation and Configuration: Example, page 62
- ATM Layer 2 QoS Configuration: Examples, page 63
- Verifying ATM Layer 2 QoS Configuration: Examples, page 65

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

ATM Configuration: Examples

This section provides the following configuration examples:

- ATM Interface Bring Up and Configuration: Example, page 59
- Point-To-Point ATM Subinterface Configuration: Example, page 59
- Layer 2 AC Creation and Configuration: Example, page 61
- VC-Class Creation and Configuration: Example, page 62
- ATM Layer 2 QoS Configuration: Examples, page 63
- Verifying ATM Layer 2 QoS Configuration: Examples, page 65
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 PVCs 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),
Layer 2 AC Creation and Configuration: Example

The following 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)# 12transport
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 12transport
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

<table>
<thead>
<tr>
<th>Interface</th>
<th>VPI</th>
<th>VCI</th>
<th>Type</th>
<th>Encaps</th>
<th>SC</th>
<th>Peak</th>
<th>Avg/Min</th>
<th>Burst</th>
<th>Cells</th>
<th>Sts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM0/1/0/0.230</td>
<td>15</td>
<td>230</td>
<td>PVC</td>
<td>AAL0</td>
<td>UBR</td>
<td>622000</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>UP</td>
</tr>
<tr>
<td>ATM0/1/0/3.19</td>
<td>17</td>
<td>19</td>
<td>PVC</td>
<td>SNAP</td>
<td>UBR</td>
<td>622000</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>UP</td>
</tr>
</tbody>
</table>
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
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>VPI</th>
<th>SC</th>
<th>Kbps</th>
<th>Kbps</th>
<th>Cells</th>
<th>Sts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM0/6/0/1.10</td>
<td>100</td>
<td>UBR</td>
<td>155000</td>
<td>N/A</td>
<td>N/A</td>
<td>UP</td>
</tr>
</tbody>
</table>

### 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 cellspsp 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 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
```
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
Additional References

The following sections provide references related to implementing ATM for Cisco IOS XR software.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples</td>
<td>Cisco IOS XR Interface and Hardware Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1483</td>
<td>Multiprotocol Encapsulation over ATM Adaptation Layer 5</td>
</tr>
<tr>
<td>RFC 1577</td>
<td>Classical IP and ARP over ATM.</td>
</tr>
<tr>
<td>RFC 2225</td>
<td>Classical IP and ARP over ATM</td>
</tr>
<tr>
<td>RFC 2255</td>
<td>The LDAP URL Format</td>
</tr>
<tr>
<td>RFC 2684</td>
<td>Multiprotocol Encapsulation over ATM Adaptation Layer 5.</td>
</tr>
<tr>
<td>RFC 4385</td>
<td>Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN</td>
</tr>
<tr>
<td>RFC 4717</td>
<td>Encapsulation Methods for Transport of Asynchronous Transfer Mode (ATM) over MPLS Networks</td>
</tr>
<tr>
<td>RFC 4816</td>
<td>Pseudowire Emulation Edge-to-Edge (PWE3) Asynchronous Transfer Mode (ATM) Transparent Cell Transport Service</td>
</tr>
</tbody>
</table>
## Technical Assistance

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

This module describes the configuration of Management Ethernet interfaces on the Cisco IOS XR Software.

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 Cisco IOS XR Getting Started Guide. This module describes how to modify the default configuration of the Management Ethernet interface after it has been configured, as described in Cisco IOS XR 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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>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.</td>
</tr>
</tbody>
</table>

## Contents

- Prerequisites for Configuring Management Ethernet Interfaces, page 70
- Information About Configuring Management Ethernet Interfaces, page 70
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, page 70

Default Interface Settings

Table 1 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.
How to Perform Advanced Management Ethernet Interface Configuration

This section contains the following procedures:

- Configuring a Management Ethernet Interface, page 71 (required)
- Configuring the Duplex Mode for a Management Ethernet Interface, page 73 (optional)
- Configuring the Speed for a Management Ethernet Interface, page 74 (optional)
- Modifying the MAC Address for a Management Ethernet Interface, page 76 (optional)
- Verifying Management Ethernet Interface Configuration, page 77 (optional)

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 Configuring General Router Features 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`

---

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Management Ethernet Interface Default Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Default Value</td>
</tr>
</tbody>
</table>
| 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. |
| 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

6. end or commit

7. show interfaces MgmtEth interface-path-id

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface MgmtEth interface-path-id</td>
<td>Enters interface configuration mode and specifies the Ethernet interface name and notation rack/slot/module/port.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0</td>
<td>The example indicates port 0 on the RP card that is installed in slot 0.</td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv4 address ip-address mask</td>
<td>Assigns an IP address and subnet mask to the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224</td>
<td>• Replace <em>ip-address</em> with the primary IPv4 address for the interface.</td>
</tr>
<tr>
<td></td>
<td>• Replace <em>mask</em> with the mask for the associated IP subnet. The network mask can be specified in either of two ways:</td>
</tr>
<tr>
<td></td>
<td>– 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.</td>
</tr>
<tr>
<td></td>
<td>– 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.</td>
</tr>
<tr>
<td><strong>Step 4</strong> mtu bytes</td>
<td>(Optional) Sets the maximum transmission unit (MTU) byte value for the interface. The default is 1514.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if# mtu 1448</td>
<td>• The default is 1514 bytes.</td>
</tr>
<tr>
<td></td>
<td>• The range for the Management Ethernet interface Interface <em>mtu</em> values is 64 to 1514 bytes.</td>
</tr>
<tr>
<td><strong>Step 5</strong> no shutdown</td>
<td>Removes the shutdown configuration, which removes the forced administrative down on the interface, enabling it to move to an up or down state.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# no shutdown</td>
<td></td>
</tr>
</tbody>
</table>
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

---

**Example:**

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

**Example:**

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface MgmtEth interface-path-id</td>
<td>Enters interface configuration mode and specifies the Management Ethernet interface name and instance.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> duplex [full</td>
<td>half]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# duplex full</td>
<td><strong>Note</strong> To return the system to autonegotiated duplex operation, use the <strong>no duplex</strong> command.</td>
</tr>
<tr>
<td><strong>Step 4</strong> end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# end</td>
<td>• When you issue the <strong>end</strong> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td>or</td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)?</td>
</tr>
<tr>
<td>commit</td>
<td>[cancel]:</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# commit</td>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td>or</td>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>

Configuring 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}
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 <strong>interface MgmtEth interface-path-id</strong></td>
<td>Enters interface configuration mode and specifies the Management Ethernet interface name and instance.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CP0/0</td>
<td></td>
</tr>
<tr>
<td>Step 3 speed [10</td>
<td>100</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# speed 100</td>
<td><strong>Note</strong> The default Management Ethernet interface speed is autonegotiated.</td>
</tr>
<tr>
<td><strong>Note</strong> To return the system to the default autonegotiated speed, use the no speed command.</td>
<td></td>
</tr>
<tr>
<td>Step 4 <strong>end</strong> or <strong>commit</strong></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>
| **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. |
# 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface MgmtEth interface-path-id</td>
<td>Enters interface configuration mode and specifies the Management Ethernet interface name and instance.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0</td>
<td></td>
</tr>
</tbody>
</table>
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`

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 3 **mac-address** *address* | 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** | 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. |

**Example:**

```
RP/0/RP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD
```

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

```
RP/0/RP0/CPU0:router(config-if)# commit
```
Configuration Examples for Management Ethernet Interfaces

This section provides the following configuration examples:

- Configuring a Management Ethernet Interface: Example, page 78

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/RP0/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/RP0/CPU0/0
MgmtEth0/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 runs, 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/RP0/CPU0/0
interface MgmtEth0/RP0/CPU0/0
   description Connected to Lab LAN
   ipv4 address 172.29.52.70 255.255.255.0
   
```
## Additional References

These sections provide references related to Management Ethernet interface configuration.

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using the Cisco IOS XR software.</td>
<td>Cisco IOS XR Getting Started Guide</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Information about configuring interfaces and other components on the Cisco CRS-1 Router from a remote Craft Works Interface (CWI) client management application</td>
<td>Cisco Craft Works Interface Configuration Guide</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by the feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no applicable MIBs for this module.</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
## Technical Assistance

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

This module describes the configuration of Ethernet interfaces on the Cisco CRS-1 Router.

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 on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.0</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.2</td>
<td>Support was added for the Cisco CRS-1 Router for the SIP-800.</td>
</tr>
<tr>
<td></td>
<td>Support for the 8-Port Gigabit Ethernet SPA was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>Support was added for egress MAC accounting on the 8-port 10-Gigabit Ethernet PLIM.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>The Layer 2 Virtual Private Network (L2VPN) feature was first supported on Ethernet interfaces.</td>
</tr>
<tr>
<td></td>
<td>Support was added for the 8-Port 1-Gigabit Ethernet SPA.</td>
</tr>
<tr>
<td>Release 3.5.0</td>
<td>Support was added on the Cisco CRS-1 Router for the 1-port 10-Gigabit Ethernet WAN SPA.</td>
</tr>
<tr>
<td>Release 4.0.0</td>
<td>Support for the following physical layer interface modules (PLIMs) was added:</td>
</tr>
<tr>
<td></td>
<td>• 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)</td>
</tr>
<tr>
<td></td>
<td>• 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)</td>
</tr>
<tr>
<td>Release 4.0.1</td>
<td>Support for the following PLIMs was added:</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>• 1-Port 100-Gigabit Ethernet PLIM (1X100GBE) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)</td>
<td></td>
</tr>
<tr>
<td>• 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP)</td>
<td></td>
</tr>
<tr>
<td>• 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)</td>
<td></td>
</tr>
<tr>
<td>Release 4.2.3</td>
<td>Support for Link Layer Discovery Protocol (LLDP) was added.</td>
</tr>
<tr>
<td>Release 4.3.2</td>
<td>Support for Egress MAC accounting on bundle Ethernet interfaces was included.</td>
</tr>
</tbody>
</table>
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, page 84
- Gigabit Ethernet Protocol Standards Overview, page 86
- MAC Address, page 87
- MAC Accounting, page 88
- Ethernet MTU, page 88
- Flow Control on Ethernet Interfaces, page 89
- 802.1Q VLAN, page 89
- VRRP, page 89
- HSRP, page 89
- Duplex Mode on Fast Ethernet Interfaces, page 90
- Fast Ethernet Interface Speed, page 90
- LLDP, page 91
- Carrier Delay on Ethernet Interfaces, page 94

Default Configuration Values for Gigabit Ethernet and 10-Gigabit Ethernet

Table 2 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 by entering the `no shutdown` command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration File Entry</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC accounting</td>
<td><code>mac-accounting</code></td>
<td><code>off</code></td>
</tr>
<tr>
<td>Flow control</td>
<td><code>flow-control</code></td>
<td>egress on ingress off</td>
</tr>
</tbody>
</table>
Default Configuration Values for Fast Ethernet

Table 3 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 by entering the `no shutdown` command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration File Entry</th>
<th>Default Value</th>
</tr>
</thead>
</table>
| 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)                                            |

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 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.

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:

- IEEE 802.3 Physical Ethernet Infrastructure, page 87
- IEEE 802.3ab 1000BASE-T Gigabit Ethernet, page 87
- IEEE 802.3z 1000 Mbps Gigabit Ethernet, page 87
- IEEE 802.3ae 10 Gbps Ethernet, page 87
- IEEE 802.3ba 100 Gbps Ethernet, page 87

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.

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

Note

A maximum of 512 MAC addresses for ingress and 512 MAC addresses for egress on each trunk interface are supported.

Egress MAC Accounting

The Cisco CRS Router provides support for Egress MAC accounting on bundle ethernet interfaces too. Egress MAC accounting supports IPv4 unicast traffic over bundles that includes both fast path and slow path traffic.

Limitations of Egress MAC Accounting

These are some of the limitations of Egress MAC Accounting:

- Egress Mac accounting can be run on a maximum of 512 mac accounting entries for each bundle interface.
- Egress Mac accounting is not supported on IPv6 and MPLS interfaces.
- Egress Mac accounting on Bundles is only applicable to Cisco CRS-3 Modular Services Line Cards.
- Egress Mac accounting is not supported if the bundle interface has a combination of Cisco CRS-1 and Cisco CRS-3 member links.
- Egress Mac accounting on bundle sub-interfaces is not supported.
- Egress Mac accounting feature is suspended when you do a process restart of aib_server. You must disable Egress Mac-accounting and re-enable it under the interface configuration mode.

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 *Cisco IOS XR IP Addresses and Services Configuration Guide*.

**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 Cisco IOS XR 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.

Table 4 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>
<thead>
<tr>
<th>duplex Command</th>
<th>speed Command</th>
<th>Resulting System Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>no duplex</td>
<td>no speed</td>
<td>Auto-negotiates both speed and duplex modes.</td>
</tr>
<tr>
<td>no duplex</td>
<td>speed 1000</td>
<td>Auto-negotiates for duplex mode and forces 1000 Mbps.</td>
</tr>
<tr>
<td>no duplex</td>
<td>speed 100</td>
<td>Auto-negotiates for duplex mode and forces 100 Mbps.</td>
</tr>
<tr>
<td>no duplex</td>
<td>speed 10</td>
<td>Auto-negotiates for duplex mode and forces 10 Mbps.</td>
</tr>
<tr>
<td>full-duplex</td>
<td>no speed</td>
<td>Forces full duplex and auto-negotiates for speed.</td>
</tr>
<tr>
<td>full-duplex</td>
<td>speed 1000</td>
<td>Forces full duplex and 1000 Mbps.</td>
</tr>
<tr>
<td>full-duplex</td>
<td>speed 100</td>
<td>Forces full duplex and 100 Mbps.</td>
</tr>
<tr>
<td>full-duplex</td>
<td>speed 10</td>
<td>Forces full duplex and 10 Mbps.</td>
</tr>
<tr>
<td>half-duplex</td>
<td>no speed</td>
<td>Forces half duplex and auto-negotiates for speed.</td>
</tr>
<tr>
<td>half-duplex</td>
<td>speed 1000</td>
<td>Forces half duplex and 1000 Mbps.</td>
</tr>
<tr>
<td>half-duplex</td>
<td>speed 100</td>
<td>Forces half duplex and 100 Mbps.</td>
</tr>
<tr>
<td>half-duplex</td>
<td>speed 10</td>
<td>Forces half duplex and 10 Mbps.</td>
</tr>
</tbody>
</table>

**LLDP**

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 of 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

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are some differences between IPv4 and IPv6 address management in LLDP:</td>
</tr>
<tr>
<td>• For IPv4, as long as the IPv4 address is configured on an interface, it can be used as an LLDP management address.</td>
</tr>
<tr>
<td>• 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.</td>
</tr>
</tbody>
</table>

• 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.

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, page 94
- Configuring LLDP, page 102

Configuring Ethernet Interfaces

This section provides the following configuration procedures:

- Configuring Gigabit Ethernet Interfaces, page 95
- Configuring a L2VPN Ethernet Port, page 100
- Configuring MAC Accounting on an Ethernet Interface, page 98
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` (on Gigabit Ethernet and Fast Ethernet interfaces only)  
10. `no shutdown`  
11. `end`  
    or  
    `commit`  
12. `show interfaces [GigabitEthernet | TenGigE | HundredGigE] interface-path-id`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>show version</code></td>
<td>(Optional) Displays the current software version, and can also be used to confirm that the router recognizes the modular services card.</td>
</tr>
</tbody>
</table>
| **Example:**  
RP/0/RP0/CPU0:router# show version | |
| **Step 2** `show interfaces [GigabitEthernet | TenGigE | HundredGigE] interface-path-id` | (Optional) Displays the configured interface and checks the status of each interface port. |
| **Example:**  
RP/0/RP0/CPU0:router# show interfaces TenGigE 0/1/0/0 | Possible interface types for this procedure are:  
- GigabitEthernet  
- HundredGigE  
- TenGigE |
| **Step 3** `configure` | Enters global configuration mode. |
| **Example:**  
RP/0/RP0/CPU0:router# configure terminal | |
### Command or Action

| Step 4 | interface [GigabitEthernet | TenGigE | HundredGigE] interface-path-id |
|--------|---------------------------------------------------------------|
| Example: | RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0 |

- **Purpose**: 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

<table>
<thead>
<tr>
<th>Step 5</th>
<th>ipv4 address ip-address mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224</td>
</tr>
</tbody>
</table>

- **Purpose**: 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 6 | flow-control {bidirectional | egress | ingress} |
|--------|------------------------------------------------------|
| Example: | RP/0/RP0/CPU0:router(config-if)# flow control ingress |

- **Purpose**: (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.

<table>
<thead>
<tr>
<th>Step 7</th>
<th>mtu bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# mtu 1448</td>
</tr>
</tbody>
</table>

- **Purpose**: (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.

<table>
<thead>
<tr>
<th>Step 8</th>
<th>mac-address value1.value2.value3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# mac address 0001.2468.ABCD</td>
</tr>
</tbody>
</table>

- **Purpose**: (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.

### Step 11
**end**

**Example:**
RP/0/RP0/CPU0:router(config-if)# end

or

**commit**

**Example:**
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.

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.

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. That apart, 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 other common parameters for Ethernet interfaces.

SUMMARY STEPS

1. configure
3. ipv4 address ip-address mask
4. mac-accounting {egress | ingress}
5. end
   or commit
6. show mac-accounting type location instance

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td>Specifies the Physical interface, virtual interface, or bundle-ether interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface [GigabitEthernet</td>
<td>TenGigE</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0</td>
<td>Note Use the show interfaces command to see a list of all interfaces currently configured on the router.</td>
</tr>
<tr>
<td></td>
<td>For more information about the syntax for the router, use the question mark (?) online help function.</td>
</tr>
</tbody>
</table>
## Configuring Ethernet Interfaces on Cisco IOS XR Software

### How to Configure Ethernet

#### Command or Action

<table>
<thead>
<tr>
<th>Step 3</th>
<th><strong>ipv4 address ip-address mask</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224</td>
</tr>
</tbody>
</table>

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:
  - It 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.
  - It can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are 1s, and the corresponding bits of the address are the 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

<table>
<thead>
<tr>
<th><strong>end</strong> or <strong>commit</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th><strong>show mac-accounting type location instance</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
</tr>
</tbody>
</table>

Displays MAC accounting statistics for an interface.
Configuring a L2VPN Ethernet Port

Use this procedure to configure an L2VPN Ethernet port.

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface [GigabitEthernet</td>
<td>TenGigE] interface-path-id</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0</td>
<td>• GigabitEthernet</td>
</tr>
<tr>
<td></td>
<td>• TenGigE</td>
</tr>
<tr>
<td><strong>Step 3</strong> l2transport</td>
<td>Enables Layer 2 transport mode on a port and enter Layer 2 transport configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# l2transport</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4**  
`l2protocol {cdp | pvst | stp | vtp}{[forward | tunnel][experimental bits]|drop}`

**Example:**  
RP/0/RP0/CP00:router(config-if-l2)# l2protocol  
stp tunnel

**Purpose**

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:**  
RP/0/RP0/CP00:router(config-if-l2)# end  
or  
RP/0/RP0/CP00:router(config-if-l2)# commit

**Purpose**

Saves configuration changes.

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

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

**Step 6**  
`show interfaces [GigabitEthernet | TenGigE]  
interface-path-id`

**Example:**  
RP/0/RP0/CP00:router# show interfaces TenGigE  
0/3/0/0

**Purpose**

(Optional) Displays statistics for interfaces on the router.
What to Do Next

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.

Configuring LLDP

This section includes the following configuration topics for LLDP:

- LLDP Default Configuration, page 102
- Enabling LLDP Globally, page 102 (required)
- Configuring Global LLDP Operational Characteristics, page 103 (optional)
- Disabling Transmission of Optional LLDP TLVs, page 105 (optional)
- Disabling LLDP Receive and Transmit Operation for an Interface, page 106 (optional)
- Verifying the LLDP Configuration, page 108

LLDP Default Configuration

Table 5 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.

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

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” section on page 106.

To enable LLDP globally, complete the following steps:
SUMMARY STEPS

1. configure
2. lldp
3. end
   or
   commit

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>lldp</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# lldp</td>
</tr>
<tr>
<td>Enables LLDP globally for both transmit and receive operation on the system.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>end</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# end</td>
</tr>
<tr>
<td>Saves configuration changes.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>commit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# commit</td>
</tr>
<tr>
<td>• When you issue the <strong>end</strong> command, the system prompts you to commit changes:</td>
<td></td>
</tr>
<tr>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Global LLDP Operational Characteristics

The “LLDP Default Configuration” section on page 102 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

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

(Optional) Specifies the LLDP packet rate.

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 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 | TenGigE] interface-path-id`

---

### Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>lldp tlv-select tlv-name disable</code></td>
<td>(Optional) Specifies that transmission of the selected TLV in LLDP packets is disabled. The <em>tlv-name</em> can be one of the following LLDP TLV types:</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config)# lldp tlv-select system-capabilities disable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config)# end</code> or <code>RP/0/RP0/CPU0:router(config)# commit</code></td>
<td>When you issue the <em>end</em> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering <em>yes</em> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering <em>no</em> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering <em>cancel</em> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <em>commit</em> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
### Steps to Configure Ethernet Interfaces

1. **configure**
   
   Example:
   ```
   RP/0/RP0/CPU0:router# configure
   ```
   
   Enters global configuration mode.

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

3. **lldp**
   
   Example:
   ```
   RP/0/RP0/CPU0:router(config-if)# lldp
   ```
   
   (Optional) Enters LLDP configuration mode for the specified interface.

4. **receive disable**
   
   Example:
   ```
   RP/0/RP0/CPU0:router(config-if)# receive disable
   ```
   
   (Optional) Disables LLDP receive operations on the interface.

5. **transmit disable**

6. **end**

   or

   **commit**
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/RP0/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/RP0/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
```

What To Do Next

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear lldp</td>
<td>Resets LLDP traffic counters or LLDP neighbor information.</td>
</tr>
<tr>
<td>show lldp entry</td>
<td>Displays detailed information about LLDP neighbors.</td>
</tr>
<tr>
<td>show lldp errors</td>
<td>Displays LLDP error and overflow statistics.</td>
</tr>
<tr>
<td>show lldp neighbors</td>
<td>Displays information about LLDP neighbors.</td>
</tr>
<tr>
<td>show lldp traffic</td>
<td>Displays statistics for LLDP traffic.</td>
</tr>
</tbody>
</table>

Configuration Examples for Ethernet

This section provides the following configuration examples:
- Configuring an Ethernet Interface: Example, page 109
- Configuring a Fast Ethernet Interface: Example, page 110
- Configuring MAC-Accounting: Example, page 111
- Configuring a Layer 2 VPN AC: Example, page 111
- Configuring LLDP: Examples, page 111

Configuring an Ethernet Interface: Example

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

```bash
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
```
Configuring Ethernet Interfaces on Cisco IOS XR Software

Configuration Examples for Ethernet

Configuring a Fast Ethernet Interface: Example

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

```
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
```

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)## interface fastethernet 0/0/2/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.30.1.2 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# duplex full
RP/0/RP0/CPU0:router(config-if)# mtu 1514
RP/0/RP0/CPU0:router(config-if)# speed 100
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 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

The following 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 pvst tunnel
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 Cisco IOS XR Software module in this document.

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

Additional References

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

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.1ag</td>
<td>—</td>
</tr>
<tr>
<td>ITU-T Y.1731</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

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

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

**Technical Assistance**

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

This module describes the configuration of Ethernet Operations, Administration, and Maintenance (OAM) on the Cisco CRS-1 Router.

Feature History for Configuring Ethernet OAM

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.9.0</td>
<td>Support for the following features was introduced:</td>
</tr>
<tr>
<td></td>
<td>• Ethernet Link OAM</td>
</tr>
<tr>
<td></td>
<td>• Ethernet CFM</td>
</tr>
<tr>
<td>Release 3.9.1</td>
<td>Support for the following features was introduced:</td>
</tr>
<tr>
<td></td>
<td>• EFD</td>
</tr>
<tr>
<td></td>
<td>• AIS</td>
</tr>
<tr>
<td></td>
<td>• The ethernet cfm mep domain command is replaced by the ethernet cfm and mep domain commands.</td>
</tr>
</tbody>
</table>
Release 4.0.0  Support for the following features was introduced:

- The **action link-fault** command is replaced by the **action uni-directional link fault** command.
- The **efd** keyword is added 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.

<table>
<thead>
<tr>
<th>Release 4.1.0</th>
<th>Support for CFM Y.1731 ITU Carrier Code (ICC)-based MEG ID (MAID) format was introduced.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.3.0</td>
<td>Support for ITU-T Y.1731 Synthetic Loss Measurement was introduced.</td>
</tr>
<tr>
<td>Release 4.3.1</td>
<td>Support for ITU-T Y.1731 Loss Measurement was introduced.</td>
</tr>
<tr>
<td>Release 5.1.2</td>
<td>Support for Ethernet CFM down MEP on VRF interfaces was introduced.</td>
</tr>
<tr>
<td>Release 5.1.3</td>
<td>Support for CFM over Bundles was introduced.</td>
</tr>
</tbody>
</table>
Prerequisites for Configuring Ethernet OAM

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

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

1. 4-Port Gigabit Ethernet physical layer interface module (PLIM)
2. 8-Port 10-Gigabit Ethernet PLIM
3. 4-Port 10-Gigabit Ethernet PLIM
4. 2-Port Gigabit Ethernet SPA
5. 5-Port Gigabit Ethernet SPA
6. 8-Port Gigabit Ethernet SPA
7. 10-Port Gigabit Ethernet SPA
8. 1-Port 10-Gigabit Ethernet SPA
9. 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:

1. Hello interval configuration
2. Remote loopback
3. 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**, page 118
- **Ethernet CFM**, page 119
- **Ethernet SLA**, page 136

**Ethernet Link OAM**

Ethernet as a Metro Area Network (MAN) or a Wide Area Network (WAN) technology benefits greatly from the implementation of Operations, Administration and Maintenance (OAM) features. Ethernet link OAM features allow Service Providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events, take actions on events, and if necessary, put specific interfaces into loopback mode for troubleshooting. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet link OAM can be configured in the following ways:

- A Link OAM profile can be configured, and this profile can be used to set the parameters for multiple interfaces.
- Link OAM can be configured directly on an interface.
  
  When an interface is also using a link OAM profile, specific parameters that are set in the profile can be overridden by configuring a different value directly on the interface.

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

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

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

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

- **Neighbor Discovery**, page 118
- **Link Monitoring**, page 119
- **MIB Retrieval**, page 119
- **Miswiring Detection (Cisco-Proprietary)**, page 119
- **SNMP Traps**, page 119

**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-1 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 inter-operable.

- 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” section on page 136.
To understand how the CFM maintenance model works, you need to understand the following concepts and features:

- Maintenance Domains, page 120
- Services, page 122
- Maintenance Points, page 122
- CFM Protocol Messages, page 125
- MEP Cross-Check, page 133
- Configurable Logging, page 134
- EFD, page 134

**Maintenance Domains**

A *maintenance domain* describes a management space for the purpose of managing and administering a network. A domain is owned and operated by a single entity and defined by the set of interfaces internal to it and at its boundary, as shown in Figure 1.

*Figure 1  CFM Maintenance Domain*

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

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

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

Each organization uses a different CFM maintenance domain. Figure 2 shows an example of the different levels of maintenance domains in a network.

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” section on page 122.

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

**Figure 3**  
Supported CFM Maintenance Domain Structure

![Diagram of CFM Maintenance Domains](image)

**Services**

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

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

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

**Maintenance Points**

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

A maintenance point is always associated with a particular CFM service, and therefore with a particular maintenance domain at a particular level. Maintenance points generally only process CFM frames at the same level as their associated maintenance domain. Frames at a higher maintenance level are always forwarded transparently, while frames at a lower maintenance level are normally dropped. This helps enforce the maintenance domain hierarchy described in the “Maintenance Domains” section on page 120, 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
Configuring Ethernet OAM on Cisco IOS XR Software

Information About Configuring Ethernet OAM

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.
Figure 4 illustrates the monitored areas for Down and Up MEPs.

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

Figure 5 shows maintenance points at different levels. Because domains are allowed to nest but not intersect (see Figure 3), a MEP at a low level always corresponds with a MEP or MIP at a higher level. In addition, only a single MIP is allowed on any interface—this is generally created in the lowest domain that exists at the interface and that does not have a MEP.

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

MIPs and Up MEPs can only exist on switched (Layer 2) interfaces, because they send and receive frames from the bridge relay function. Down MEPs can be created on switched (Layer 2) or routed (Layer 3) interfaces.
MEPs continue to operate normally if the interface they are created on is blocked by the Spanning Tree Protocol (STP); that is, CFM frames at the level of the MEP continue to be sent and received, according to the direction of the MEP. MEPs never allow CFM frames at the level of the MEP to be forwarded, so the STP block is maintained.

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

Note

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

CFM Protocol Messages

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

This section describes the following CFM messages:

- Continuity Check (IEEE 802.1ag and ITU-T Y.1731), page 125
- Loopback (IEEE 802.1ag and ITU-T Y.1731), page 127
- Linktrace (IEEE 802.1ag and ITU-T Y.1731), page 128
- Explorer Linktrace (Cisco), page 130
- Alarm Indication Signal (ITU-T Y.1731), page 131
- Delay and Jitter Measurement (ITU-T Y.1731), page 132
- Synthetic Loss Measurement (ITU-T Y.1731), page 132
- Loss Measurement (ITU-T Y.1731), page 133

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

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

MIPs also receive CCMs. MIPs use the information to build a MAC learning database that is used when responding to Linktrace. For more information about Linktrace, see the “Linktrace (IEEE 802.1ag and ITU-T Y.1731)” section on page 128.
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
- 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.
Figure 7 shows an example of CFM loopback message flow between a MEP and MIP.

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

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

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. LTM are multicast, and carry the unicast target MAC address as data within the frame. They are intercepted at each hop where there is a maintenance point, and either retransmitted or dropped to discover the unicast path to the target MAC address.
Figure 8 shows an example of CFM linktrace message flow between MEPs and MIPs.

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

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

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

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

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

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

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

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

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

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

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

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

![Exploratory Linktrace Messages and Replies](image)

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

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

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

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

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

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

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

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

- Detection of CCM defects, as described “Continuity Check (IEEE 802.1ag and ITU-T Y.1731)” section on page 125.
Information About Configuring Ethernet OAM

- Loss of continuity.
- Receipt of AIS messages.
- Failure in the underlying transport, such as when an interface is down.

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

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

The router supports one-way and two-way delay measurement using two packet types:
- Delay Measurement Message (DMM)
- Delay Measurement Response (DMR)

These packets are unicast similar to loopback messages. The packets carry timestamps generated by the system time-of-day clock to support more accurate delay measurement, and also support an SLA manageability front-end.

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” section on page 136.

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 referred to as the source-to-destination (sd) sequence number and the destination-to-source (ds) sequence number respectively.

Figure 11 shows an example of how the sequence numbers are used to calculate the Frame Loss Ratio (FLR) in each direction.
**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.
Figure 12 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.

**CFM over Bundles**

CFM is supported on bundle interfaces, bundle sub-interfaces and bundle member interfaces with the bundle members spanning only on the Modular SC 40G and Modular SC 140G Line Cards.

**Restrictions for Configuration of CFM on Bundles**

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

- Only Layer 2 and Layer 3 bundle Ethernet interfaces and sub-interfaces are supported except for those matching VLAN tag ‘any’.
- No support for UP MEP over L2TPv3 core.
- No support for UP MEP on bridge-domain attachment circuits.
- Only CFM Level-0 must be configured on the bundle member interface.
- Any level (0 – 7) is supported on bundle sub-interfaces.
- CFM is not supported on mixed bundles with MSC-X links.
- The maximum scale supported is 2000 sessions with a minimum CCM interval of 1 second.
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

**Loss Measurement**: Loss Measurement is an extension to the existing Ethernet SLA feature; it adds the functionality for loss measurement defined in the Y.1731 and G.8021 ITU-T standards. This is used to accurately measure the loss of data traffic, by exchanging CFM frames containing sent and received frame counters. It is also used to measure the availability of the network by tracking periods of high loss over time. Loss Measurement messages can be used to perform these measurements:

- Data packet loss
- SLA probe packet loss
- Out of order SLA Probe packet count
- SLA Probe Packet corruption count

**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 targeted 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:

\[
\frac{<\text{num\_sent}> - <\text{num\_rcvd}> }{<\text{num\_sent}>}
\]

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:

- Ethernet SLA Statistic, page 139
- Ethernet SLA Measurement Packet, page 140
- Ethernet SLA Sample, page 140
- Ethernet SLA Probe, page 140
- Ethernet SLA Burst, page 141
- Ethernet SLA Schedule, page 141
- Ethernet SLA Bucket, page 141
- Ethernet SLA Aggregation Bin, page 141
- Ethernet SLA Operation Profile, page 141
- Ethernet SLA Operation, page 141
- Ethernet SLA On-Demand Operation, page 142

**Ethernet SLA Statistic**

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 referred 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.

Ethernet SLA Schedule

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

Ethernet SLA Bucket

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

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

Ethernet SLA Aggregation Bin

Rather than storing each sample separately within a bucket, an alternative is to aggregate the samples into bins. An aggregation bin is a range of sample values, and contains a counter of the number of samples that were received that fall within that range. The set of bins forms a histogram. When aggregation is enabled, each bucket contains a separate set of bins. See Figure 13 on page 201.

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.
Information About Configuring Ethernet OAM

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 FRU 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” section on page 172.

---

**How to Configure Ethernet OAM**

This section provides these configuration procedures:

- Configuring Ethernet Link OAM, page 143
- Configuring Ethernet CFM, page 152
- Configuring Ethernet SLA, page 172

**Configuring Ethernet Link OAM**

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

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

- Configuring an Ethernet OAM Profile, page 144
Configuring an Ethernet OAM Profile

Perform the following steps to configure an Ethernet OAM profile.

**SUMMARY STEPS**

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet oam profile profile-name</td>
<td>Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# ethernet oam profile Profile_1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> link-monitor</td>
<td>Enters the Ethernet OAM link monitor configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# link-monitor</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> symbol-period window window</td>
<td>(Optional) Configures the window size (in milliseconds) for an Ethernet OAM symbol-period error event. The range is 1000 to 60000. The default value is 1000.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period window 60000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> symbol-period threshold low threshold high</td>
<td>(Optional) Configures the thresholds (in symbols) that trigger an Ethernet OAM symbol-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is 0 to 60000000. The default low threshold is 1.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period threshold low 1000000 high 60000000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> frame window window</td>
<td>(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.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# frame window 60</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> frame threshold low threshold high</td>
<td>(Optional) Configures the thresholds (in symbols) that triggers an Ethernet OAM frame error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is from 0 to 60000000. The default low threshold is 1.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> frame-period window window</td>
<td>(Optional) Configures the window size (in milliseconds) for an Ethernet OAM frame-period error event. The range is from 100 to 60000. The default value is 1000.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window 60000</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Ethernet OAM on Cisco IOS XR Software

#### How to Configure Ethernet OAM

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>frame-period threshold low</strong> <strong>threshold high</strong> <strong>threshold</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0# config-eoam-lm#</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>frame-period threshold low 100 high 1000000</code></td>
</tr>
<tr>
<td>(Optional)</td>
<td>Configures the thresholds (in errors per million frames) that trigger an Ethernet OAM frame-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold.</td>
</tr>
<tr>
<td>The range is from 0 to 1000000.</td>
<td></td>
</tr>
<tr>
<td>The default low threshold is 60000.</td>
<td></td>
</tr>
<tr>
<td>The IEEE 802.3 standard defines threshold crossing events as number of error frames in a window.</td>
<td></td>
</tr>
<tr>
<td>To comply with the standards, the low and high threshold for frame-period events is measured in errors per million frames.</td>
<td></td>
</tr>
<tr>
<td>Hence, the calculation to determine the remote low and high threshold is (configured threshold * frame window in received Bridge Protocol Data Unit (BPDU))/1000000. For example, if the received frame window=300, then high threshold is 20000 * 300 / 1000000 yielding the result 6.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>frame-seconds window</strong> <strong>window</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0# config-eoam-lm#</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>frame-seconds window 900000</code></td>
</tr>
<tr>
<td>(Optional)</td>
<td>Configures the window size (in milliseconds) for the OAM frame-seconds error event.</td>
</tr>
<tr>
<td>The range is 10000 to 900000.</td>
<td></td>
</tr>
<tr>
<td>The default value is 6000.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>frame-seconds threshold low</strong> <strong>threshold high</strong> <strong>threshold</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0# config-eoam-lm#</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>frame-seconds threshold low 3 threshold high 900</code></td>
</tr>
<tr>
<td>(Optional)</td>
<td>Configures the thresholds (in seconds) that trigger a frame-seconds error event. The high threshold value can be configured only in conjunction with the low threshold value.</td>
</tr>
<tr>
<td>The range is 1 to 900.</td>
<td></td>
</tr>
<tr>
<td>The default value is 1.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0# config-eoam-lm#</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>exit</code></td>
</tr>
<tr>
<td>Exits back to Ethernet OAM mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>mib-retrieval</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0# config-eoam-lm#</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>mib-retrieval</code></td>
</tr>
<tr>
<td>Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>connection timeout</strong> <strong>&lt;timeout&gt;</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0# config-eoam-lm#</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>connection timeout 30</code></td>
</tr>
<tr>
<td>Configures the connection timeout period for an Ethernet OAM session, as a multiple of the hello interval.</td>
<td></td>
</tr>
<tr>
<td>The range is 2 to 30.</td>
<td></td>
</tr>
<tr>
<td>The default value is 5.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td><strong>mode</strong> **{active</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0# config-eoam-lm#</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>mode passive</code></td>
</tr>
<tr>
<td>Configures the Ethernet OAM mode. The default is active.</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>16</td>
<td>`require-remote mode {active</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-eoam)# require-remote mode active</td>
</tr>
<tr>
<td>17</td>
<td><code>require-remote link-monitoring</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-eoam)# require-remote link-monitoring</td>
</tr>
<tr>
<td>18</td>
<td><code>require-remote mib-retrieval</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval</td>
</tr>
<tr>
<td>19</td>
<td>`action capabilities-conflict {disable</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-eoam)# action capabilities-conflict efd</td>
</tr>
<tr>
<td>20</td>
<td>`action critical-event {disable</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-eoam)# action critical-event error-disable-interface</td>
</tr>
<tr>
<td>21</td>
<td>`action discovery-timeout {disable</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-eoam)# action discovery-timeout efd</td>
</tr>
<tr>
<td>22</td>
<td>`action dying-gasp {disable</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface</td>
</tr>
</tbody>
</table>
# How to Configure Ethernet OAM

## Command or Action

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

<table>
<thead>
<tr>
<th>Step 24</th>
<th>action remote-loopback disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-eoam)# action remote-loopback disable</td>
</tr>
<tr>
<td>Purpose:</td>
<td>Specifies that no action is taken on an interface when a remote-loopback event occurs. The default action is to create a syslog entry.</td>
</tr>
<tr>
<td>Note:</td>
<td>If you change the default, the <strong>log</strong> keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.</td>
</tr>
</tbody>
</table>

| Step 25 | action session-down {disable | efd | error-disable-interface} |
|---------|-----------------------------|
| Example: | RP/0/RP0/CPU0:router(config-eoam)# action session-down efd |
| Purpose: | Specifies the action that is taken on an interface when an Ethernet OAM session goes down. |
| Note: | If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs. |

<table>
<thead>
<tr>
<th>Step 26</th>
<th>action session-up disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-eoam)# action session-up disable</td>
</tr>
<tr>
<td>Purpose:</td>
<td>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.</td>
</tr>
<tr>
<td>Note:</td>
<td>If you change the default, the <strong>log</strong> keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.</td>
</tr>
</tbody>
</table>

| Step 27 | action uni-directional link-fault {disable | efd | error-disable-interface} |
|---------|-------------------------------|
| Example: | RP/0/RP0/CPU0:router(config-eoam)# action session-down efd |
| Purpose: | Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. |
| Note: | If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs. |
| Note: | In Cisco IOS XR Release 4.x, this command replaces the **action link-fault** command. |

| Step 28 | action wiring-conflict {disable | efd | log} |
|---------|---------------------------------------------|
| Example: | RP/0/RP0/CPU0:router(config-eoam)# action session-down efd |
| Purpose: | Specifies the action that is taken on an interface when a wiring-conflict event occurs. The default is to put the interface into error-disable state. |
| Note: | If you change the default, the **error-disable-interface** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and put the interface into error-disable state when the event occurs. |
## Attaching an Ethernet OAM Profile to an Interface

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

### SUMMARY STEPS

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

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure terminal</td>
</tr>
<tr>
<td>Step 2 interface</td>
<td>Enters interface configuration mode and specifies the Ethernet interface name and notation rack/slot/module/port.</td>
</tr>
<tr>
<td>[GigabitEthernet</td>
<td>Note: The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.</td>
</tr>
<tr>
<td></td>
<td>TenGigE] interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0</td>
</tr>
<tr>
<td>Step 3 ethernet oam</td>
<td>Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ethernet oam</td>
</tr>
<tr>
<td>Step 4 profile</td>
<td>Attaches the specified Ethernet OAM profile (profile-name), and all of its configuration, to the interface.</td>
</tr>
<tr>
<td>profile-name</td>
<td>Example: RP/0/RP0/CPU0:router(config-if-eoam)# profile Profile_1</td>
</tr>
</tbody>
</table>
### Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

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

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

To see all of the default Ethernet OAM configuration settings, see the “Verifying the Ethernet OAM Configuration” section on page 151.

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

### SUMMARY STEPS

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> end</td>
<td>Ends the configuration session and exits to the EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>
**DETAILED STEPS**

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

**Verifying the Ethernet OAM Configuration**

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

```plaintext
RP/0/RP0/CPU0:router# show ethernet oam configuration
Thu Aug  5 22:07:06.870 DST
```

**Note** 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
Configuring Ethernet OAM

GigabitEthernet0/4/0/0:
Hello interval: 1s
Link monitoring enabled: Y
Remote loopback enabled: N
Mib retrieval enabled: N
Uni-directional link-fault detection enabled: N
Configured mode: Active
Connection timeout: 5
Symbol period window: 0
Symbol period low threshold: 1
Symbol period high threshold: None
Frame window: 1000
Frame low threshold: 1
Frame high threshold: None
Frame period window: 1000
Frame period low threshold: 1
Frame period high threshold: None
Frame seconds window: 60000
Frame seconds low threshold: 1
Frame seconds high threshold: None
High threshold action: None
Link fault action: Log
Dying gasp action: Log
Critical event action: Log
Discovery timeout action: Log
Capabilities conflict action: Log
Wiring conflict action: Error-Disable
Session up action: Log
Session down action: Log
Remote loopback action: Log
Require remote mode: Ignore
Require remote MIB retrieval: N
Require remote loopback support: N
Require remote link monitoring: N

Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:

- Configuring a CFM Maintenance Domain, page 153 (required)
- Configuring Services for a CFM Maintenance Domain, page 154 (required)
- Enabling and Configuring Continuity Check for a CFM Service, page 156 (optional)
- Configuring Automatic MIP Creation for a CFM Service, page 158 (optional)
- Configuring Cross-Check on a MEP for a CFM Service, page 159 (optional)
- Configuring Other Options for a CFM Service, page 161 (optional)
- Configuring CFM MEPs, page 163 (required)
- Configuring Y.1731 AIS, page 165 (optional)
- Configuring EFD for a CFM Service, page 169 (optional)
- Verifying the CFM Configuration, page 171
- Troubleshooting Tips, page 171
Configuring a CFM Maintenance Domain

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 ethernet cfm</td>
<td>Enters Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
</tbody>
</table>
| Step 3 domain domain-name level level-value [id [null] dns DNS-name] [mac H.H.H] [string string] } | 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. |
| Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1 | |
### Command or Action

<table>
<thead>
<tr>
<th>Step 4</th>
<th>traceroute cache hold-time minutes size entries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000</td>
</tr>
</tbody>
</table>

(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.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>end or commit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# commit</td>
</tr>
</tbody>
</table>

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 Services for a CFM Maintenance Domain

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

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

**SUMMARY STEPS**

1. configure
2. ethernet cfm
3. domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]
4. service service-name [down-meps | xconnect group xconnect-group-name p2p xconnect-name][id [string text] | [number number] | [vlan-id id-number] | [vpn-id oui-vpnid]]
5. end or commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet cfm</td>
<td>Enters Ethernet CFM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]</td>
<td>Creates and names a container for all domain configurations at a specified maintenance level, and enters CFM domain configuration mode. The <em>id</em> is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service service-name {down-meps</td>
<td>xconnect group xconnect-group-name ptp xconnect-name} [id [string text]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> When you use the <code>end</code> command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
<td></td>
</tr>
<tr>
<td>– Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>– Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>– Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>
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 [down-meps | xconnect 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

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

**Step 4**
```
service service-name {down-meps | xconnect 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
```

**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
```

**Step 6**
```
continuity-check archive hold-time minutes
```

**Example:**
```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100
```

**Step 7**
```
continuity-check loss auto-traceroute
```

**Example:**
```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute
```

**Step 8**
```
end or commit
```

**Example:**
```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit
```

### Purpose

- **Step 4** 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** (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** (Optional) Configures how long information about peer MEPs is stored after they have timed out.

- **Step 7** (Optional) Configures automatic triggering of a traceroute when a MEP is declared down.

- **Step 8** Saves configuration changes.
  - When you use the `end` command, the system prompts you to commit changes:
    
    Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
    
    - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
    
    - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
    
    - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.
  - Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring Automatic MIP Creation for a CFM Service

For more information about the algorithm for creating MIPs, see the “MIP Creation” section on page 123.

To configure automatic MIP creation for a CFM service, complete the following steps:

SUMMARY STEPS

1. configure
2. ethernet cfm
3. domain domain-name level level-value [id [null] | [dns DNS-name] | [mac H.H.H] | [string string]]
4. service service-name [down-meps | xconnect group xconnect-group-name p2p xconnect-name] [id [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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet cfm</td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain domain-name level level-value [id [null]</td>
<td>[dns DNS-name]</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service service-name [down-meps</td>
<td>xconnect group xconnect-group-name p2p xconnect-name] [id [string text]</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
</tbody>
</table>
To configure cross-check on a MEP for a CFM service and specify the expected set of MEPs, complete the following steps:

**SUMMARY STEPS**

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

---

**Configuring Cross-Check on a MEP for a CFM Service**

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

**SUMMARY STEPS**

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

---

**Command or Action** | **Purpose**
--- | ---
**Step 5** | **Command or Action** | **Purpose**
--- | ---
mip auto-create {all | lower-mep-only} | (Optional) Enables the automatic creation of MIPs in an xconnect.

**Example:**

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

**Step 6**

```
end
```

or

```
commit
```

**Example:**

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

---

**Configuring Cross-Check on a MEP for a CFM Service**

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

**SUMMARY STEPS**

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

---

**Command or Action** | **Purpose**
--- | ---
**Step 5** | **Command or Action** | **Purpose**
--- | ---
mip auto-create {all | lower-mep-only} | (Optional) Enables the automatic creation of MIPs in an xconnect.

**Example:**

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

**Step 6**

```
end
```

or

```
commit
```

**Example:**

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

---

**Configuring Cross-Check on a MEP for a CFM Service**

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

**SUMMARY STEPS**

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

---

**Command or Action** | **Purpose**
--- | ---
**Step 5** | **Command or Action** | **Purpose**
--- | ---
mip auto-create {all | lower-mep-only} | (Optional) Enables the automatic creation of MIPs in an xconnect.

**Example:**

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

**Step 6**

```
end
```

or

```
commit
```

**Example:**

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

---

**Configuring Cross-Check on a MEP for a CFM Service**

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

**SUMMARY STEPS**

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

---

**Command or Action** | **Purpose**
--- | ---
**Step 5** | **Command or Action** | **Purpose**
--- | ---
mip auto-create {all | lower-mep-only} | (Optional) Enables the automatic creation of MIPs in an xconnect.

**Example:**

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

**Step 6**

```
end
```

or

```
commit
```

**Example:**

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

---

**Configuring Cross-Check on a MEP for a CFM Service**

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

**SUMMARY STEPS**

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

---

**Command or Action** | **Purpose**
--- | ---
**Step 5** | **Command or Action** | **Purpose**
--- | ---
mip auto-create {all | lower-mep-only} | (Optional) Enables the automatic creation of MIPs in an xconnect.

**Example:**

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

**Step 6**

```
end
```

or

```
commit
```

**Example:**

```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit
```
## How to Configure Ethernet OAM

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet cfm</td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]</td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The <strong>id</strong> is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service service-name {down-meps</td>
<td>xconnect group xconnect-group-name p2p xconnect-name}[id [string text]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> mep crosscheck</td>
<td>Enters CFM MEP crosscheck configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Other Options for a CFM Service

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

**SUMMARY STEPS**

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

---

### Command or Action

**Step 6**

**mep-id** mep-id-number [mac-address mac-address]

**Example:**

RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> mep-id mep-id-number [mac-address mac-address]</td>
<td>Enables cross-check on a MEP.</td>
</tr>
<tr>
<td>Note Repeat this command for every MEP that you want included in the expected set of MEPs for cross-check.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>• When you use the <strong>end</strong> command, the system prompts you to commit changes:</td>
<td></td>
</tr>
<tr>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)?</td>
<td></td>
</tr>
<tr>
<td>[cancel]:</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Other Options for a CFM Service

To configure other options for a CFM service, complete the following steps:
# How to Configure Ethernet OAM

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet cfm</td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]</td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The <strong>id</strong> is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service service-name {down-meps</td>
<td>xconnect group xconnect-group-name p2p xconnect-name}[id [string text]</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> maximum-meps number</td>
<td>(Optional) Configures the maximum number (2 to 8190) of MEPs across the network, which limits the number of peer MEPs recorded in the database.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
| **Step 6** | **log** {ais|continuity-check errors|continuity-check mep changes|crosscheck errors|efd}
**Example:**
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors
| | *(Optional) Enables logging of certain types of events.*
| **Step 7** | **end**
or
**commit**
**Example:**
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit
| | Saves configuration changes.
- When you use the **end** command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

### Configuring CFM MEPs
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.

### SUMMARY STEPS
1. **configure**
2. **interface** {GigabitEthernet | TenGigE} **interface-path-id**
3. **ethernet cfm**
4. **mep domain** **domain-name** **service** **service-name** **mep-id** **id-number**
5. **cos** **cos**
6. **end**
or
**commit**
## How to Configure Ethernet OAM

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface</td>
<td>Type of Ethernet interface on which you want to create a MEP. Enter GigabitEthernet or TenGigE and the physical interface or virtual interface.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface</td>
</tr>
<tr>
<td></td>
<td>gigabitethernet 0/1/0/1</td>
</tr>
<tr>
<td><strong>Step 3</strong> ethernet cfm</td>
<td>Enters interface Ethernet CFM configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# ethernet cfm</td>
</tr>
<tr>
<td><strong>Step 4</strong> mep domain</td>
<td>Creates a maintenance end point (MEP) on an interface and enters interface CFM MEP configuration mode.</td>
</tr>
<tr>
<td>domain domain-name</td>
<td></td>
</tr>
<tr>
<td>service service-name</td>
<td></td>
</tr>
<tr>
<td>mep-id id-number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if-cfm)# mep domain</td>
</tr>
<tr>
<td></td>
<td>Dm1 service Sv1 mep-id 1</td>
</tr>
</tbody>
</table>
How to Configure Ethernet OAM

Configuring Y.1731 AIS

This section has the following step procedures:

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

Configuring AIS in a CFM Domain Service

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

SUMMARY STEPS

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 5</td>
<td>cos cos</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7</td>
</tr>
<tr>
<td>Step 6</td>
<td>end</td>
</tr>
<tr>
<td>or</td>
<td>commit</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit</td>
</tr>
</tbody>
</table>

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

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

– Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
– Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
– Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet cfm</td>
<td>Enters Ethernet CFM global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain name level level</td>
<td>Specifies the domain and domain level.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service name xconnect group xconnect-group-name p2p xconnect-name</td>
<td>Specifies the service and cross-connect group and name.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service B1 bridge group BG1 bridge-domain BD2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ais transmission [interval {1s</td>
<td>1m}] [cos cos]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure Ethernet OAM

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6 log ais</td>
<td>Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais</td>
<td></td>
</tr>
<tr>
<td>Step 7 end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit</td>
<td></td>
</tr>
</tbody>
</table>

- When you issue the **end** command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
## Configuring Ethernet OAM on Cisco IOS XR Software

### How to Configure Ethernet OAM

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface gigabitethernet interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# interface gigabitethernet 0/1/0/2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ethernet cfm</td>
<td>Enters Ethernet CFM interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ais transmission up interval 1m cos cos</td>
<td>Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit</td>
<td></td>
</tr>
</tbody>
</table>

- When you issue the **end** command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring EFD for a CFM Service

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

Restrictions

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

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 ethernet cfm</td>
<td>Enters CFM configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td>Step 3 domain domain-name level level-value</td>
<td>Specifies or creates the CFM domain and enters CFM domain configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm-dmn)# domain D1 level 1</td>
<td></td>
</tr>
<tr>
<td>Step 4 service service-name down-meps</td>
<td>Specifies or creates the CFM service for down MEPS and enters CFM domain service configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 down-meps</td>
<td></td>
</tr>
<tr>
<td>Step 5 efd</td>
<td>Enables EFD on all down MEPs in the down MEPS service.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# efd</td>
<td></td>
</tr>
</tbody>
</table>
### How to Configure Ethernet OAM

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 6**
   
   `log efd`
   
   **Example:**
   
   `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log efd`
   
   (Optional) Enables logging of EFD state changes on an interface.
| **Step 7**
   
   `end`
   
   or
   
   `commit`
   
   **Example:**
   
   `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`
   
   Saves configuration changes.
   
   - When you issue the `end` command, the system prompts you to commit changes:
     
     Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
     
     - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
     - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
     - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.
   
   - Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Verifying the EFD Configuration

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

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

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

Troubleshooting Tips

To troubleshoot problems within the CFM network, perform the following 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# 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# traceroute ethernet cfm domain D1 service S1 mep-id 16 source interface GigabitEthernet 0/0/0/0
```

Traceroutes in domain D1 (level 4), service S1
Source: MEP-ID 1, interface GigabitEthernet0/0/0/0
Traceroute at 2009-05-18 12:09:10 to 0001.0203.0402, TTL 64, Trans ID 2:

<table>
<thead>
<tr>
<th>Hop</th>
<th>Hostname/Last</th>
<th>Ingress MAC/name</th>
<th>Egress MAC/name</th>
<th>Relay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ios</td>
<td>0001.0203.0400</td>
<td>[Down]</td>
<td>FDB</td>
</tr>
<tr>
<td></td>
<td>0000-0001.0203.0400</td>
<td>Gi0/0/0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>abc</td>
<td></td>
<td>0001.0203.0401</td>
<td>FDB</td>
</tr>
<tr>
<td></td>
<td>ios</td>
<td></td>
<td>[Ok]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>bcd</td>
<td>0001.0203.0402</td>
<td>[Ok]</td>
<td>Hit</td>
</tr>
<tr>
<td></td>
<td>abc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GigE0/0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Replies dropped: 0

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

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

Configuring Ethernet SLA

This section describes how to configure Ethernet SLA.

Ethernet SLA Configuration Guidelines

⚠️ Caution

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

Before you configure Ethernet SLA, consider the following guidelines:

- **Aggregation**—Use of the `aggregate none` command significantly increases the amount of memory required because each individual measurement is recorded, rather than just counts for each aggregation bin. When you configure aggregation, consider that more bins will require more memory.

- **Buckets archive**—When you configure the `buckets archive` command, consider that the more history that is kept, the more memory will be used.

- **Measuring two statistics (such as both delay and jitter)** will use approximately twice as much memory as measuring one.

- **Separate statistics** are stored for one-way source-to-destination and destination-to-source measurements, which consumes twice as much memory as storing a single set of round-trip statistics.

- **The Cisco CRS-1 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.
The following procedure provides the steps to configure Ethernet Service Level Agreement (SLA) monitoring at Layer 2.

To configure SLA, perform the following tasks:

- Configuring an SLA Operation Profile, page 173
- Configuring a Schedule for an SLA Operation Probe in a Profile, page 174
- Configuring SLA Probe Parameters in a Profile, page 176
- Configuring SLA Statistics Measurement in a Profile, page 178
- Configuring an SLA Operation, page 180
- Configuring an On-Demand SLA Operation, page 182
- Configuring an On-Demand Ethernet SLA Operation for CFM Synthetic Loss Measurement, page 184

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 | cfm-loss-measurement }
4. end
   or
   commit

**DETAILED STEPS**

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

**Example:**

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters the SLA configuration mode.</td>
</tr>
<tr>
<td>ethernet sla</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0:router# ethernet sla
```
How to Configure Ethernet OAM

This section describes how to configure a schedule for an SLA operation probe on an ongoing basis within an SLA profile. For information about how to configure a schedule for a limited, on-demand SLA operation, see the “Configuring an On-Demand SLA Operation” section on page 182.

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**

---

**Configuring a Schedule for an SLA Operation Probe in a Profile**

**Command or Action**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>profile</strong> profile-name <strong>type</strong> {cfm-delay-measurement</td>
<td>cfm-loopback</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0:router(config-sla)# profile Prof1 type cfm-loopback
```

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

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

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

**Example:**

```
RP/0/RP0/CPU0:router(config-sla)# commit
```
# Configuring Ethernet OAM on Cisco IOS XR Software

## How to Configure Ethernet OAM

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Schedules an operation probe in a profile. A profile may contain only one schedule.</td>
</tr>
<tr>
<td>`schedule every week on day [at hh:mm] [for duration {seconds</td>
<td>minutes</td>
</tr>
<tr>
<td>Or</td>
<td></td>
</tr>
<tr>
<td>`schedule every day [at hh:mm] [for duration {seconds</td>
<td>minutes</td>
</tr>
<tr>
<td>Or</td>
<td></td>
</tr>
<tr>
<td>`schedule every number [hours</td>
<td>minutes][first at hh:mm[start]] [for duration {seconds</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof)# schedule every week on Monday at 23:30 for 1 hour</td>
<td></td>
</tr>
<tr>
<td>Or</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof)# schedule every day at 11:30 for 5 minutes</td>
<td></td>
</tr>
<tr>
<td>Or</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof)# schedule every 2 hours first at 13:45:01</td>
<td></td>
</tr>
<tr>
<td>Or</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof)# schedule every 6 hours for 2 hours</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>end</code></td>
<td></td>
</tr>
<tr>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit</td>
<td></td>
</tr>
</tbody>
</table>

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

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring 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} or send packet {every number {milliseconds | seconds | minutes | hours} | once}
3. packet size bytes [test pattern {hex 0xHHHHHHHH | pseudo-random}]
4. priority priority
5. synthetic loss calculation packets number
6. end or commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> probe</td>
<td>Enters the SLA profile probe configuration mode.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router(config-sla-prof)# probe
```

| **Step 2** send burst {every number {seconds | minutes | hours} | once} packet count packets interval number {seconds | milliseconds} or send packet {every number {milliseconds | seconds | minutes | hours} | once} | Configures the number and timing of packets sent by a probe in an operations profile. |

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
```
### Configuring Ethernet OAM on Cisco IOS XR Software

#### How to Configure Ethernet OAM

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><strong>packet size</strong> bytes [test pattern {hex 0xHHHHHHHH</td>
<td>pseudo-random}]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof-pb)# packet size 9000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><strong>priority</strong> priority</td>
<td>Configures the priority of outgoing SLA probe packets.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof-pb)# priority 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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 &quot;on-the-wire&quot; 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.</td>
</tr>
<tr>
<td></td>
<td>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 &quot;loss-measurement counters&quot; configuration under the MEP properties submode.</td>
</tr>
</tbody>
</table>
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.

Prerequisites

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.

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:

SUMMARY STEPS

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>`statistics measure {one-way-delay-ds</td>
<td>one-way-delay-sd</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof)# statistics measure round-trip-delay</td>
<td>Configures the size and number of bins into which to aggregate the results of statistics collection.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# aggregate bins 100 width 10000</td>
<td>• For delay measurements, a width between 1-10000, in milliseconds, must be specified if the number of bins is at least 2.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>`aggregate {bins count width width</td>
<td>none}`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# aggregate bins 100 width 10000</td>
<td>• For jitter measurements, a width between 1-10000, in milliseconds, must be specified if the number of bins is at least 3.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>`buckets size number {per-probe</td>
<td>probes}`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# buckets size 100 per-probe</td>
<td>• For loss measurements, a width between 1-100, in percentage points, must be specified if the number of bins is at least 2.</td>
</tr>
</tbody>
</table>
Configuring an SLA Operation

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

SUMMARY STEPS

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

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>buckets archive number</strong></td>
<td>Configures the number of buckets to store in memory.</td>
</tr>
<tr>
<td><strong>end</strong> or <strong>commit</strong></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

#### Step 4

**Example:**

```
RP/0/RP0/CPU0:router# buckets archive 50
```

#### Step 5

**Example:**

```
RP/0/RP0/CPU0:router# 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.
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
</tbody>
</table>
| `interface [GigabitEthernet | TenGigE] interface-path-id` | Physical interface or virtual interface.  
| **Note**          | Use the `show interfaces` command to see a list of all interfaces currently configured on the router.  
| **Example:**      | For more information about the syntax for the router, use the question mark (?) online help function. |
| RP/0/RP0/CPU0:router(config-if)# interface gigabitethernet 0/1/0/1 |         |
| **Step 2**        |         |
| `ethernet cfm`    | Enters interface CFM configuration mode. |
| **Example:**      |         |
| RP/0/RP0/CPU0:router(config-if)# ethernet cfm |         |
| **Step 3**        |         |
| `mep domain domain-name service service-name mep-id id-number` | Creates a MEP on an interface and enters interface CFM MEP configuration mode.  
| **Example:**      |         |
| RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1 |         |
| **Step 4**        |         |
| `sla operation profile profile-name target 
{mep-id id | mac-address mac-address}` | Creates an operation instance from a MEP to a specified destination.  
| **Example:**      |         |
| RP/0/RP0/CPU0:router(config-if-cfm-mep)# sla operation profile Profile_1 target mac-address 01:23:45:67:89:ab |         |
| **Step 5**        |         |
| `end` or `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.** |         |
| **Example:**      |         |
| RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit |         |
Configuring an On-Demand SLA Operation

The Cisco CRS-1 Router supports configuration of on-demand SLA operations to run on an as-needed basis for a finite period of time.

This section includes the following topics:

- Configuration Guidelines, page 182
- Configuring an On-Demand Ethernet SLA Operation for CFM Delay Measurement, page 182
- Configuring an On-Demand Ethernet SLA Operation for CFM Loopback, page 183
- Configuring an On-Demand Ethernet SLA Operation for CFM Synthetic Loss Measurement, page 184

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, including 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.

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:
### 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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ethernet sla on-demand operation type cfm-loopback probe [packet size bytes [test pattern {hex 0xHHHHHHHH</td>
<td>pseudo-random}]] [priority number]</td>
</tr>
<tr>
<td></td>
<td>• Send a burst once for a packet count of 10 and interval of 1 second (10-second probe).</td>
</tr>
<tr>
<td></td>
<td>• Use default class of service (CoS) for the egress interface.</td>
</tr>
<tr>
<td></td>
<td>• Measure all statistics, including both one-way and round-trip delay and jitter statistics.</td>
</tr>
<tr>
<td></td>
<td>• Aggregate statistics into one bin.</td>
</tr>
<tr>
<td></td>
<td>• Schedule now.</td>
</tr>
<tr>
<td></td>
<td>• Display results on the console.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0# 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
```

### 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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ethernet sla on-demand operation type cfm-loopback probe [packet size bytes [test pattern {hex 0xHHHHHHHH</td>
<td>pseudo-random}]] [priority number]</td>
</tr>
<tr>
<td></td>
<td>• Send a burst once for a packet count of 10 and interval of 1 second (10-second probe).</td>
</tr>
<tr>
<td></td>
<td>• Use default class of service (CoS) for the egress interface.</td>
</tr>
<tr>
<td></td>
<td>• Measure all statistics.</td>
</tr>
<tr>
<td></td>
<td>• Aggregate statistics into one bin.</td>
</tr>
<tr>
<td></td>
<td>• Schedule now.</td>
</tr>
<tr>
<td></td>
<td>• Display results on the console.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0# 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
```
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 domain D1 source interface TenGigE 0/6/1/0 target mac-address 2.3.4
```

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

Command Purpose
---
ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe [priority number]
[send {packet | burst} {once | every number {milliseconds | 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]

Example:
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.

Displays information about errors that are preventing configured SLA operations from becoming active, as well as any warnings that have occurred.

Displays information about configured SLA operations.
Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

- Configuration Examples for EOAM Interfaces, page 185
- Configuration Examples for Ethernet CFM, page 187
- Configuration Examples for Ethernet SLA, page 198

Configuration Examples for EOAM Interfaces

This section provides the following configuration examples:

- Configuring an Ethernet OAM Profile Globally: Example, page 185
- Configuring Ethernet OAM Features on an Individual Interface: Example, page 185
- Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example, page 186
- Clearing Ethernet OAM Statistics on an Interface: Example, page 187
- Enabling SNMP Server Traps on a Router: Example, page 187

Configuring an Ethernet OAM Profile Globally: Example

The following example shows how to configure an Ethernet OAM profile globally:

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

Configuring Ethernet OAM Features on an Individual Interface: Example

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

```
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 low 100 high 900
  exit
mib-retrieval
  connection timeout 30
  require-remote mode active
  require-remote link-monitoring
  require-remote mib-retrieval
  action link-fault error-disable-interface
  action dying-gasp error-disable-interface
  action critical-event error-disable-interface
  action discovery-timeout error-disable-interface
  action session-down error-disable-interface
  action capabilities-conflict error-disable-interface
  action wiring-conflict error-disable-interface
  action remote-loopback error-disable-interface
commit

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

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

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
Clearing Ethernet OAM Statistics on an Interface: Example

The following 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

The following 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, page 187
- Ethernet CFM Service Configuration: Example, page 187
- Continuity Check for an Ethernet CFM Service Configuration: Example, page 188
- MIP Creation for an Ethernet CFM Service Configuration: Example, page 188
- Cross-check for an Ethernet CFM Service Configuration: Example, page 188
- Other Ethernet CFM Service Parameter Configuration: Example, page 188
- MEP Configuration: Example, page 188
- Ethernet CFM Show Command: Examples, page 188
- AIS for CFM Configuration: Examples, page 192
- AIS for CFM Show Commands: Examples, page 192
- EFD Configuration: Examples, page 196
- Displaying EFD Information: Examples, page 196

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
```

Ethernet CFM Service Configuration: Example

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

```
service Cross_Connect_1 xconnect group XG1 p2p X1
```

commit
Continuity Check for an Ethernet CFM Service Configuration: Example

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

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

MIP Creation for an Ethernet CFM Service Configuration: Example

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

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

Cross-check for an Ethernet CFM Service Configuration: Example

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

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

Other Ethernet CFM Service Parameter Configuration: Example

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

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

MEP Configuration: Example

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

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

Ethernet CFM Show Command: Examples

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

Example 1

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

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

<table>
<thead>
<tr>
<th>Domain/Level</th>
<th>Service</th>
<th>Interface</th>
<th>Type</th>
<th>ID</th>
<th>MAC</th>
</tr>
</thead>
</table>
Example 2

The following 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

* An Up MEP is configured for this domain on interface GigabitEthernet0/1/2/3.234 and an Up MEP is also configured for domain blort, which is at the same level (5).

* A MEP is configured on interface GigabitEthernet0/3/2/1.1 for this domain/service, which has CC interval 100ms, but the lowest interval supported on that interface is 1s

Example 3

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

```
RP/0/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

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Interface (State)</th>
<th>Dir MEPs/Err RD Defects AIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Gi1/1/0/1.234 (Up)</td>
<td>0/0 N A L7</td>
</tr>
</tbody>
</table>
```

Domain fred (level 5), Service barney

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Interface (State)</th>
<th>Dir MEPs/Err RD Defects AIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Gi0/1/0/0.234 (Up)</td>
<td>3/2 Y RPC L6</td>
</tr>
</tbody>
</table>
```

Example 4

The following 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

```
<table>
<thead>
<tr>
<th>St</th>
<th>ID MAC address</th>
<th>Port</th>
<th>Up/Downtime</th>
<th>CcmRcvd</th>
<th>SeqErr</th>
<th>RDI Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>0011.2233.4455 Up 00:00:01</td>
<td>1234</td>
<td>0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R&gt;</td>
<td>4455.6677.8899 Up 03:04</td>
<td>3456</td>
<td>0 234 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>2 1122.3344.5566 Up 6h</td>
<td>3254</td>
<td>0 0 3254</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2 7788.9900.1122 Test 00:13</td>
<td>2345</td>
<td>6 20 2345</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>3 2233.4455.6677 Up 00:23</td>
<td>30</td>
<td>0 0 30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>3 3344.5566.7788 Down 00:34</td>
<td>12345</td>
<td>0 300 1234</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Domain fred (level 7), Service fig
Down MEP on GigabitEthernet0/10/0/12.123, MEP-ID 3

Example 5

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

```
RP/0/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
CPM 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
CPM 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
CPM 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
The following example shows how to configure Alarm Indication Signal (AIS) transmission for a CFM domain service:

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

Example 2
The following example shows how to configure AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received:

```
RP/0/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 Cross_Connect_1 xconnect group XG1 p2p
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais
```

The following 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/RP0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7
```

AIS for CFM Show Commands: Examples

This section includes the following examples:

- `show ethernet cfm interfaces ais Command: Example, page 193`
- `show ethernet cfm local meps Command: Examples, page 193`
show ethernet cfm interfaces ais Command: Example

The following 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

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

show ethernet cfm local meps Command: Examples

**Example 1: Default**

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

```
RP/0/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/RSP0/RP0/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
<table>
<thead>
<tr>
<th>Interface state: Up</th>
<th>MAC address: 1122.3344.5566</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)</td>
<td></td>
</tr>
<tr>
<td>CCM generation enabled: No</td>
<td></td>
</tr>
<tr>
<td>AIS generation enabled: Yes (level: 7, interval: 1s)</td>
<td></td>
</tr>
<tr>
<td>Sending AIS: Yes (started 01:32:56 ago)</td>
<td></td>
</tr>
<tr>
<td>Receiving AIS: Yes (from lower MEP, started 01:32:56 ago)</td>
<td></td>
</tr>
</tbody>
</table>
Example 3: Verbose
The following example shows how to display verbose statistics for MEPs in a domain service:

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

```
RP/0/RSP0RP0/CPU0:router# show ethernet cfm local meps domain foo service bar verbose
```

```
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)

  Packet     Sent       Received
            ---------  -----------------------------------------------
    CCM       0           0  (out of seq: 0)
    LBM       0           0
    LBR       0           0  (out of seq: 0, with bad data: 0)
    AIS       5576        0
    LCK       -           0

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

```
RP/0/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

The following 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
```

The following 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, page 197
- show ethernet cfm local meps detail Command: Example, page 197
show efd interfaces Command: Example

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

```
RP/0/RP0/CPU0:router# show efd interfaces
Server VLAN MA
=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~=~
Configuration Examples for Ethernet SLA

This section includes the following examples:

- Ethernet SLA Profile Type Configuration: Examples, page 198
- Ethernet SLA Probe Configuration: Examples, page 198
- Profile Statistics Measurement Configuration: Examples, page 199
- Scheduled SLA Operation Probe Configuration: Examples, page 200
- Ethernet SLA Operation Probe Scheduling and Aggregation Configuration: Example, page 200
- Ongoing Ethernet SLA Operation Configuration: Example, page 201
- On-Demand Ethernet SLA Operation Basic Configuration: Examples, page 202
- Ethernet SLA Y.1731 SLM Configuration: Examples, page 202
- Ethernet SLA Show Commands: Examples, page 203

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
```

Configuration Examples for Ethernet OAM

Example 2
This example has the same characteristics as the configuration in Example 1, but sends a single burst of 50 packets, one second apart:

```plaintext
configure
ethernet sla
profile Prof1 type cfm-loopback
probe
  send burst once packet count 50 interval 1 second
  packet size 9000 test pattern hex 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:

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

Profile Statistics Measurement Configuration: Examples

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:

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

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

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

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

Figure 13 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
```
Figure 13  
**SLA Probe Scheduling Operation With Bin Aggregation**

This example schedules a probe with the following characteristics:

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

**Ongoing Ethernet SLA Operation Configuration: Example**

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:

```bash
RP/0/RP0/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:

```bash
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 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:

```bash
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:

```bash
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/RP0/CPU0:router# show ethernet sla statistics history detail on-demand

Interface GigabitEthernet0/0/0/0.1
Domain mydom Service myser to 0123.4567.890A
On-demand operation ID #1, packet type 'cfm-delay-measurement'
Started at 15:38 on 06 July 2010 UTC, runs every 1 hour for 1 hour

Round Trip Delay
~~~~~~~~~~~~~~~~
1 bucket per probe

Bucket started at 15:38 on Tue 06 Jul 2010 UTC, lasting 1 hour:
  Pkts sent: 1200; Lost: 4 (0%); Corrupt: 600 (50%); Misordered: 0 (0%)
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, 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:

<table>
<thead>
<tr>
<th>Time</th>
<th>Result</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>04:50:00.0</td>
<td>1.50%</td>
<td>(3 of 200)</td>
</tr>
<tr>
<td>04:50:10.0</td>
<td>2.00%</td>
<td>(4 of 200)</td>
</tr>
<tr>
<td>04:50:20.0</td>
<td>5.50%</td>
<td>(11 of 200)</td>
</tr>
<tr>
<td>04:50:30.0</td>
<td>3.00%</td>
<td>(6 of 200)</td>
</tr>
<tr>
<td>04:50:40.0</td>
<td>0.50%</td>
<td>(1 of 200)</td>
</tr>
<tr>
<td>04:50:50.0</td>
<td>0.00%</td>
<td>(0 of 200)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Range</th>
<th>Count</th>
<th>Cum. Count</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 30%</td>
<td>20 (13.9%)</td>
<td>20 (13.9%)</td>
<td>21.00%</td>
</tr>
<tr>
<td>30 to 60%</td>
<td>71 (49.3%)</td>
<td>91 (63.2%)</td>
<td>57.90%</td>
</tr>
<tr>
<td>60 to 100%</td>
<td>49 (34.0%)</td>
<td>144 (100.0%)</td>
<td>62.00%</td>
</tr>
</tbody>
</table>

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 Cisco IOS XR Software” module later in this document.

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

Additional References

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

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>
### Related Topic

<table>
<thead>
<tr>
<th>Information about user groups and task IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Document Title</strong></td>
</tr>
<tr>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference</em></td>
</tr>
</tbody>
</table>

| Information about configuring interfaces and other components on the Cisco CRS-1 Router from a remote Craft Works Interface (CWI) client management application |
| **Document Title** |
| *Cisco Craft Works Interface Configuration Guide* |

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.1ag</td>
<td><em>Connectivity Fault Management</em></td>
</tr>
<tr>
<td>ITU-T Y.1731 (July 2011)</td>
<td><em>OAM Functions and Mechansims for Ethernet Based Networks</em></td>
</tr>
<tr>
<td>MEF 16</td>
<td><em>Metro Ethernet Forum, Technical Specification MEF 16, Ethernet Local Management Interface (E-LMI), January 2006</em></td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
</table>
| IEEE8021-CFM-MIB | To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: 

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

### Technical Assistance

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

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.2</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>This feature was updated as follows:</td>
</tr>
<tr>
<td></td>
<td>• To support the 1:N redundancy feature, users can configure the minimum number of active links using the <code>bundle minimum-active links</code> command.</td>
</tr>
<tr>
<td></td>
<td>• To support the 1:N redundancy feature, users can configure the minimum bandwidth in kbps using the <code>bundle minimum-active links</code> command.</td>
</tr>
<tr>
<td></td>
<td>• Support was added for VLAN subinterfaces on Ethernet link bundles.</td>
</tr>
<tr>
<td></td>
<td>• Output for <code>show bundle bundle-Ether</code> command and <code>show bundle bundle-POS</code> command was modified.</td>
</tr>
<tr>
<td></td>
<td>• The <code>reasons</code> keyword was added to the <code>show bundle bundle-Ether</code> command and the <code>show bundle bundle-POS</code> command.</td>
</tr>
<tr>
<td></td>
<td>• The <code>bundle id</code> command was changed from <code>bundle-id</code>.</td>
</tr>
<tr>
<td></td>
<td>• BFD over bundled VLANs using static routes.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>The configuration procedures in this module were modified with enhancements.</td>
</tr>
<tr>
<td>Release 3.7.0</td>
<td>Note was added, specifying that link bundling is supported on the multi-shelf Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.8.0</td>
<td>This feature was updated as follows:</td>
</tr>
<tr>
<td></td>
<td>• The <code>reasons</code> keyword was removed from the <code>show bundle bundle-Ether</code> command and the <code>show bundle bundle-POS</code> command. Now, if a port is in a state other than the distributing state, the output of both commands displays the reason.</td>
</tr>
<tr>
<td></td>
<td>• The <code>hot-standby</code> keyword was added to the <code>bundle maximum-active links</code> command.</td>
</tr>
<tr>
<td></td>
<td>• The <code>lacp fast-switchover</code> command was added.</td>
</tr>
<tr>
<td>Release 3.8.4</td>
<td>Bundle member links are put into new err-disable link interface status and admin-down protocol state when a bundle interface is shut down.</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>Support for super short LACP was added.</td>
</tr>
<tr>
<td></td>
<td>Support for load balancing was added.</td>
</tr>
<tr>
<td></td>
<td>Support for a maximum of 64 member links per bundle was added.</td>
</tr>
<tr>
<td>Release 4.0.0</td>
<td>Support for the following physical layer interface modules (PLIMs) was added:</td>
</tr>
<tr>
<td></td>
<td>• 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)</td>
</tr>
<tr>
<td></td>
<td>• 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)</td>
</tr>
</tbody>
</table>
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 4.3.2 Support for Multichassis Link Aggregation was included.

## Contents

This module includes the following sections:

- Prerequisites for Configuring Link Bundling, page 212
- Information About Configuring Link Bundling, page 214
- How to Configure Link Bundling, page 228
- Configuration Examples for Link Bundling, page 276
- Additional References, page 280
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, page 212

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 WDM PHY 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.

- 24-Port 10GE DX Line Card, Packet Transport Optimized with SFP+ optics
- 24-Port 10GE DX Line Card, Service Edge Optimized with SFP+ optics
- 20-Port GE Modular Port Adapter (MPA) with SFP optics
- 4-Port 10GE Modular Port Adapter (MPA) with SFP+ optics
- 2-Port 100GE DX Line Card, Packet Transport Optimized with CFP optics
- 2-Port 100GE DX Line Card, Service Edge Optimized with CFP optics
### Information About Configuring Link Bundling

To configure link bundling, you must understand the following concepts:

- Link Bundling Overview, page 214
- Features and Compatible Characteristics of Link Bundles, page 215
- Link Aggregation Through LACP, page 216
- Multichassis Link Aggregation, page 217
- LACP Short Period Time Intervals, page 225
- Load Balancing, page 226
- QoS and Link Bundling, page 227
- Link Bundle Configuration Overview, page 227
- Nonstop Forwarding During RP Switchover, page 228
- Link Switchover, page 228

### 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.
- EtherChannel 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 Link Bundles

Link bundles support the following features:

- ACL
- Basic IP
- Basic MPLS
- MPLS VPN
- Sampled Netflow
- BGP Policy Accounting
- HSRP/VRRP
- VLAN Bundling (Ethernet only)
- Basic IP
- Basic MPLS
- MPLS VPN
- Inter-AS
- WRED/MDRR per member interface.

The following list describes the properties and limitations of link bundles:

- A bundle contains links, each of which has LACP enabled or disabled. If a bundle contains links, some that have LACP enabled and some that have LACP disabled, the links with LACP disabled are not aggregated in the bundle.
- Bundle membership can span across several modular services cards that are installed in a single router and across SPAS in the same service card.
- 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.
- 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. 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.
• 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, internal processes
select the member link, and all traffic for that flow is sent over that member.

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.

Multichassis Link Aggregation

The Multichassis Link Aggregation (MC-LAG) feature provides an end to end interchassis redundancy solution for the Carrier Ethernet Networks. MC-LAG involves two devices collaborating to act as a single LAG from the perspective of a (third) connected device, thus providing device-level as well as link-level redundancy.

To achieve this, two devices co-ordinate with each other to present a single LACP bundle (spanning the two devices) to a partner device. Only one of the devices forwards traffic at any one time, eliminating the risk of forwarding loops. When a failure occurs, these devices coordinate to perform a switchover, changing the device on which traffic is being forwarded by manipulating the link LACP states.
The existing pseudowire redundancy in the core network coordinates with the redundancy in the access network based on:

- Multichassis Link Aggregation Control Protocol (mLACP)
- Interchassis Communication Protocol (ICCP)

The mLACP protocol defines the expected behavior between the two devices and uses the Interchassis Control Protocol (ICCP) to exchange TLVs and identify peer devices to operate with. At the edge of a provider’s network, a simple customer edge (CE) device that only supports standard LACP is connected to two provider edge (PE) devices. Thus the CE device is dual-homed, providing better L2 redundancy from the provider’s side. In mLACP terminology, the CE device is referred to as a dual-homed device (DHD) and each PE device is known as a point of attachment (POA). The POA forwarding traffic for the bundle is the active device for that bundle, while the other POA is the standby device.

### Failure Cases

MC-LAG provides redundancy, switching traffic to the unaffected POA while presenting an unchanged bundle interface to the DHD, for these failure events:

- Link failure: A port or link between the DHD and one of the POAs fails.
- Device failure: Meltdown or reload of one of the POAs, with total loss of connectivity (to the DHD, the core and the other POA).
- Core isolation: A POA loses its connectivity to the core network, and therefore is of no value, being unable to forward traffic to or from the DHD.

A loss of connectivity between the POAs leads both devices to assume that the other has experienced device failure, causing them to attempt to take on the Active role. This is known as a split brain scenario and can happen in either of the following cases:

- All other connectivity remains; only the link between POAs is lost.
- One POA is isolated from the core network (i.e. a core isolation scenario where the connection between the two POAs was over the core network).

MC-LAG by itself does not provide a means to avoid this situation; resiliency in the connection between the POAs is a requirement. The DHD is given the responsibility of mitigating the problem by setting a limit on the number of links, within the bundle, that can be active. As such only the links connected to one of the POAs can be active at any one point of time.

### Interchassis Communication Protocol

Figure 14 shows the graphical representation of the Interchassis Communication Protocol (ICCP).
Two POAs communicate with each other over an LDP link using the Interchassis Communication Protocol (ICCP). ICCP is an LDP based protocol wherein an LDP session is created between the POAs in a redundancy group, and the ICCP messages are carried over that LDP session. The PE routers in a redundancy group may be a single-hop (directly connected) or a multi-hop away from one another. The ICCP protocol manages the setup and controls the redundancy groups. It also establishes, maintains, and tears down ICCP connections. The ICCP protocol uses route-watch to monitor the connectivity to the PEs in a given redundancy group. It is also responsible for tracking core isolation failures. It notifies all client applications of failure (core isolation and active PE failure).

To operate ICCP, the devices are configured as members of redundancy groups (RGs).

**Note**
In the mLACP configuration, two devices are configured to be members of each RG (until a device-level failure occurs leaving only a single member). However, each device can be a member of more than one RG.

In each redundancy group, a POA's mLACP peer is the other POA in that group, with which it communicates using mLACP over ICCP. For each bundle, the POA and DHD at each end are LACP partners, communicating using the standard LACP protocol.

**Access Network Redundancy Model**

The Multichassis Link Aggregation Control Protocol (mLACP) based redundancy between the customer edge device (CE) or access network and the provider edge (PE) device is achieved by allowing the CE to be connected to two PE routers. The two PE routers synchronize the data through ICCP; therefore they appear as a single device to the CE.
The CE is also called dual-homed device (DHD) and the PE is also called point of attachment (POA). The pair of POAs that is connected to the single DHD forms a redundancy group (RG).

At any given time, only one POA is active for a bundle. Only the set of links between the DHD and the active POA actively sends traffic. The set of links between the DHD and the standby POA does not forward traffic. When the multichassis link bundle software detects that the connection to the active POA has failed, the software triggers the standby POA to become the active POA, and the traffic flows using the links between the DHD and newly active POA.

The ICCP protocol operates between the active and the standby POAs, and allows the POAs to coordinate their configuration, determine which POA is active, and trigger a POA to become active. Applications running on the two POAs (mLACP, IGMP snooping, DHCP snooping or ANCP) synchronize their state using ICCP.
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.

Failure Modes

The mLACP feature provides network resiliency by protecting against port, link, and node failures. Figure 16 depicts the various failure modes.

Figure 16 Failure Modes

These are the failure categories:

- A—DHD uplink port failure. The port on the DHD that is connected to the POA fails.
- B—DHD uplink failure. The connection between the DHD and the POA fails.
- C—Active POA downlink port failure.
- D—Active POA node failure.
- E—Active POA uplink failure (network isolation). The links between the active POA and the core network fails

Core Network Redundancy Model

This section explains:

- One-way pseudowire redundancy
- Two-way pseudowire redundancy
One-way Pseudowire Redundancy

Figure 17 shows the VPWS one-way pseudowire redundancy model. Only one end of the pseudowire is protected by a backup pseudowire.

Figure 17  VPWS one-way Pseudowire Redundancy

Two-way Pseudowire Redundancy

Figure 18 shows the VPWS two-way pseudowire redundancy model. In this topology, each T-PE at the end of a PW has a primary and a backup PW. The state of the PW is coordinated with the state of the mLACP link between the DHD and the PE.

Figure 18  VPWS two-way Pseudowire Redundancy

Switchovers

Switchovers, which is changing the Active/Standby roles of the POAs, are performed using dynamic priority management or brute force behavior.

Dynamic Priority Management

Dynamic Priority Management involves co-ordination between the POAs to manipulate the LACP port priorities of their member links. Two priority values are tracked for each links:
• A configured priority which can either be configured explicitly, or defaults to 32768
• An operational priority used in LACP negotiations, which may differ from the configured priority if switchovers have occurred.

Higher priority LACP links are always selected ahead of lower priority LACP links. This means the operational priorities can be manipulated to force the standard LACP Selection Logic (on the POAs and on the DHD) to select desired links on both ends.

For example, consider a case where the DHD has two links to each POA, and each POA is configured with minimum-active links is 2. (This means the bundle goes down on the POA if the number of active links falls below 2.) The operational priorities for the member links are 1 on POA-1 and 2 on POA-2. This means that POA-1 is active (being higher priority) and the links on POA-2 are held in Standby state. The sequence of events in a switchover is as follows:

1. A link fails on POA-1, causing the number of active links to fall below the minimum of 2.
2. POA-1 changes the operational priority of both its links to 3, so the links on POA 2 are now higher priority.
3. POA-1 sends a LACP message to the DHD and an mLACP message to POA-2, informing both devices of the change.
4. The DHD tries to activate the links connected to POA-2 as these now have the highest priority.
5. POA-2 also ensures that its links have the highest priority and activates its links to the DHD.

At this point the switchover is complete.

Brute Force Behavior

In a brute force switchover, port priorities are not modified. Instead the failing POA sends a single dying gasp to the DHD over LACP, forcing it to deselect the link. It then terminates LACP communications on that link. This only leaves links between the DHD and POA-2, as links that can be selected. So, both ends select those links.

MC-LAG Topologies

This section illustrates the supported MC-LAG topologies.

Figure 19  VPWS One-way Pseudowire Redundancy in Redundancy Group

Multi-chassis LAG (MC-LAG)  L2

Inter-chassis Control Channel (ICC)

DHD1  L1  PE1  PW 2  PW 4
PW 1  PW 3

PE3  L2

PE2  L4  DHD2

L3
Figure 20  VPWS Two-way Pseudowire Redundancy

Multi-chassis LAG (MC-LAG)  L2

DHD1  L1

PE1  PW 2  PW 1  PW 3  PW 4

PE3  L3  L4

DHD2

Inter-chassis Control Channel (ICC)

Figure 21  VPLS Pseudowires in One Redundancy Group

Multi-chassis LAG (MC-LAG)  L2

DHD1  L1

PE1  PW 2  PW 1  PW 3  PW 4

PE3  L3  L4

DHD2

Inter-chassis Control Channel (ICC)

Figure 22  VPLS Pseudowires in Two Redundancy Groups

Multi-chassis LAG (MC-LAG)  L2

DHD1  L1

PE1  PW 2  PW 1  PW 3  PW 4

PE3  L3  L4

DHD2

Inter-chassis Control Channel (ICC)
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.
QoS and Link Bundling

On the Cisco CRS 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 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.

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, page 229
- Configuring EFP Load Balancing on an Ethernet Link Bundle, page 233
- Configuring VLAN Bundles, page 235
- Configuring POS Link Bundles, page 242
Configuring Ethernet Link Bundles

This section describes how to configure an Ethernet link bundle.

**Note**
MAC accounting is not supported on Ethernet link bundles.

**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**

The creation of an Ethernet link bundle involves creating a bundle and adding member interfaces to that bundle, as shown in the steps that follow.

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. Repeat Step 8 through Step 11 to add more links to the bundle you created in Step 2.
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-Ether bundle-id
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface Bundle-Ether bundle-id</code></td>
<td>Creates a new Ethernet link bundle with the specified bundle-id. The range is 1 to 65535.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3</td>
<td>This <code>interface Bundle-Ether</code> command enters you into the interface configuration submode, where you can enter interface specific configuration commands are entered. Use the <code>exit</code> command to exit from the interface configuration submode back to the normal global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>ipv4 address ipv4-address mask</code></td>
<td>Assigns an IP address and subnet mask to the virtual interface using the <code>ipv4 address</code> configuration subcommand.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0</td>
<td>Note:</td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>bundle minimum-active bandwidth kbps</code></td>
<td>(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>bundle minimum-active links links</code></td>
<td>(Optional) Sets the number of active links required before you can bring up a specific bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>bundle maximum-active links links [hot-standby]</code></td>
<td>(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-high-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby</td>
<td><strong>Note</strong> The priority of the active and standby links is based on the value of the <code>bundle port-priority</code> command.</td>
</tr>
</tbody>
</table>
### How to Configure Link Bundling

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong></td>
<td>lACP fast-switchover</td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# lACP fast-switchover</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>exit</td>
<td>Exits interface configuration submode for the Ethernet link bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>interface {GigabitEthernet</td>
<td>TenGigE} interface-path-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>bundle id bundle-id [mode {active</td>
<td>on</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# bundle-id 3</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If you do not specify the mode keyword, the default mode is on (LACP is not run over the port).</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>bundle port-priority priority</td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>no shutdown</td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# no shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>exit</td>
<td>Exits interface configuration submode for the Ethernet interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### How to Configure Link Bundling

**Command or Action**

| Step 14 | interface [GigabitEthernet | TenGigE] number 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  
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 16**

exit

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

Exits interface configuration mode.
### 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

Perform the following steps to configure EFP Load Balancing on an Ethernet link bundle:

1. `configure`
2. `interface Bundle-Ether bundle-id l2transport`
3. `bundle load-balance hash hash-value [auto]`
4. `end`
   or
   `commit`

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 17</strong> <code>exit</code></td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0_CPU0:router(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong> <code>Perform Step 1 through Step 15 on the remote end of the connection.</code></td>
<td>Brings up the other end of the link bundle.</td>
</tr>
<tr>
<td><strong>Step 19</strong> <code>show bundle Bundle-Ether bundle-id</code></td>
<td>(Optional) Shows information about the specified Ethernet link bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0_CPU0:router# show bundle Bundle-Ether 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 20</strong> <code>show lacp bundle Bundle-Ether bundle-id</code></td>
<td>(Optional) Shows detailed information about LACP ports and their peers.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0_CPU0:router# show lacp bundle Bundle-Ether 3</td>
<td></td>
</tr>
</tbody>
</table>
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface Bundle-Ether bundle-id l2transport</td>
<td>Creates a new Ethernet link bundle with the specified bundle-id and with Layer 2 transport enabled. The range is 1 to 65535.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3 l2transport</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bundle load-balance hash hash-value [auto]</td>
<td>Configures all egressing traffic on the fixed members of a bundle to flow through the same physical member link.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1</td>
<td>• 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.</td>
</tr>
<tr>
<td>or</td>
<td>• auto—The physical member link through which all egressing traffic on this bundle will flow is automatically chosen.</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>or</td>
<td>• When you issue the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td>commit</td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# end</td>
<td>– Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td>or</td>
<td>– Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td>or</td>
<td>– Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td>or</td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
Configuring VLAN Bundles

This section describes how to configure a VLAN bundle. The creation of a VLAN bundle involves three main tasks:

1. Create an Ethernet bundle
2. Create VLAN subinterfaces and assign them to the Ethernet bundle.
3. Assign Ethernet links to the Ethernet bundle.

These tasks are described 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

The creation of a VLAN link bundle is described in the steps that follow.

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. encapsulation vlan vlan-id
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 bundle you created in Step 2.
24. end
   or
   commit

25. Perform Step 1 through Step 23 on the remote end of the 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 ]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface Bundle-Ether bundle-id</td>
<td>Creates and names a new Ethernet link bundle.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3</td>
<td>This <code>interface Bundle-Ether</code> command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the <code>exit</code> command to exit from the interface configuration submode back to the normal global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv4 address ipv4-address mask</td>
<td>Assigns an IP address and subnet mask to the virtual interface using the <code>ipv4 address</code> configuration subcommand.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> bundle minimum-active bandwidth kbps</td>
<td>(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> bundle minimum-active links links</td>
<td>(Optional) Sets the number of active links required before you can bring up a specific bundle.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> bundle maximum-active links links [ hot-standby ]</td>
<td>(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.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The priority of the active and standby links is based on the value of the <code>bundle port-priority</code> command.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 7    | `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.  
**Note** |
| 8    | `exit` | Exits the interface configuration submode. |
| 9    | `interface Bundle-Ether bundle-id.vlan-id` | 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. |
| 10   | `encapsulation vlan vlan-id` | 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).** |
| 11   | `ipv4 address ipv4-address mask` | Assigns an IP address and subnet mask to the subinterface. |
| 12   | `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. |
| 13   | `exit` | Exits subinterface configuration mode for the VLAN subinterface. |
How to Configure Link Bundling

Step 14
Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.

```
interface Bundle-Ether bundle-id_vlan-id
dot1q vlan vlan-id
ipv4 address ipv4-address mask
no shutdown
exit
```

**Example:**
```
RP/0/RP0/CPU0:router(config-subif)# interface Bundle-Ether 3.1
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 20.2.3.4/24
RP/0/RP0/CPU0:router(config-subif)# no shutdown
exit
```

(Optional) Adds more subinterfaces to the bundle.

Step 15
```
end
or
commit
```

**Example:**
```
RP/0/RP0/CPU0:router(config-subif)# end
```
Saves configuration changes.

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

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
```

  -Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

  -Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

  -Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

Step 16
```
exit
```
Exits interface configuration mode.

**Example:**
```
RP/0/RP0/CPU0:router(config-subif)# end
```

Step 17
```
exit
```
Exits global configuration mode.

**Example:**
```
RP/0/RP0/CPU0:router(config)# exit
```
## Configuring Link Bundling on Cisco IOS XR Software

### How to Configure Link Bundling

1. **Step 18**
   - **Command or Action**: `configure`
   - **Example**: RP/0/RP0/CPU0:router # configure
   - **Purpose**: Enters global configuration mode.

2. **Step 19**
   - **Command or Action**: `interface {GigabitEthernet | TenGigE} interface-path-id`
   - **Example**: RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0
   - **Purpose**: 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.

3. **Step 20**
   - **Command or Action**: `bundle id bundle-id [mode {active | on | passive}]`
   - **Example**: RP/0/RP0/CPU0:router(config-if)# bundle-id 3
   - **Purpose**: 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).

4. **Step 21**
   - **Command or Action**: `bundle port-priority priority`
   - **Example**: RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1
   - **Purpose**: (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.

5. **Step 22**
   - **Command or Action**: `no shutdown`
   - **Example**: RP/0/RP0/CPU0:router(config-if)# no shutdown
   - **Purpose**: (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.

6. **Step 23**
   - **Command or Action**: —
   - **Purpose**: Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.
### Command or Action

**Step 24**

- `end`  
- `commit`

**Example:**

```
RP/0/RP0/CPU0:router(config-subif)# end
```

**Step 25**

Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.

**Step 26**

```
show bundle Bundle-Ether bundle-id
```

**Example:**

```
RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3
```

**Step 27**

```
show vlan interface
```

**Example:**

```
RP/0/RP0/CPU0:router # show vlan interface
```

### Purpose

Saves configuration changes.

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

  Uncommitted changes found, commit them before exiting(yes/no/cancel)?

  [cancel]:

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

(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.”

Displays the current VLAN interface and status configuration.
### Command or Action

**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.
  - `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

The creation of a bundled POS interface involves configuring both the bundle and the member interfaces, as shown in these 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 the bundle you created in Step 2.
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 bundle-id
20. show lacp bundle bundle-POS bundle-id
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface Bundle-POS bundle-id</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv4 address ipv4-address mask</td>
<td>Assigns an IP address and subnet mask to the virtual interface using the ip address configuration subcommand.</td>
</tr>
<tr>
<td><strong>Step 4</strong> bundle minimum-active bandwidth kbps</td>
<td>(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.</td>
</tr>
<tr>
<td><strong>Step 5</strong> bundle minimum-active links links</td>
<td>(Optional) Sets the number of active links required before you can bring up a specific bundle.</td>
</tr>
</tbody>
</table>
| **Step 6** bundle maximum-active links links [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 | (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 | Exits the interface configuration submode. |
### Command or Action

**Step 9**  
`interface POS interface-path-id`

**Example:**  
RP/0/RP0/CPU0:router(config)# interface POS 0/1/0/0

**Purpose:**  
Enteres 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

**Purpose:**  
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

**Purpose:**  
(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

**Purpose:**  
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

**Purpose:**  
Exits the interface configuration submode for the POS interface.

**Step 14**  
Repeat Step 8 through Step 11 to add more links to a bundle

**Purpose:**  
(Optional) Adds more links to the bundle you created in Step 2.

---

**Command or Action** | **Purpose**
--- | ---
`interface POS interface-path-id` | Enters POS interface configuration mode and specifies the POS interface name and interface-path-id notation `rack/slot/module/port`.
`bundle id bundle-id [mode {active | on | passive}]` | 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).
`bundle port-priority priority` | (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.
`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.
`exit` | Exits the interface configuration submode for the POS interface.
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.
### Command or Action

<table>
<thead>
<tr>
<th>Step 15</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>or</td>
<td>• When you issue the <strong>end</strong> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td>commit</td>
<td>Uncommitted changes found, commit them before exiting (yes/no/cancel)?</td>
</tr>
<tr>
<td>Example:</td>
<td>[cancel]:</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# end</td>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td>or</td>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>

### Purpose

**Saves configuration changes.**

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

<table>
<thead>
<tr>
<th>Step 16</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>exit</td>
<td>Exits interface configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

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

### Purpose

**Exits interface configuration mode.**

<table>
<thead>
<tr>
<th>Step 17</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>exit</td>
<td>Exits global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

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

### Purpose

**Exits global configuration mode.**

<table>
<thead>
<tr>
<th>Step 18</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform Step 1 through Step 15 on the remote end of the connection.</td>
<td>Brings up the other end of the link bundle.</td>
</tr>
</tbody>
</table>

### Purpose

**Brings up the other end of the link bundle.**

<table>
<thead>
<tr>
<th>Step 19</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>show bundle Bundle-POS number</td>
<td>(Optional) Shows information about the specified POS link bundle.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RP0/CPU0:router# show bundle Bundle-POS 1

<table>
<thead>
<tr>
<th>Step 20</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>show lacp bundle Bundle-POS bundle-id</td>
<td>(Optional) Shows detailed information about LACP ports and their peers.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RP0/CPU0:router# show lacp bundle Bundle-POS 3
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

To enable an LACP short period time interval, using the default time of 1 second, perform the following steps.

1. configure
2. interface GigabitEthernet interface-path
3. bundle id number mode active
4. lacp period short
5. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface GigabitEthernet interface-path</td>
<td>Creates a Gigabit Ethernet interface and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bundle id number mode active</td>
<td>Specifies the bundle interface and puts the member interface in active mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>lacp period short</code></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# lacp period short</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>end</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>or <code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# end</td>
</tr>
<tr>
<td>or <code>RP/0/RP0/CPU0:router(config-if)# commit</code></td>
<td></td>
</tr>
</tbody>
</table>

- When you issue the **end** command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

To configure custom receive and transmit intervals for LACP packets, perform the following steps.

**Router A**

1. configure
2. interface GigabitEthernet *interface-path*
3. bundle id *number* mode active
4. lACP period short
5. commit

**Router B**

6. configure
7. interface GigabitEthernet *interface-path*
8. bundle id *number* mode active
9. lACP period short
10. commit

**Router A**

11. configure
12. interface GigabitEthernet *interface-path*
13. lACP period short transmit *interval*
14. commit

**Router B**

15. configure
16. interface GigabitEthernet *interface-path*
17. lACP period short transmit *interval*
18. commit

**Router A**

19. configure
20. interface GigabitEthernet *interface-path*
21. `lacp period short receive interval`
22. `commit`

Router B
23. `configure`
24. `interface GigabitEthernet interface-path`
25. `lacp period short receive interval`
26. `commit` or `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>interface Bundle-Ether bundle-id</code></td>
<td>Creates and names a new Ethernet link bundle.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3</code></td>
<td>This <code>interface Bundle-Ether</code> command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the <code>exit</code> command to exit from the interface configuration submode back to the normal global configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td><code>ipv4 address ipv4-address mask</code></td>
<td>Assigns an IP address and subnet mask to the virtual interface using the <code>ipv4 address</code> configuration subcommand.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>bundle minimum-active bandwidth kbps</code></td>
<td>(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>bundle minimum-active links links</code></td>
<td>(Optional) Sets the number of active links required before you can bring up a specific bundle.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2</code></td>
<td></td>
</tr>
</tbody>
</table>
## Configuring Link Bundling on Cisco IOS XR Software

### How to Configure Link Bundling

#### 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).

**Note** The default number of active links allowed in a single bundle is 8.

**Note** 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.
How to Configure Link Bundling

## Command or Action

<table>
<thead>
<tr>
<th>Step 12</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-subif)# exit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits subinterface configuration mode for the VLAN subinterface.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 13</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 14</strong></td>
<td>end or commit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-subif)# end</code> or <code>RP/0/RP0/CPU0:router(config-subif)# commit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>-</td>
<td>When you issue the <code>end</code> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>- Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>- Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>- Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 15</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-subif)# exit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits interface configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 16</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RPS0/CPU0:router(config)# exit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits global configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 17</th>
<th>show ethernet trunk bundle-ether instance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config)# show ethernet trunk bundle-ether instance</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>(Optional) Displays the interface configuration. The Ethernet bundle instance range is from 1 through 65535.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 18</th>
<th>configure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router # configure</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
### How to Configure Link Bundling

**Step 19**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`interface {GigabitEthernet</td>
<td>TenGigE}` interface-path-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface</td>
<td></td>
</tr>
<tr>
<td>GigabitEthernet 1/0/0/0</td>
<td></td>
</tr>
</tbody>
</table>

**Note** A VLAN bundle is not active until you add an Ethernet interface on both ends of the link bundle.

**Step 20**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`bundle id bundle-id [mode {active</td>
<td>on</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# bundle-id 3</td>
<td></td>
</tr>
</tbody>
</table>

**Note** If you do not specify the mode keyword, the default mode is on (LACP is not run over the port).

**Step 21**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>no shutdown</code></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# no shutdown</td>
<td></td>
</tr>
</tbody>
</table>

**Step 22**

Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.
### Step 23

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 24

Performs 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

(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

(Optional) Displays the interface configuration.

The Ethernet bundle instance range is from 1 through 65535.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-subif)# end</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-subif)# commit</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show bundle Bundle-Ether bundle-id [reasons]</code></td>
<td>(Optional) Shows information about the specified Ethernet link bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3 reasons</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show ethernet trunk bundle-ether instance</code></td>
<td>(Optional) Displays the interface configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td><strong>Router A</strong></td>
<td></td>
</tr>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface GigabitEthernet</td>
<td>Creates a Gigabit Ethernet interface and enters interface configuration mode.</td>
</tr>
<tr>
<td>interface-path</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1</td>
<td></td>
</tr>
<tr>
<td>Step 3 bundle id number mode active</td>
<td>Specifies the bundle interface and puts the member interface in active mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active</td>
<td></td>
</tr>
<tr>
<td>Step 4 lacp period short</td>
<td>Enables the short period time interval.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# lacp period short</td>
<td></td>
</tr>
<tr>
<td>Step 5 commit</td>
<td>Saves configuration changes and exits to EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Router B</strong></td>
<td></td>
</tr>
<tr>
<td>Step 6 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 7 interface GigabitEthernet</td>
<td>Creates a Gigabit Ethernet interface and enters interface configuration mode.</td>
</tr>
<tr>
<td>interface-path</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1</td>
<td></td>
</tr>
<tr>
<td>Step 8 bundle id number mode active</td>
<td>Specifies the bundle interface and puts the member interface in active mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active</td>
<td></td>
</tr>
<tr>
<td>Step 9 lacp period short</td>
<td>Enables the short period time interval.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# lacp period short</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>10</td>
<td>commit</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>configure</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>12</td>
<td>interface GigabitEthernet interface-path</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1</td>
</tr>
<tr>
<td>13</td>
<td>lacp period short transmit interval</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# lacp period short transmit 500</td>
</tr>
<tr>
<td>14</td>
<td>commit</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>configure</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>16</td>
<td>interface GigabitEthernet interface-path</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1</td>
</tr>
<tr>
<td>17</td>
<td>lacp period short transmit interval</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# lacp period short transmit 500</td>
</tr>
<tr>
<td>18</td>
<td>commit</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure Link Bundling

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 19</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 20</strong> interface GigabitEthernet interface-path</td>
<td>Creates a Gigabit Ethernet interface and enters interface configuration mode at one end of the connection.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 21</strong> lACP period short receive interval</td>
<td>Configures the short period receive time interval for LACP packets at one end of the connection.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# lACP period short receive 500</td>
<td>Valid values are 100 to 1000 milliseconds in multiples of 100, such as 100, 200, 300, and so on.</td>
</tr>
<tr>
<td><strong>Step 22</strong> commit</td>
<td>Saves configuration changes and exits to EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Router B</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 23</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 24</strong> interface GigabitEthernet interface-path</td>
<td>Creates a Gigabit Ethernet interface and enters interface configuration mode at one end of the connection.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 25</strong> lacp period short receive interval</td>
<td>Configures the short period receive time interval for LACP packets at one end of the connection. Valid values are 100 to 1000 milliseconds in multiples of 100, such as 100, 200, 300, and so on.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# lacp period short receive 500</td>
<td></td>
</tr>
<tr>
<td><strong>Step 26</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# end or commit</td>
<td></td>
</tr>
</tbody>
</table>

- When you issue the end command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring Multichassis Link Aggregation

Perform these tasks to configure Multichassis Link Aggregation (MC-LAG):

- Configuring Interchassis Communication Protocol, page 258
- Configuring Multichassis Link Aggregation Control Protocol Session, page 261
- Configuring Multichassis Link Aggregation Control Protocol Bundle, page 263
- Configuring Dual-Homed Device, page 265
- Configuring Access Backup Pseudowire, page 267
- Configuring One-way Pseudowire Redundancy in MC-LAG, page 270
- Configuring VPWS Cross-Connects in MC-LAG, page 272
- Configuring VPLS in MC-LAG, page 274

Configuring Interchassis Communication Protocol

Perform this task to configure Interchassis Communication Protocol (ICCP).

**SUMMARY STEPS**

1. `configure`
2. `redundancy iccp group group-id`
3. `member neighbor neighbor-ip-address`
4. `backbone interface interface-type-id`
5. `isolation recovery-delay delay`
6. `end`
   or
   `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# <code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>redundancy iccp group group-id</code></td>
<td>Adds an ICCP redundancy group.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)#<code>redundancy iccp group 100</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 3
- **member neighbor neighbor-ip-address**

  Example:
  
  ```
  RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)#
  member neighbor 10.1.1.1
  ```

  Configures ICCP members.
  This is the ICCP peer for this redundancy group. Only one neighbor can be configured per redundancy group. The IP address is the LDP router-ID of the neighbor. This configuration is required for ICCP to function.

#### Step 4
- **backbone interface interface-type-id**

  Example:
  
  ```
  RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)#
  backbone interface GigabitEthernet0/1/0/2
  ```

  Configures ICCP backbone interfaces.
  This is an optional configuration to detect isolation from the network core, and triggers switchover to the peer POA if the POA on which the failure is occurring is active. Multiple backbone interfaces can be configured for each redundancy group. When all backbone interfaces are not UP, this is an indication of core isolation. When one or more backbone interfaces are UP, then the POA is not isolated from the network core. Backbone interfaces are typically the interfaces which L2VPN pseudowires can use.
### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th><strong>isolation recovery-delay delay</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)# isolation recovery-delay 30</td>
</tr>
</tbody>
</table>

### Purpose

Configures the isolation parameters and specifies delay before clearing isolation condition after recovery from failure.

Isolation recovery delay timer is started once the core isolation condition has cleared. When the timer expires, the POA can take over as the active POA (depending on other conditions like bundle recovery delay timer). This allows:

- the network core to reconverge after the backbone interfaces have come up
- ICCP state to be exchanged in order for POAs to know what state they are supposed to be in so that MCLAG bundles do not flap excessively.

This is an optional configuration; if not configured, the delay is set to 180 seconds, by default.

<table>
<thead>
<tr>
<th>Step 6</th>
<th><strong>end</strong> or <strong>commit</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-redundancy-iccp-group)# end</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-redundancy-iccp-group)# commit</td>
</tr>
</tbody>
</table>

### Purpose

Saves configuration changes.

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

  Uncommitted changes found, commit them before exiting(yes/no/cancel)?

  - 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 Multichassis Link Aggregation Control Protocol Session

Perform this task to enable a Multichassis Link Aggregation Control Protocol (mLACP) session.

SUMMARY STEPS

1. configure
2. redundancy iccp group group-id
3. mlacp system mac mac-id
4. mlacp system priority priority
5. mlacp node node-id
6. end
or
commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> redundancy iccp group group-id</td>
<td>Adds an ICCP redundancy group.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)# redundancy iccp group 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> mlacp system mac mac-id</td>
<td>Configures the LACP system ID to be used in this ICCP Group.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)# mlacp system mac 1.1.1</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The mac-id is a user configured value for the LACP system LAG-ID to be used by the POAs. It is highly recommended that the mac-ids have the same value on both POAs. You can have different LAG-IDs for different groups.</td>
</tr>
<tr>
<td><strong>Step 4</strong> mlacp system priority priority</td>
<td>Sets the LACP system priority to be used in this ICCP Group.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)# mlacp system priority 10</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>It is recommended that system priority of the POAs be configured to a lower numerical value (higher priority) than the LACP LAG ID of the DHD. If the DHD has higher system priority then dynamic priority management cannot work and brute force switchover is automatically used.</td>
</tr>
</tbody>
</table>
### How to Configure Link Bundling

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 5** | **mlacp node node-id** Sets the LACP system priority to be used in this ICCP Group.  
**Example:**  
RP/0/RP0/CPU0:router#(config-redundancy-iccp-group)# mlacp node 1  
**Note** The `node-id` must be unique for each POA. |
| **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. |
Configuring Multichassis Link Aggregation Control Protocol Bundle

Perform this task to configure a Multichassis Link Aggregation Control Protocol (mLACP) bundle.

**SUMMARY STEPS**

1. `configure`
2. `interface Bundle-Ether bundle-id`
3. `mac-address mac-id`
4. `bundle wait-while milliseconds`
5. `lacp switchover suppress-flaps milliseconds`
6. `mlacp iccp-group group-id`
7. `mlacp port-priority priority`
8. `end`
   or
   `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>[RP/0/RP0/CPU0:router# configure]</code></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface Bundle-Ether bundle-id</code></td>
<td>Creates and names a new Ethernet link bundle.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>[RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3]</code></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>mac-address mac-id</code></td>
<td>Sets the MAC address on the interface.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>[RP/0/RP0/CPU0:router#(config-if)# mac-address 1.1.1]</code></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>bundle wait-while milliseconds</code></td>
<td>Sets the wait-while timeout for members of this bundle.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>[RP/0/RP0/CPU0:router#(config-if)# bundle wait-while 100]</code></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>lacp switchover suppress-flaps milliseconds</code></td>
<td>Sets the time for which to suppress flaps during a LACP switchover.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>[RP/0/RP0/CPU0:router#(config-if)# lacp switchover suppress-flaps 300]</code></td>
</tr>
</tbody>
</table>

*Note* Configuring the same MAC address on both POAs is highly recommended.
### Command or Action | Purpose
--- | ---
**Step 6** | **mlacp iccp-group group-id**<br><br>**Example:**<br>RP/0/RP0/CPU0:router#(config-if)# mlacp iccp-group 10
| Configures the ICCP redundancy group in which this bundle should operate.

**Step 7** | **mlacp port-priority priority**<br><br>**Example:**<br>RP/0/RP0/CPU0:router#(config-if)# mlacp port-priority 10
| Sets the starting priority for all member links on this device when running mLACP.<br>Note: Lower value indicates higher priority. If you are using dynamic priority management the priority of the links change when switchovers occur.

**Step 8** | **end**<br>**or**<br>**commit**<br><br>**Example:**<br>RP/0/RP0/CPU0:router(config-if)# end<br>RP/0/RP0/CPU0:router(config-if)# commit
| Saves configuration changes.<br>

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

Perform this task to configure the dual-homed device (DHD).

SUMMARY STEPS

1. configure
2. interface Bundle-Ether bundle-id
3. bundle wait-while milliseconds
4. lacp switchover suppress-flaps milliseconds
5. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface Bundle-Ether bundle-id</td>
<td>Creates and names a new Ethernet link bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router#(config-if)# interface Bundle-Ether 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bundle wait-while milliseconds</td>
<td>Sets the wait-while timeout for members of this bundle.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router#(config-if)# bundle wait-while 100</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure Link Bundling

The members added to the bundle on one POA go Active, and the members on the other POA are in Standby state. This can be verified by using the show bundle command on either POA to display the membership information for correctly configured members on both the POAs:

```
RP/0/RP0/CPU0:router# show bundle

Bundle-Ether1
Status: Up
Local links <active/standby/configured>: 1 / 0 / 1
Local bandwidth <effective/available>: 1000000 (1000000) kbps
MAC address (source): 0000.deaf.0000 (Configured)
Minimum active links / bandwidth: 1 / 1 kbps
Maximum active links: 64
Wait while timer: 100 ms
LACP: Operational
  Flap suppression timer: 300 ms
mLACP: Operational
  ICCP Group: 1
Role: Active
Foreign links <active/configured>: 0 / 1
Switchover type: Non-revertive
Recovery delay: 300 s
Maximize threshold: Not configured
IPv4 BFD: Not configured
```

Command or Action | Purpose
--- | ---
Step 4 `lacp switchover suppress-flaps milliseconds` | Sets the time for which to suppress flaps during a LACP switchover.

**Example:**
```
RP/0/RP0/CPU0:router#(config-if)# lacp switchover suppress-flaps 300
```

Step 5 `end` or `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.**

```
RP/0/RP0/CPU0:router(config-if)# end
or
RP/0/RP0/CPU0:router(config-if)# commit
```
Configuring Access Backup Pseudowire

Perform this task to add a backup pseudowire to a VPLS Access pseudowire.

SUMMARY STEPS

1. configure
2. l2vpn
3. bridge group bridge-group name
4. bridge-domain bridge-domain name
5. neighbor A.B.C.D ip-address pw-id pseudowire-id
6. pw-class {class-class name}
7. backup neighbor A.B.C.D ip-address pw-id pseudowire-id
8. pw-class {class-class name}
9. end
or commit

DETAILED STEPS

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

**Example:**

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

<table>
<thead>
<tr>
<th>Step 2</th>
<th>l2vpn</th>
<th>Enters L2VPN configuration mode.</th>
</tr>
</thead>
</table>

**Example:**

```
RP/0/RP0/CPU0:router(config)# l2vpn
RP/0/RP0/CPU0:router(config-l2vpn)#
```

<table>
<thead>
<tr>
<th>Step 3</th>
<th>bridge group bridge-group-name</th>
<th>Creates a bridge group so that it can contain bridge domains and then assigns network interfaces to the bridge domain.</th>
</tr>
</thead>
</table>

**Example:**

```
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group cisco
RP/0/RP0/CPU0:router(config-l2vpn-bg)#
```
### How to Configure Link Bundling

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> bridge-domain bridge-domain-name</td>
<td>Establishes a bridge domain and enters l2vpn bridge group bridge domain configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RPO/CPU0:router(config-l2vpn-bg)# bridge-domain abc</td>
<td></td>
</tr>
<tr>
<td>RP/0/RPO/CPU0:router(config-l2vpn-bg-bd)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor A.B.C.D pw-id pseudowire-id</td>
<td>Configures the pseudowire segment.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RPO/CPU0:router(config-l2vpn-bg-bd)# neighbor 10.2.2.2 pw-id 2000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> pw-class {class-name}</td>
<td>Configures the pseudowire class template name to use for the pseudowire.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RPO/CPU0:router(config-l2vpn-bg-bd-pw)# pw-class class1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> backup neighbor A.B.C.D pw-id pseudowire-id</td>
<td>Adds a backup pseudowire to a VPLS access pseudowire (PW).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RPO/CPU0:router(config-l2vpn-bg-bd-pw)# backup neighbor 10.2.2.2 pw-id 2000</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 8**  
`pw-class {class-name}`

**Example:**  
RP/0/RP0/CPU0:router(config-l2vpn-bd-pw)#  
pw-class class2

**Purpose:**  
Configures the pseudowire class template name to use for the backup pseudowire.

**Step 9**  
end  
or  
commit

**Example:**  
RP/0/RP0/CPU0:router(config-l2vpn-bd-mac)# end  
or  
RP/0/RP0/CPU0:router(config-l2vpn-bd-mac)# commit

**Purpose:**  
Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
  [cancel]:
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring One-way Pseudowire Redundancy in MC-LAG

Perform this task to allow one-way pseudowire redundancy behavior when the redundancy group is configured.

**SUMMARY STEPS**

1. `configure`
2. `l2vpn`
3. `pw-class {class-name}`
4. `encapsulation mpls`
5. `redundancy one-way`
6. `end`
    or
    `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router# configure</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>l2vpn</code></td>
</tr>
</tbody>
</table>
| **Example:**      | `RP/0/RP0/CPU0:router(config)# l2vpn`  
|                  | `RP/0/RP0/CPU0:router(config-l2vpn)#` |
| **Step 3**        | `pw-class {class-name}` | Configures the pseudowire class template name to use for the pseudowire. |
| **Example:**      | `RP/0/RP0/CPU0:router(config-l2vpn)# pw-class class1` |
| **Step 4**        | `encapsulation mpls` | Configures the pseudowire encapsulation to MPLS. |
| **Example:**      | `RP/0/RP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls` |
### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Configuring Link Bundling on Cisco IOS XR Software</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>redundancy one-way</strong></td>
<td>Configures one-way PW redundancy behavior.</td>
</tr>
</tbody>
</table>

**Example:**

```plaintext
RP/0/RP0/CPU0:router(config-l2vpn-pwc-mpls)# redundancy one-way
```

**Note:** The **redundancy one-way** command is effective only if the redundancy group is configured.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Configuring Link Bundling on Cisco IOS XR Software</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>end</strong> or <strong>commit</strong></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

**Example:**

```plaintext
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-mac)# end
or
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-mac)# commit
```

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

See also:
- [Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router](#)
Configuring VPWS Cross-Connects in MC-LAG

Perform this task to configure VPWS cross-connects in MC-LAG.

SUMMARY STEPS

1. configure
2. l2vpn
3. pw-status
4. xconnect group group-name
5. p2p xconnect-name
6. interface type interface-path-id
7. neighbor A.B.C.D ip-address pw-id pseudowire-id
8. pw-class {class-class name}
9. backup neighbor A.B.C.D ip-address pw-id pseudowire-id
10. pw-class {class-class name}
11. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> pw-status</td>
<td>Enables pseudowire status.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-l2vpn)# pw-status</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>When the attachment circuit changes redundancy state to Active, Active pw-status is sent over the primary and backup pseudowires. When the attachment circuit changes redundancy state to Standby, Standby pw-status is sent over the primary and backup pseudowires.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>xconnect group group-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-l2vpn)# xconnect group grp_1</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters the name of the cross-connect group.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>p2p xconnect-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-l2vpn-xc)# p2p p1</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters a name for the point-to-point cross-connect.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Specifies the interface type ID.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>neighbor A.B.C.D pw-id pseudowire-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 10.2.2.2 pw-id 2000</td>
</tr>
</tbody>
</table>
| **Purpose:** | Configures the pseudowire segment for the cross-connect. 
Optionally, you can disable the control word or set the transport-type to Ethernet or VLAN. |
| **Step 8** | pw-class {class-name} |
| **Example:** | RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# pw-class c1 |
| **Purpose:** | Configures the pseudowire class template name to use for the pseudowire. |
| **Step 9** | backup neighbor A.B.C.D pw-id pseudowire-id |
| **Example:** | RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# backup neighbor 10.2.2.2 pw-id 2000 |
| **Purpose:** | Adds a backup pseudowire. |
### Configuring VPLS in MC-LAG

Perform this task to configure VPLS in MC-LAG.

**SUMMARY STEPS**

1. `configure`
2. `l2vpn`
3. `pw-status`
4. `bridge group bridge-group-name`
5. `bridge-domain bridge-domain-name`

---

**Command or Action**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td><code>pw-class {class-name}</code></td>
<td>Configures the pseudowire class template name to use for the backup pseudowire.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# pw-class c2</td>
<td></td>
</tr>
</tbody>
</table>
| 11   | `end` or `commit` | Saves configuration changes.  
   | **Example:** RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# end or RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# 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. |

**Prerequisite**

You must have configured ICCP as shown in the procedure Configuring Interchassis Communication Protocol.

**Note**

You can use the `show iccp group`, `show l2vpn iccp-sm` and `show lacp bundle-ether` commands to monitor ICCP-SM.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**

`configure`

**Example:**

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

Enters global configuration mode.

| **Step 2**

`l2vpn`

**Example:**

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

Enters L2VPN configuration mode.

| **Step 3**

`pw-status`

**Example:**

```
RP/0/RP0/CPU0:router(config-l2vpn)# pw-status
```

(Optional) Enables pseudowire status.

All the pseudowires in the VFI are always active, independent of the attachment circuit redundancy state.

| **Step 4**

`bridge group bridge-group-name`

**Example:**

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-group cisco
```

Creates a bridge group so that it can contain bridge domains and then assigns network interfaces to the bridge domain.

| **Step 5**

`bridge-domain bridge-domain-name`

**Example:**

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# bridge-domain abc
```

Establishes a bridge domain and enters L2VPN bridge group bridge domain configuration mode.

| **Step 6**

`interface type interface-path-id`

**Example:**

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-bd)# interface bundle-ether 1.1
```

Specifies the interface type ID.

| **Step 7**

`vfi {vfi-name}`

**Example:**

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-ac)# vfi vfi-east
```

Enters virtual forwarding instance (VFI) configuration mode.
### Configuration Examples for Link Bundling

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td><code>neighbor A.B.C.D pw-id pseudowire-id</code></td>
<td>Configures the pseudowire segment for the cross-connect. Optionally, you can disable the control word or set the transport-type to Ethernet or VLAN.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi)# neighbor 10.2.2.2 pw-id 2000</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td></td>
</tr>
<tr>
<td><code>pw-class [class-name]</code></td>
<td>Configures the pseudowire class template name to use for the pseudowire.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# pw-class canada</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td></td>
</tr>
<tr>
<td><code>end</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# end</code></td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# commit</code></td>
<td></td>
</tr>
</tbody>
</table>

- You can configure 8 member links on a bundle.

### Configuration Examples for Link Bundling

This section contains the following examples:

- **Example:** Configuring an Ethernet Link Bundle, page 277
- **Example:** Configuring a VLAN Link Bundle, page 277
- **Example:** Configuring a POS Link Bundle, page 277
- **Example:** Configuring EFP Load Balancing on an Ethernet Link Bundle, page 278
- **Examples:** Configuring LACP Short Periods, page 279
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-if)# 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
```

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)# 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)# ip addr 20.2.3.4/24
RP/0/RP0/CPU0:Router(config-subif)# no shutdown
RP/0/RP0/CPU0:Router(config-subif)# exit
RP/0/RP0/CPU0:Router(config)# interface gig 0/1/5/7
RP/0/RP0/CPU0:Router(config-if)# bundle-id 1 mode act
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
```
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
```

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
```
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):

```plaintext
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**
```plaintext
config
interface gigabitethernet 0/0/0/1
  bundle id 1 mode active
  lacp period short
commit
```

**Router B**
```plaintext
config
interface gigabitethernet 0/0/0/1
  bundle id 1 mode active
  lacp period short
commit
```

**Router A**
```plaintext
config
interface gigabitethernet 0/0/0/1
  lacp period short transmit 100
commit
```

**Router B**
```plaintext
config
interface gigabitethernet 0/0/0/1
  lacp period short transmit 100
commit
```

**Router A**
```plaintext
config
interface gigabitethernet 0/0/0/1
  lacp period short receive 100
commit
```

**Router B**
```plaintext
config
interface gigabitethernet 0/0/0/1
  lacp period short receive 100
commit
```
Additional References

The following sections provide references related to link bundle configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router</td>
<td>Cisco IOS XR Getting Started Guide</td>
</tr>
<tr>
<td>using the Cisco IOS XR software</td>
<td></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.3ad (incorporated as Annex 43 into 802.3-2002)</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The IEEE-defined MIB for Link Aggregation (defined in 802.3 Annex</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR</td>
</tr>
<tr>
<td>30C)</td>
<td>Software, use the Cisco MIB Locator found at the following URL:</td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and</td>
<td>—</td>
</tr>
<tr>
<td>support for existing RFCs has not been modified by this feature.</td>
<td></td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchable technical content, including links to products, technologies,</td>
<td></td>
</tr>
<tr>
<td>solutions, technical tips, and tools. Registered Cisco.com users can</td>
<td></td>
</tr>
<tr>
<td>log in from this page to access even more content.</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Traffic Mirroring on the Cisco IOS XR Software

This module describes the configuration of traffic mirroring on the Cisco CRS Router. Traffic mirroring is sometimes called port mirroring, or switched port analyzer (SPAN).

Feature History for Configuring Traffic Mirroring on the Cisco CRS Router

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.3.0</td>
<td>This feature was introduced on the Cisco CRS Router.</td>
</tr>
</tbody>
</table>

Contents

- Restrictions for Traffic Mirroring, page 284
- Information about Traffic Mirroring, page 284
- Configuring Traffic Mirroring, page 287
- Traffic Mirroring Configuration Examples, page 292
- Additional References, page 296
- Additional References, page 296
Restrictions for Traffic Mirroring

A maximum of eight monitoring sessions, and 800 source ports are supported.

You can configure 800 source ports on a single monitoring session, or configure an aggregate total of 800 source ports on a maximum of 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).
• MPLS traffic or tunnel traffic.
• Layer 2 traffic mirroring.
• VRF at destination ports.
• Mirroring for POS interfaces.
• Mirroring of egress traffic.

Information about Traffic Mirroring

These sections provide information about traffic mirroring:

• Introduction to Traffic Mirroring, page 284
• Traffic Mirroring Terminology, page 285

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 to 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.
Configuring Traffic Mirroring on the Cisco IOS XR Software

Information about Traffic Mirroring

Implementing Traffic Mirroring on the Cisco 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 at most 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.
Configuring Traffic Mirroring on the Cisco IOS XR Software

Information about Traffic Mirroring

Figure 26  Network Analysis on a Cisco CRS Router With Traffic Mirroring

In Figure 26, 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 the following 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:

• A destination is defined by IP address (IPv4 or IPv6), and is not tied to a specific interface (as routing decides which interface the mirrored packets are actually sent over).
• No two monitor sessions must have the same destination IP address.
• Any interface over which the mirrored traffic could potentially be routed must not be configured as a source port.
Figure 27  Network Analysis on a Cisco CRS Router With Traffic Mirroring

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Source traffic mirroring ports (can be ingress or egress traffic ports)</td>
</tr>
<tr>
<td>2</td>
<td>Destination traffic mirroring port</td>
</tr>
</tbody>
</table>

Configuring Traffic Mirroring

These tasks describe how to configure traffic mirroring:

- How to Configure Layer-3 Traffic Mirroring, page 287
- How to Configure ACL-Based Traffic Mirroring, page 289

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`
## Configuring Traffic Mirroring on the Cisco IOS XR Software

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>monitor-session session-name [ipv4</td>
<td>ipv6]</td>
</tr>
<tr>
<td>3</td>
<td>destination next-hop ip address</td>
<td>Configures the destination for the current monitor-session to be a next-hop IP address (whose type matches that of the monitor-session). <strong>Note</strong> 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.</td>
</tr>
<tr>
<td>4</td>
<td>exit</td>
<td>Exits monitor session configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td>5</td>
<td>interface source-interface</td>
<td>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.</td>
</tr>
<tr>
<td>6</td>
<td>monitor-session session-name [ipv4</td>
<td>ipv6] [direction {rx-only</td>
</tr>
</tbody>
</table>
Configuring Traffic Mirroring on the Cisco IOS XR Software

Prerequisites

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>`

### How to Configure ACL-Based Traffic Mirroring

**Prerequisites**

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>`
Configuring Traffic Mirroring on the Cisco IOS XR Software

4. exit
5. interface source-interface
6. ethernet-services access-group access-list-name ingress
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router# configure</strong></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>**monitor-session session-name [ipv4</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config)# monitor-session mon1</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RP/0/RP0/CPU0:router(config-mon)#</strong></td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> 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.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>destination next-hop ip address</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config-mon)# destination next-hop ipv4 254.23.24.5</strong></td>
</tr>
<tr>
<td></td>
<td>Configures the destination for the current monitor-session to be a next-hop IP address (whose type matches that of the monitor-session).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> 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.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config-mon)# exit</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RP/0/RP0/CPU0:router(config)#</strong></td>
</tr>
<tr>
<td></td>
<td>Exits monitor session configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>interface source-interface</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config)# interface gigabitethernet0/0/11</strong></td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>6</td>
<td>`ethernet-services access-group access-list-name [ingress</td>
</tr>
<tr>
<td></td>
<td>egress]`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# ethernet-services access-group acl1 ingress</td>
</tr>
<tr>
<td>7</td>
<td>`monitor-session session-name [ipv4</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 direction rx-only</td>
</tr>
<tr>
<td>8</td>
<td><code>acl</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if-mon)# acl</td>
</tr>
<tr>
<td>9</td>
<td><code>end</code> or <code>commit</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# end</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td>10</td>
<td><code>show monitor-session [session-name] status [detail] [error]</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show monitor-session</td>
</tr>
</tbody>
</table>

**Troubleshooting ACL-Based Traffic Mirroring**

Note the following 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, although traffic is mirrored, no access list configuration is applied.

This example correctly shows both the **capture** keyword in the ACL definition and the **acl** command configured on the interface:

```plaintext
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
    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, page 292
- Monitor Session Statistics: Example, page 293
- Layer 3 ACL-Based Traffic Mirroring: Example, page 293

Viewing Monitor Session Status: Example

This example shows sample output of the `show monitor-session` command with the **status** keyword:

```plaintext
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/RP0/CPU0:router# show monitor-session counters
Monitor-session msl
    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/RP0/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/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# monitor-session msl
RP/0/RP0/CPU0:router(config-mon)# destination next-hop 10.1.1.0
RP/0/RP0/CPU0:router(config-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface gig0/2/0/11
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group span ingress
RP/0/RP0/CPU0:router(config-if)# monitor-session msl
RP/0/RP0/CPU0:router(config-if-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 access-list span
RP/0/RP0/CPU0:router(config-ipv4-acl)# 5 permit ipv4 any any dscp 5 capture
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any
RP/0/RP0/CPU0:router(config-ipv4-acl)# commit
```

Troubleshooting Traffic Mirroring

When you have issues with your traffic mirroring, begin your 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
```
In the preceding example, the line marked as `<Session status>` can indicate one of these configuration errors:

<table>
<thead>
<tr>
<th>Session Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session is not configured globally</td>
<td>The session does not exist in global configuration. Check <code>show run</code> command output to ensure that a session with the right name has been configured.</td>
</tr>
<tr>
<td>Destination next-hop IPv4/IPv6 address <code>&lt;addr&gt;</code> is not configured</td>
<td>The IPv4 or IPv6 address that has been configured as the destination does not exist.</td>
</tr>
<tr>
<td>Destination next-hop IPv4 address <code>&lt;addr&gt;</code> not reachable</td>
<td>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 <code>show monitor-session status detail</code> command.</td>
</tr>
</tbody>
</table>

The `<Source interface status>` can report these messages:

<table>
<thead>
<tr>
<th>Source Interface Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational</td>
<td>Everything appears to be working correctly in traffic mirroring PL. Please follow up with the platform teams in the first instance, if mirroring is not operating as expected.</td>
</tr>
<tr>
<td>Not operational (Session is not configured globally)</td>
<td>The session does not exist in global configuration. Check the <code>show run</code> command output to ensure that a session with the right name has been configured.</td>
</tr>
<tr>
<td>Not operational (destination not known)</td>
<td>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).</td>
</tr>
<tr>
<td>Not operational (destination not active)</td>
<td>The destination interface or pseudowire is not in the Up state. See the corresponding <code>Session status</code> error messages for suggested resolution.</td>
</tr>
<tr>
<td>Not operational (source state <code>&lt;down-state&gt;</code>)</td>
<td>The source interface is not in the Up state. You can verify the state using the <code>show interfaces</code> 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).</td>
</tr>
<tr>
<td>Error: see detailed output for explanation</td>
<td>Traffic mirroring has encountered an error. Run the <code>show monitor-session status detail</code> command to display more information.</td>
</tr>
</tbody>
</table>
The `show monitor-session status detail` command displays full details of the configuration parameters, and of any errors encountered. For example:

```
RP/0/RP0/CPU0:router# show monitor-session status detail

Monitor-session foo
Destination next-hop GigabitEthernet 0/0/0/0
Source Interfaces-----------------
GigabitEthernet 0/1/0/0.100:
  Direction: Both
  Status:    Operating
GigabitEthernet 0/2/0/0.200:
  Direction: Tx
  Status:    Error: <blah>

Monitor session bar
No destination configured
Source Interfaces-----------------
GigabitEthernet 0/3/0/0.100:
  Direction: Rx
  Status:    Not operational(no destination)
```

This detailed output may give you a clear indication of what the problem is.

Here are additional trace and debug commands:

```
RP/0/RP0/CPU0:router# show monitor-session platform trace ?
  all     Turn on all the trace
  errors  Display errors
  events  Display interesting events

RP/0/RP0/CPU0:router# show monitor-session trace ?
  process  Filter debug by process

RP/0/RP0/CPU0:router# debug monitor-session platform ?
  all     Turn on all the debugs
  errors  CRS SPAN EA errors
  event   CRS SPAN EA event
  info    CRS SPAN EA info

RP/0/RP0/CPU0:router# debug monitor-session platform all
RP/0/RP0/CPU0:router# debug monitor-session platform event
RP/0/RP0/CPU0:router# debug monitor-session platform info
RP/0/RP0/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/RP0/CPU0:router# show monitor-session status
RP/0/RP0/CPU0:router# show monitor-session status errors
RP/0/RP0/CPU0:router# show monitor-session status internal
```
### Additional References

These sections provide references related to implementing traffic mirroring.

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td><em>Cisco IOS XR Master Commands List for the Cisco CRS Router</em></td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference for the Cisco CRS Router</em></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference for the Cisco CRS Router</em></td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>—</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchable technical content, including links to products, technologies,</td>
<td></td>
</tr>
<tr>
<td>solutions, technical tips, and tools. Registered Cisco.com users can</td>
<td></td>
</tr>
<tr>
<td>log in from this page to access even more content.</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Virtual Loopback and Null Interfaces on Cisco IOS XR Software

This module describes the configuration of loopback and null interfaces on the Cisco CRS Router. 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.

Feature History for Configuring Loopback and Null Interfaces on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>This module was updated to include information on configuring virtual IPV4 management interfaces.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Configuring Virtual Interfaces, page 300
- Information About Configuring Virtual Interfaces, page 300
- How to Configure Virtual Interfaces, page 302
- Configuration Examples for Virtual Interfaces, page 305
- Additional References, page 307
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, page 300
- Null Interface Overview, page 300
- Virtual Management Interface Overview, page 301
- Active and Standby RPs and Virtual Interface Configuration, page 301

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.

**Note**
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, page 302 (Required)
- Configuring Null Interfaces, page 303 (Required)
- Configuring Virtual IPV4 Interfaces, page 304 (Required)

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Enters interface configuration mode and names the new loopback interface.</td>
</tr>
<tr>
<td>interface loopback interface-path-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router#(config)# interface Loopback 3</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>Assigns an IP address and subnet mask to the virtual loopback interface using the ipv4 address configuration command.</td>
</tr>
<tr>
<td>ipv4 address ip-address</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38/32</td>
<td></td>
</tr>
</tbody>
</table>
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 type interface-path-id

Command or Action | Purpose
---|---
end or 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.

Example:

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

Step 5 show interfaces type interface-path-id

(Optional) Displays the configuration of the loopback interface.

Example:

RP/0/RP0/CPU0:router# show interfaces Loopback 3
### Configuring Virtual IPV4 Interfaces

This task explains how to configure an IPv4 virtual interface.

#### SUMMARY STEPS

1. `configure`  
2. `ipv4 address virtual address ip-address subnet mask`  
3. `end` or `commit`
Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Defines an IPv4 virtual address for the management Ethernet interface.</td>
</tr>
<tr>
<td><code>ipv4 address virtual address ipv4-address/mask</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-null0)# end</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-null0)# commit</td>
</tr>
</tbody>
</table>

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

  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

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

Configuration Examples for Virtual Interfaces

This section provides the following configuration examples:

- Configuring a Loopback Interface: Example, page 305
- Configuring a Null Interface: Example, page 306

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
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
```
Additional References

The following sections provide references related to loopback and null interface configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using the Cisco IOS XR Software.</td>
<td>Cisco IOS XR Getting Started Guide</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no applicable MIBs for this module.</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
## Technical Assistance

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

This module describes the configuration of clear channel SONET controllers on the Cisco CRS-1 Router.

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>Support was added on the Cisco CRS-1 Router for the Cisco 1-Port OC-768c/STM-256c POS PLIM.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>Support was added on the Cisco CRS-1 Router for the following hardware:</td>
</tr>
<tr>
<td></td>
<td>• Cisco 2 port OC-48/STM-16 POS SPA</td>
</tr>
<tr>
<td></td>
<td>• Cisco 4 port OC-48/STM-16 POS SPA</td>
</tr>
<tr>
<td>Release 3.8.0</td>
<td>The delay trigger line command was updated to line delay trigger in the</td>
</tr>
<tr>
<td></td>
<td>following sections:</td>
</tr>
<tr>
<td></td>
<td>• How to Configure Clear Channel SONET Controllers</td>
</tr>
<tr>
<td></td>
<td>• Configuring a Hold-off Timer to Prevent Fast Reroute from being Triggered</td>
</tr>
</tbody>
</table>
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

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

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
Information About Configuring SONET Controllers

To configure SONET controllers, you must understand the following concepts:

- SONET Controller Overview, page 311
- Default Configuration Values for SONET Controllers, page 312
- SONET APS, page 313

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

Table 6 describes some default configuration parameters that are present on SONET controllers.

### Table 6  SONET Controller Default Configuration Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Configuration File Entry</th>
</tr>
</thead>
</table>
| 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. |
| 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. |
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, page 314
- Configuring SONET APS, page 318
- Configuring a Hold-off Timer to Prevent Fast Reroute from Being Triggered, page 323

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.

Prerequisites

- 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” section on page 318.

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
# Configuring Clear Channel SONET Controllers on Cisco IOS XR Software

## How to Configure Clear Channel SONET Controllers

### DETAILED STEPS

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

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

<table>
<thead>
<tr>
<th><strong>Step 2</strong> controller sonet interface-path-id</th>
<th>Enters SONET controller configuration submode and specifies the SONET controller name and instance identifier with the rack/slot/module/port notation.</th>
</tr>
</thead>
</table>

**Example:**
```
RP/0/RP0/CPU0:router(config)# controller sonet 0/1/0/0
```

| **Step 3** clock source {internal | line} | 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.

<table>
<thead>
<tr>
<th><strong>Step 4</strong> line delay trigger value</th>
<th>(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.</th>
</tr>
</thead>
</table>

**Example:**
```
RP/0/RP0/CPU0:router(config-sonet)# line delay trigger 3000
```

<table>
<thead>
<tr>
<th><strong>Step 5</strong> line delay clear value</th>
<th>(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.</th>
</tr>
</thead>
</table>

**Example:**
```
RP/0/RP0/CPU0:router(config-sonet)# line delay clear 4000
```
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> framing {sdh</td>
<td>sonet}</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sonet)# framing sonet</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> loopback {internal</td>
<td>line}</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sonet)# loopback internal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> overhead {j0</td>
<td>s1s0} byte-value</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sonet)# overhead s1s0</td>
<td>• The default byte value for the <strong>j0</strong> keyword is 0xcc, and the default byte value for the <strong>s1s0</strong> keyword is 0.</td>
</tr>
<tr>
<td></td>
<td>• The range of valid values for <strong>j0</strong> and <strong>s1s0</strong> is 0 through 255.</td>
</tr>
</tbody>
</table>
Configuring Clear Channel SONET Controllers on Cisco IOS XR Software

How to Configure Clear Channel SONET Controllers

Step 9

**path** *keyword* [values]

(Optional) Configures SONET controller path values.

Example:

```
RP/0/RP0/CPU0:router(config-sonet)# path
delay trigger 25
```

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.
- **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.
Configuring Clear Channel SONET Controllers on Cisco IOS XR Software

How to Configure Clear Channel SONET Controllers

HC-318
Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router

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.

Prerequisites

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 restrictions:

- 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 APS group.
5.  exit
6.  interface loopback number
7.  ipv4 address ip-address mask
8.  exit
9.  interface pos interface-path-id
    or
    interface serial interface-path-id
10. ipv4 address ip-address mask
11. pos crc {16 | 32}
    or
    crc {16 | 32}
12. encapsulation {frame-relay | hdlc | ppp} (Serial interfaces only)
13. keepalive {interval | disable}[retry]
14. no shutdown
15. Repeat Step 9 to 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 to Step 19 for each channel of the group.
22. end
    or
    commit
23. exit
24. exit
25. show aps
26. show aps group [number]
# How to Configure Clear Channel SONET Controllers

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>aps group number</code></td>
<td>Adds an APS group with a specified number and enters APS group configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config)# aps group 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> To use the <code>aps group</code> command, you must be a member of a user group associated with the proper task IDs for <code>aps</code> commands.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> The <code>aps group</code> command is used even when a single protect group is configured.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> `channel {0</td>
<td>1} local sonet interface`</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config-aps)# channel 0 local SONET 0/0/0/1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> If the protect channel is local, it must be assigned using the <code>channel</code> command before any of the working channels is assigned.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> Repeat Step 3 for each channel in the group.</td>
<td>—</td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>exit</code></td>
<td>Exits APS group configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>interface loopback number</code></td>
<td>(Optional) Configures a loopback interface if a two-router APS is desired and enters interface configuration mode for a loopback interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config)# interface loopback 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> In this example, the loopback interface is used as the interconnect.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>ipv4 address ip-address mask</code></td>
<td>Assigns an IPV4 address and subnet mask to the loopback interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.0.1 255.255.255.224</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> <code>exit</code></td>
<td>Exits interface configuration mode for a loopback interface, and enters global configuration mode.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>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 <code>rack/slot/module/port/T3Num/T1num:instance</code> notation.</td>
</tr>
<tr>
<td><code>interface pos interface-path-id</code>&lt;br&gt;or&lt;br&gt;<code>interface serial interface-path-id</code></td>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config)# interface POS 0/2/0/0&lt;br&gt;or&lt;br&gt;RP/0/RP0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Assigns an IPv4 address and subnet mask to the interface.</td>
</tr>
<tr>
<td><code>ipv4 address ip-address mask</code></td>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.0.1 255.255.255.224</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>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.</td>
</tr>
<tr>
<td>`pos crc {16</td>
<td>32}<code>&lt;br&gt;or&lt;br&gt;</code>crc {16</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>(Serial interfaces only) Set the Layer 2 encapsulation of an interface.</td>
</tr>
<tr>
<td>`encapsulation {frame-relay</td>
<td>hdlc</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>Sets the keepalive timer for the channel, where:</td>
</tr>
<tr>
<td>`keepalive {interval</td>
<td>disable}[retry]`</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if)# keepalive disable</td>
<td><strong>Step 14</strong></td>
</tr>
<tr>
<td><code>no shutdown</code></td>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if)# no shutdown</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>Repeat Step 9 through Step 13 for each channel in the group.</td>
</tr>
<tr>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
### Configuring Clear Channel SONET Controllers

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 16</strong> exit</td>
<td>Exits interface configuration mode, and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 17</strong> controller sonet interface-path-id</td>
<td>Enters SONET controller configuration mode and specifies the SONET controller name and instance identifier with the rack/slot/module/port notation.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# controller sonet 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong> ais-shut</td>
<td>Configures SONET path values such as alarm indication signal (AIS) at shut down.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sonet)# ais-shut</td>
<td></td>
</tr>
<tr>
<td><strong>Step 19</strong> path scrambling disable</td>
<td>(Optional) Disables synchronous payload envelope (SPE) scrambling.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sonet)# path scrambling disable</td>
<td>Note SPE scrambling is enabled by default.</td>
</tr>
<tr>
<td><strong>Step 20</strong> clock source {internal</td>
<td>line}</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sonet)# clock source internal</td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• The line clock (line) is the default.</td>
</tr>
<tr>
<td><strong>Step 21</strong> Repeat Step 16 through Step 19 for each channel in the group.</td>
<td>—</td>
</tr>
<tr>
<td><strong>Step 22</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-sonet)# end or RP/0/RP0/CPU0:router(config-sonet)# commit</td>
<td>• When you issue the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
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.

Prerequisites

Configure SONET APS, as describe in the “Configuring SONET APS” section on page 318.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller sonet interface-path-id</td>
<td>Enters SONET configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# controller sonet 0/6/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> line delay trigger value or path delay trigger value</td>
<td>Configures SONET port delay trigger values in milliseconds.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sonet)# line delay trigger 250 or RP/0/RP0/CPU0:router(config-sonet)# path delay trigger 300</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sonet)# end or RP/0/RP0/CPU0:router(config-sonet)# commit</td>
<td></td>
</tr>
</tbody>
</table>

**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**.
Configuration Examples for SONET Controllers

This section contains the following examples:

- SONET Controller Configuration: Example, page 325
- SONET APS Group Configuration: Example, page 325

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 on page 314. 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 sls0 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
```
Additional References

The following sections provide references related to SONET controller configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using the Cisco IOS XR Software</td>
<td>Cisco IOS XR Getting Started Guide</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Configuring AAA Services on Cisco IOS XR Software module of Cisco IOS XR System Security Configuration Guide</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no applicable MIBs for this module.</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/support">http://www.cisco.com/support</a></td>
</tr>
</tbody>
</table>
Configuring Clear Channel T3/E3 Controllers on Cisco IOS XR Software

This module describes the configuration of clear channel T3/E3 controllers on the Cisco CRS Router. You must configure the T3/E3 controller before you can configure an associated serial interface.

Feature History for Configuring T3/E3 Controller Interfaces

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.4.1</td>
<td>This feature was introduced on the Cisco CRS-1 Router for the 4-Port Clear Channel T3/E3 SPA.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Configuring T3/E3 Controllers, page 327
- Information About T3/E3 Controllers and Serial Interfaces, page 327
- How to Configure Clear Channel T3/E3 Controllers, page 329
- Configuring Serial Interfaces on Cisco IOS XR Software Configuration Examples, page 346
- Additional References, page 348

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:

- Configuration Overview, page 328
- Default Configuration Values for T3 and E3 Controllers, page 329

Cisco 2-Port and 4-Port Clear Channel T3/E3 SPA

This section describes the types of loopback supported on the 2-Port and 4-Port Clear Channel T3/E3 SPA:

- Local loopback
- Network payload loopback (Configure the local framer to send all data received from the remote side back to the remote side.)
- Network line loopback (Configure the local LIU to send all data received from the remote side back to the remote side.)
- Remote line loopback (Use FEAC to request the remote interface to loop back to SPA—T3 only)

Configuration Overview

Configuring a channelized T3 controller and its associated serial interfaces is a 4-step process:

Step 1 Configure the T3 controller, and set the mode for that controller to T1 or E1.
Step 2 Configure the T1 or E1 controller.
Step 3  Create channel groups and assign DS0 time slots to these channel groups as desired.

Step 4  Configure the serial interfaces that are associated with the individual channel groups, as described in the Configuring Serial Interfaces on Cisco IOS XR Software module later in this document.

Default Configuration Values for T3 and E3 Controllers

Table 7 describes the default configuration parameters that are present on the T3 and E3 controllers.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Configuration File Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame type for the data line</td>
<td>For T3: C-bit framing</td>
<td>framing {auto-detect</td>
</tr>
<tr>
<td></td>
<td>For E3: G.751</td>
<td></td>
</tr>
<tr>
<td>Clocking for individual T3/E3 links</td>
<td>internal</td>
<td>clock source {internal</td>
</tr>
<tr>
<td>Cable length</td>
<td>224 feet</td>
<td>cablelength feet</td>
</tr>
<tr>
<td>Maintenance data link (MDL) messages (T3 only)</td>
<td>disable</td>
<td>mdl transmit {idle-signal</td>
</tr>
<tr>
<td>National reserved bits for an E3 port (E3 only)</td>
<td>enable, and the bit pattern value is 1.</td>
<td>national bits {disable</td>
</tr>
</tbody>
</table>

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, page 330
- Configuring a Clear Channel E3 Controller, page 332
- Modifying the Default E3 Controller Configuration, page 334
- Configuring a Clear Channel T3 Controller, page 337
- Modifying the Default T3 Controller Configuration, page 338
- Configuring Serial Interfaces on Cisco IOS XR Software, page 343
- Configuring BERT, page 343
How to Configure Clear Channel T3/E3 Controllers

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.

Prerequisites

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 [e3 | t3]` 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 { e3 | t3 }
3. end
   or
   commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> hw-module subslot subslot-id cardtype {e3</td>
<td>t3}</td>
</tr>
<tr>
<td><strong>Examples:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# hw-module subslot 0/1/0 cardtype e3</td>
<td></td>
</tr>
</tbody>
</table>

- **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.


How to Configure Clear Channel T3/E3 Controllers

When an E3 controller is in clear channel mode, it carries a single serial interface. The E3 controllers are configured using the E3 configuration mode.

Prerequisites

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

Command or Action | Purpose
--- | ---
**Step 3** end or commit | Saves configuration changes.

Example:
```
RP/0/RP0/CPU0:router(config)# end
```

```
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.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>configure</th>
<th>Enters global configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>controller e3 interface-path-id</td>
<td>Specifies the E3 controller name in the notation rack/slot/module/port and enters E3 configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>mode serial</td>
<td>Configures the mode of the port to be clear channel serial.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-e3)# mode serial</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>no shutdown</td>
<td>Removes the shutdown configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-e3)# no shutdown</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 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.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Configure Clear Channel T3/E3 Controllers

How to Configure Clear Channel T3/E3 Controllers

HC-334

Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router

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 Serial Interfaces on Cisco IOS XR Software” section later in this module.

- Configure the associated serial interface, as described in the Configuring Serial Interfaces on Cisco IOS XR Software 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.

Prerequisites

You must configure a clear channel E3 controller, as described in the “Configuring a Clear Channel E3 Controller” section earlier in this module.

---

### Command or Action | Purpose
---|---
**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.

---
### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller e3 interface-path-id</td>
<td>Specifies the E3 controller name in the notation rack/slot/module/port and enters E3 configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> clock source {internal</td>
<td>line}</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-e3)# clock source internal</td>
<td>Note: The default clock source is <strong>internal</strong>.</td>
</tr>
<tr>
<td><strong>Step 4</strong> cablelength feet</td>
<td>(Optional) Specifies the distance of the cable from the router to the network equipment.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-e3)# cablelength 250</td>
<td>Note: The default cable length is 224 feet.</td>
</tr>
<tr>
<td><strong>Step 5</strong> framing {g751</td>
<td>g832}</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-e3)# framing g832</td>
<td>Note: The default framing for E3 is G.751.</td>
</tr>
</tbody>
</table>
How to Configure Clear Channel T3/E3 Controllers

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 6    | national bits {disable | enable} | (Optional) Enables or disables the 0x1F national reserved bit pattern on the E3 port.  
   | Example:          | Note:   |
   |       | RP/0/RP0/CPU0:router(config-e3)# national bits enable | The E3 national bit is enabled by default, and the bit pattern value is 1. |
| 7    | no shutdown      | Removes the shutdown configuration.  
   | Example:          | - 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. |
|       | RP/0/RP0/CPU0:router(config-e3)# no shutdown | |
| 8    | end or           | Saves configuration changes.  
   |       | commit          | - When you issue the end command, the system prompts you to commit changes:  
   | Example:          |   - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.  
   |       | RP/0/RP0/CPU0:router(config-e3)# end       |   - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.  
   |       | or                                             |   - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.  
   |       | RP/0/RP0/CPU0:router(config-e3)# commit      | - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session. |
| 9    | show controllers e3 interface-path-id | (Optional) Displays information about the E3 controllers.  
   | Example:          |   | RP/0/RP0/CPU0:router# show controllers e3       | 0/1/0/0 |

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 Serial Interfaces on Cisco IOS XR Software Configuring Serial Interfaces on Cisco IOS XR Software Configuring BERT” section later in this module.
- Configure the associated serial interface, as described in the Configuring Serial Interfaces on Cisco IOS XR Software module later in this document.
Configuring a Clear Channel T3 Controller

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.

Prerequisites

You must use the `hw-module subslot cardtype` command to set the card to support T3, as described in the “Setting the Card Type” section 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller t3 interface-path-id</td>
<td>Specifies the T3 controller name in the <code>rack/slot/module/port</code> notation and enters T3 configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> mode serial</td>
<td>Configures the mode of the port to be clear channel serial.</td>
</tr>
<tr>
<td><strong>Note</strong> 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.</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Clear Channel T3/E3 Controllers on Cisco IOS XR Software

How to Configure Clear Channel T3/E3 Controllers

HC-338
Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router

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 Serial Interfaces on Cisco IOS XR Software Configuring Serial Interfaces on Cisco IOS XR Software Configuring BERT” section later in this module.
- Configure the associated serial interface, as described in the Configuring Serial Interfaces on Cisco IOS XR Software module.

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 on page 329.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4 no shutdown</td>
<td>Removes the shutdown configuration.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-t3)# no shutdown</td>
<td>• 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.</td>
</tr>
<tr>
<td>Step 5 end or commit</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>
| Example: RP/0/RP0/CPU0:router(config-t3)# end or RP/0/RP0/CPU0:router(config-t3)# commit | • When you issue the end command, the system prompts you to commit changes:
  - Uncommitted changes found, commit them before exiting(yes/no/cancel)?
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
  • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session. |
| Step 6 show controllers t3 interface-path-id | (Optional) Displays information about the T3 controllers. |
| Example: RP/0/RP0/CPU0:router# show controllers t3 0/1/0/0 | |
Prerequisites

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller T3 interface-path-id</td>
<td>Specifies the T3 controller name in the notation rack-slot/module/port and enters T3 configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> clock source {internal</td>
<td>line}</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The default clock source is internal.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-t3)# clock source internal</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 4</th>
<th>cablelength feet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-t3)# cablelength 250</td>
</tr>
</tbody>
</table>

(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.

**Note**  
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.

<table>
<thead>
<tr>
<th>Step 8</th>
<th>no shutdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-t3)# no shutdown</td>
</tr>
</tbody>
</table>

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.
### 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 Serial Interfaces on Cisco IOS XR Software” section on page 343 later in this module.

- Configure the associated serial interface, as described in the Configuring Serial Interfaces on Cisco IOS XR Software module.

### Detailed Steps

#### Step 9

**Command or Action**

- `end`
- `commit`

**Purpose**

Saves configuration changes.

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

  Uncommitted changes found, commit them before exiting(yes/no/cancel)?

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

**Example:**

```
RP/0/RP0/CPU0:router(config-t3)# end
or
RP/0/RP0/CPU0:router(config-t3)# commit
```

#### Step 10

**Command or Action**

- `show controllers t3 interface-path-id`

**Purpose**

(Optional) Displays information about the T3 controllers.

**Example:**

```
RP/0/RP0/CPU0:router# show controllers t3 0/1/0/0
```

---

### Step 1

**Command or Action**

- `show controllers t1 interface-path-id`

**Purpose**

(Optional) Displays information about the T1 controllers you created in Step 3.

**Example:**

```
RP/0//CPU0:router# show controllers t3 0/1/0/0
```

### Step 2

**Command or Action**

- `configure`

**Purpose**

Enters global configuration mode.

**Example:**

```
RP/0//CPU0:router# configure
```
### How to Configure Clear Channel T3/E3 Controllers

<table>
<thead>
<tr>
<th>Step 3</th>
<th><strong>controller tl interface-path-id</strong></th>
<th>Enters T1 configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config)# controller tl 0/3/0/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**framing {sf</td>
<td>esf}**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• sf—Superframe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• esf—Extended super frame</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config-t1)# framing esf</td>
<td>Note: The default frame type for T1 is Extended superframe (esf).</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>**yellow {detection</td>
<td>generation} {disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Yellow alarms are detected and generated on the T1 channel by default.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config-t1e1)# yellow detection enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>**clock source {internal</td>
<td>line}**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The default clock source is <strong>internal</strong>.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config-t1e1)# clock source internal</td>
<td>Note: When configuring clocking on a serial link, you must configure one end to be <strong>internal</strong>, and the other end to be <strong>line</strong>. If you configure <strong>internal</strong> clocking on both ends of a connection, framing slips occur. If you configure <strong>line</strong> clocking on both ends of a connection, the line does not come up.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>**fdl {ansi</td>
<td>att} {enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: FDL ansi and att are disabled by default.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config-t1e1)# fdl ansi enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>no shutdown</strong></td>
<td>Removes the shutdown configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config-t1e1)# no shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>channel-group channel-group-number</strong></td>
<td>Creates a T1 channel group and enters channel group configuration mode for that channel group.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config-t1)# channel-group 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>timeslots range</strong></td>
<td>Associates one or more DS0 time slots to a channel group and creates an associated serial subinterface on that channel group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Range is from 1 to 24 time slots.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• You can assign all 24 time slots to a single channel group, or you can divide the time slots among several channel groups.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0//CPU0:router(config-t1-channel_group)# timeslots 7-12</td>
<td>Note: Each individual T1 controller supports a total of 24 DS0 time slots.</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Step 11</th>
<th>speed kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0//CPU0:router(config-t1e1-channel_group)# speed 64</td>
</tr>
<tr>
<td>(Optional) Specifies the speed of the DS0s in kilobits per second. Valid values are 56 and 64.</td>
<td></td>
</tr>
<tr>
<td>Note: The default speed is 64 kbps.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 12</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0//CPU0:router(config-t1-channel_group)# exit</td>
</tr>
<tr>
<td>Exits channel group configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 13</th>
<th>Repeat Step 9 through Step 12 to assign time slots to a channel group. Each controller can contain up to 24 time slots.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Step 14</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0//CPU0:router(config-t1)# exit</td>
</tr>
<tr>
<td>Exits T1 configuration mode and enters global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 15</th>
<th>Repeat Step 2 through Step 14 to assign more channel groups to a controller as desired.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Step 16</th>
<th>end or commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0//CPU0:router(config-t3)# end</td>
</tr>
<tr>
<td>RP/0//CPU0:router(config-t3)# commit</td>
<td></td>
</tr>
<tr>
<td>Saves configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• When you issue the end command, the system prompts you to commit changes:</td>
<td></td>
</tr>
<tr>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)?</td>
<td></td>
</tr>
<tr>
<td>[cancel]:</td>
<td></td>
</tr>
<tr>
<td>– Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>– Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>– Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>
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.

**Prerequisites**

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.

**SUMMARY STEPS**

1. `configure`
2. `controller [t3 | e3] interface-path-id`
3. `pattern pattern`
4. `bert interval time`
5. `bert error [number]`
6. `end`
   or
7. `commit`
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>controller [t3</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# controller t3 0/1/0/0</td>
<td>Specifies the controller name and instance in the notation rack/slot/module/port, and enters T3 or E3 controller configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>bert pattern pattern</td>
</tr>
</tbody>
</table>
| 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.  
- Use the *commit* command to save the configuration changes to the running configuration file and remain within the configuration session.
What to Do Next

Configure the serial interfaces that are associate with the controllers you tested, as described in the "Configuring Serial Interfaces on Cisco IOS XR Software" module.

---

### Configuring Serial Interfaces on Cisco IOS XR Software Configuration Examples

This section contains the following examples:

- Configuring a Clear Channel T3 Controller: Example, page 346
- Configuring BERT on a T3 Controller: Example, page 347

#### 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 0/3/0/0
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
```
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# config
RP/0/RP0/CPU0(config)# controller t3 0/3/0/1
RP/0/RP0/CPU0(config-t3)# bert pattern 0s

Run bert from exec mode for the bert config to take effect

RP/0/RP0/CPU0(config-t3)# exit
RP/0/RP0/CPU0(config)# exit

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]
RP/0/RP0/CPU0# bert t3 0/3/0/1 start

RP/0/RP0/CPU0# bert t3 0/3/0/1 stop

RP/0/RP0/CPU0# 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
```
Additional References

The following sections provide references related to T3/E3 and T1 controllers.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using Cisco IOS XR software</td>
<td>Cisco IOS XR Getting Started Guide</td>
</tr>
<tr>
<td>Cisco IOS XR AAA services configuration information</td>
<td>Cisco IOS XR System Security Configuration Guide and Cisco IOS XR System Security Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>• IF-MIB</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
<tr>
<td>• DS3-MIB</td>
<td></td>
</tr>
<tr>
<td>• CISCO-DS3-MIB</td>
<td></td>
</tr>
<tr>
<td>• DS1-MIB</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>Not supported on the 4-Port Clear Channel T3/E3 SPA.</td>
</tr>
<tr>
<td>• Entity MIBs</td>
<td></td>
</tr>
</tbody>
</table>
RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

Technical Assistance

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

This module describes the configuration of dense wavelength division multiplexing (DWDM) controllers on the Cisco CRS-1 Router.

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.3.0</td>
<td>This feature was introduced on the Cisco 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.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>Support was added for user configuration of the laser, TTI strings, and BDI insertion, as well as performance monitoring.</td>
</tr>
<tr>
<td>Release 3.9.1</td>
<td>Support for IPoDWDM was added.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Configuring DWDM Controller Interfaces, page 352
- Information About the DWDM Controllers, page 352
- Information about IPoDWDM, page 353
- How to Configure DWDM Controllers, page 355
- How to Perform Performance Monitoring on DWDM Controllers, page 360
- Configuring IPoDWDM, page 364
- Configuration Examples, page 370
- Additional References, page 373
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 the following 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 28.

![Figure 28 OTN Optical Channel Structure](image)

**Information about IPoDWDM**

Cisco IOS XR software includes the IP over Dense Wavelength Division Multiplexing (IPoDWDM) feature.

IPoDWDM is supported on the following 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 the following 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 SRLG and Optical Layer DWDM Ports, page 364.
- Administrative state of DWDM optical ports, see Configuring the Administrative State of DWDM Optical Ports, page 366.
- FEC-FRR trigger threshold, window size, revert threshold, and revert window size, see Configuring Proactive FEC-FRR Triggering, page 368.

**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:

- Configuring the Optical Parameters, page 355
- Configuring G.709 Parameters, page 357

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.

Prerequisites

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 ]`
# Configuring Dense Wavelength Division Multiplexing Controllers on Cisco IOS XR Software

## How to Configure DWDM Controllers

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller dwdm interface-path-id</td>
<td>Specifies the DWDM controller name in the notation rack/slot/module/port and enters DWDM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> admin-state {maintenance</td>
<td>out-of-service}</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# admin-state maintenance</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td>Saves configuration changes. This performs the shutdown from the previous step. When the controller has been shut down, you can proceed with the configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> rx-los-threshold power-level</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# rx-los-threshold -10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> wavelength channel-number</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# wavelength 1</td>
<td><strong>Note</strong> 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.</td>
</tr>
<tr>
<td><strong>Step 7</strong> transmit-power power-level</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# transmit-power 10</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure DWDM Controllers

**Command or Action**

**Step 8**

```
end
or
commit
```

**Example:**

```
RP/0/RP0/CPU0:Router(config-dwdm)# end
or
RP/0/RP0/CPU0:Router(config-dwdm)# commit
```

**Purpose**

Saves configuration changes.

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

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

**Step 9**

```
admin-state in-service
```

**Example:**

```
RP/0/RP0/CPU0:Router(config-dwdm)# admin-state in-service
```

**Purpose**

Places the DWDM port in In Service (IS) state, to support all normal operation.

**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
```

**Purpose**

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**

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.

**Prerequisites**

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 | enhanced | 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller dwdm interface-path-id</td>
<td>Specifies the DWDM controller name in the notation rack/slot/module/port and enters DWDM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/0</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 3** admin-state maintenance
or
admin-state out-of-service | Disables the DWDM controller. You must disable the controller before you can use the DWDM configuration commands. |
| **Example:** RP/0/RP0/CPU0:Router(config-dwdm)# admin-state out-of-service | |
| **Step 4** 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. |
| **Example:** RP/0/RP0/CPU0:Router(config-dwdm)# commit | |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><code>g709 disable</code></td>
<td>(Optional) Disables the G.709 wrapper. The wrapper is enabled by default. <strong>Note</strong> The <code>g709 disable</code> command is available on the Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM only.</td>
</tr>
<tr>
<td>6</td>
<td>`loopback {internal</td>
<td>line}`</td>
</tr>
<tr>
<td>7</td>
<td>`g709 fec {disable</td>
<td>standard}`</td>
</tr>
<tr>
<td>8</td>
<td>`g709 {odu</td>
<td>otu} report alarm disable`</td>
</tr>
<tr>
<td>9</td>
<td>`g709 otu overhead tti {expected</td>
<td>sent} {ascii</td>
</tr>
</tbody>
</table>
| 10   | `end` or `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. |
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, page 360

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

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.
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`
10. `show controllers dwdm interface-path-id pm history [15-min | 24-hour | fec | optics | otn]`
11. `show controllers dwdm interface-path-id pm interval [15-min | 24-hour] [fec | optics | otn] index`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:Router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>controller dwdm interface-path-id</td>
<td>Specifies the DWDM controller name in the notation rack/slot/module/port and enters DWDM configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>controller dwdm</code> 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>`pm {15-min</td>
<td>24-hour} fec threshold {ec-bits</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>pm 15-min fec threshold ec-bits 49000000</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>pm 15-min fec threshold uc-words xxxx</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min fec threshold ec-bits 49000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min fec threshold uc-words xxxx</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>`pm {15-min</td>
<td>24-hour} optics threshold {lbc</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>pm 15-min optics 0pt max xxx</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>pm 15-min optics threshold lbc min xxx</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opc max xxx</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold lbc min xxx</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Dense Wavelength Division Multiplexing Controllers on Cisco IOS XR Software

How to Perform Performance Monitoring on DWDM Controllers

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`pm {15-min</td>
<td>24-hour} otn threshold otn-parameter threshold`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Rp/0/RP0/CPU0:Router(config-dwdm)# pm 15-min otn threshold bbe-pm-ne xxx</td>
<td>• bbe-pm-fe—Far-end path monitoring background block errors (BBE-PM)</td>
</tr>
<tr>
<td>Rp/0/RP0/CPU0:Router(config-dwdm)# pm 15-min otn threshold es-sm-fe xxx</td>
<td>• bbe-pm-ne—Near-end path monitoring background block errors (BBE-PM)</td>
</tr>
<tr>
<td></td>
<td>• bbe-sm-fe—Far-end section monitoring background block errors (BBE-SM)</td>
</tr>
<tr>
<td></td>
<td>• bbe-sm-ne—Near-end section monitoring background block errors (BBE-SM)</td>
</tr>
<tr>
<td></td>
<td>• bber-pm-fe—Far-end path monitoring background block errors ratio (BBER-PM)</td>
</tr>
<tr>
<td></td>
<td>• bber-pm-ne—Near-end path monitoring background block errors ratio (BBER-PM)</td>
</tr>
<tr>
<td></td>
<td>• bber-sm-fe—Far-end section monitoring background block errors ratio (BBER-SM)</td>
</tr>
<tr>
<td></td>
<td>• bber-sm-ne—Near-end section monitoring background block errors ratio (BBER-SM)</td>
</tr>
<tr>
<td></td>
<td>• es-pm-fe—Far-end path monitoring errored seconds (ES-PM)</td>
</tr>
<tr>
<td></td>
<td>• es-pm-ne—Near-end path monitoring errored seconds (ES-PM)</td>
</tr>
<tr>
<td></td>
<td>• es-sm-fe—Far-end section monitoring errored seconds (ES-SM)</td>
</tr>
<tr>
<td></td>
<td>• es-sm-ne—Near-end section monitoring errored seconds (ES-SM)</td>
</tr>
<tr>
<td></td>
<td>• esr-pm-fe—Far-end path monitoring errored seconds ratio (ESR-PM)</td>
</tr>
<tr>
<td></td>
<td>• esr-pm-ne—Near-end path monitoring errored seconds ratio (ESR-PM)</td>
</tr>
<tr>
<td></td>
<td>• esr-sm-fe—Far-end section monitoring errored seconds ratio (ESR-SM)</td>
</tr>
<tr>
<td></td>
<td>• esr-sm-ne—Near-end section monitoring errored seconds ratio (ESR-SM)</td>
</tr>
<tr>
<td></td>
<td>• fc-pm-fe—Far-end path monitoring failure counts (FC-PM)</td>
</tr>
<tr>
<td></td>
<td>• fc-pm-ne—Near-end path monitoring failure counts (FC-PM)</td>
</tr>
<tr>
<td></td>
<td>• fc-sm-fe—Far-end section monitoring failure counts (FC-SM)</td>
</tr>
<tr>
<td></td>
<td>• fc-sm-ne—Near-end section monitoring failure counts (FC-SM)</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
<tr>
<td>6</td>
<td>`pm {15-min</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> 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</td>
</tr>
<tr>
<td>7</td>
<td>`pm {15-min</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> 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</td>
</tr>
</tbody>
</table>
Configuring IPoDWDM

This section provides the following configuration procedures:

- Configuring the SRLG and Optical Layer DWDM Ports, page 364
- Configuring the Administrative State of DWDM Optical Ports, page 366
- Configuring Proactive FEC-FRR Triggering, page 368

Configuring the SRLG and Optical Layer DWDM Ports

Use the following 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:Router# config</td>
<td></td>
</tr>
<tr>
<td>Step 2 controller dwdm interface-path-id</td>
<td>Specifies the DWDM controller and enters DWDM controller mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1</td>
<td></td>
</tr>
<tr>
<td>Step 3 network srlg value1 value2 value3</td>
<td>Configures the Shared Risk Link Group (SRLG).</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:Router(config-dwdm)# network srlg value1 value2 value3</td>
<td></td>
</tr>
<tr>
<td>Step 4 network port id id-number</td>
<td>Assigns an identifier number to a port for the Multi Service Transport Protocol (MSTP).</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:Router(config-dwdm)# network port id 1/0/1/1</td>
<td></td>
</tr>
</tbody>
</table>
**Configuring the Administrative State of DWDM Optical Ports**

Use the following 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`
   or
   `interface tengige interface-path-id`
6.  `maintenance disable`
7.  `end`
   or
   `commit`

---

**Command or Action**

**Purpose**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 5  <code>network connection id id-number</code></td>
<td>Configures a connection identifier for the Multi Service Transport Protocol (MSTP).</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:Router(config-dwdm)# network connection id 1/1/1</code></td>
<td></td>
</tr>
<tr>
<td>Step 6  <code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:Router(config-dwdm)# end</code> or <code>RP/0/RP0/CPU0:Router(config-dwdm)# commit</code></td>
<td>• When you issue the <code>end</code> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router# config</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller dwdm interface-path-id</td>
<td>Specifies the DWDM controller and enters DWDM controller mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> admin-state {in-service</td>
<td>maintenance</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# admin-state maintenance</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits to the previous mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> interface pos interface-path-id or interface tengige interface-path-id</td>
<td>Specifies the interface and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> 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</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Dense Wavelength Division Multiplexing Controllers on Cisco IOS XR Software

Configuring IPoDWDM

HC-368
Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router

Configuring Proactive FEC-FRR Triggering

Use the following 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

Step 6

**Command or Action**

maintenance disable

**Purpose**

Provisions the maintenance embargo flag, which prevents maintenance activities from being performed on an interface.

**Example:**

RP/0/RP0/CPU0:Router(config-if)# maintenance disable

Step 7

**Command or Action**

end
  or
  commit

**Purpose**

Saves configuration changes.

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

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

Step 8

**Command or Action**

maintenance disable

**Purpose**

Provisions the maintenance embargo flag, which prevents maintenance activities from being performed on an interface.

**Example:**

RP/0/RP0/CPU0:Router(config-if)# maintenance disable

Step 9

**Command or Action**

end
  or
  commit

**Purpose**

Saves configuration changes.

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

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
RP/0/RP0/CPU0:Router# config

<table>
<thead>
<tr>
<th><strong>Step 2</strong> controller dwdm interface-path-id</th>
<th>Specifies the DWDM controller and enters DWDM controller mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 3</strong> proactive</th>
<th>Enables automatic triggering of FEC-FRR.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# proactive enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 4</strong> logging signal file-name</th>
<th>Enables 10 millisecond proactive monitoring of FEC-FRR.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# logging signal LogFile1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 5</strong> proactive trigger threshold x-coefficient y-power</th>
<th>Configures the trigger threshold of FEC-FRR in the form of ( x \times 10^{-y} ).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# proactive trigger threshold 1 9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 6</strong> proactive trigger window window</th>
<th>Configures the trigger window (in milliseconds) in which FRR may be triggered.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# proactive trigger window 10000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 7</strong> proactive revert threshold x-coefficient y-power</th>
<th>Configures the revert threshold (in the form of ( x \times 10^{-y} )) to trigger reverting from the FEC-FRR route back to the original route.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:Router(config-dwdm)# proactive revert threshold 1 9</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Dense Wavelength Division Multiplexing Controllers on Cisco IOS XR Software

Configuration Examples

This section includes the following examples:

- Turning On the Laser: Example, page 370
- Turning Off the Laser: Example, page 371
- DWDM Controller Configuration: Examples, page 371
- DWDM Performance Monitoring: Examples, page 371
- IPoDWDM Configuration: Examples, page 372

Turning On the Laser: Example

This is a required configuration beginning in Cisco IOS XR Software Release 3.9.0. The DWDM cards will not operate without this configuration.

The following example shows how to turn on the laser and place a DWDM port in In Service (IS) state:

```
RP/0/RP0/CPU0:Router(config-dwdm)# 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

This configuration replaces the laser off and shutdown (DWDM) configuration commands.

The following 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

The following 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
```

The following 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 0/2/0/0
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

The following 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
```
**Configuration Examples**

**IPoDWDM Configuration: Examples**

This section includes the following examples:

- SRLG and Optical Layer DWDM Port Configuration: Examples, page 372
- Administrative State of DWDM Optical Ports Configuration: Examples, page 373
- Proactive FEC-FRR Triggering Configuration: Examples, page 373

**SRLG and Optical Layer DWDM Port Configuration: Examples**

The following 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

The following 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

Additional References

These sections provide references related to DWDM controller configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>
Configuring Dense Wavelength Division Multiplexing Controllers on Cisco IOS XR Software

Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial system bootup and configuration information for a router using Cisco IOS XR software</td>
<td><em>Cisco IOS XR Getting Started Guide</em></td>
</tr>
<tr>
<td>Cisco IOS XR AAA services configuration information</td>
<td><em>Cisco IOS XR System Security Configuration Guide</em> and <em>Cisco IOS XR System Security Command Reference</em></td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITU-T G.709/Y.1331</td>
<td>Interfaces for the optical transport network (OTN)</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
<tr>
<td>OTN-MIB</td>
<td>IPoDWDM MIB</td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
</tr>
</tbody>
</table>

Technical Assistance

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

This module describes the configuration of Packet-over-SONET/SDH (POS) interfaces on the Cisco CRS Router.

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 on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.2</td>
<td>Support for the following hardware was introduced on the Cisco CRS-1 Router:</td>
</tr>
<tr>
<td></td>
<td>• 1-Port OC-192c/STM-64 POS/RPR XFP SPA</td>
</tr>
<tr>
<td></td>
<td>• 4-Port OC-3c/STM-1 POS SPA</td>
</tr>
<tr>
<td></td>
<td>• SIP-800</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>Support for the 8-Port OC-12c/STM-4 POS SPA was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>Support was added on the Cisco CRS-1 Router for the following hardware:</td>
</tr>
<tr>
<td></td>
<td>• Cisco 2-port OC-48c/STM16c POS SPA</td>
</tr>
<tr>
<td></td>
<td>• Cisco 4-port OC-48c/STM16c POS SPA</td>
</tr>
<tr>
<td>Release 3.4.1</td>
<td>Support was added on the Cisco CRS-1 Router for the Cisco 1-Port OC-192c/STM-64 POS/RPR VSR Optics SPA.</td>
</tr>
</tbody>
</table>

### Contents

- Prerequisites for Configuring POS Interfaces, page 376
- Information About Configuring POS Interfaces, page 376
- How to Configure a POS Interface, page 379
- Configuration Examples for POS Interfaces, page 386
- Additional References, page 388
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 on Cisco IOS XR Software” module.
  - A DWDM controller, as described in the “Configuring Dense Wavelength Division Multiplexing Controllers on Cisco IOS XR Software” 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:

- Cisco HDLC Encapsulation, page 377
- PPP Encapsulation, page 377
- Keepalive Timer, page 378
- Default Settings for POS Interfaces, page 376

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 Table 8 are present. These default settings can be changed by configuration.
Table 8  POS Modular Services Card and PLIM Default Interface Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration File Entry</th>
<th>Default Settings</th>
</tr>
</thead>
</table>
| 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 POS 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 enabled by default for POS interfaces.

Cisco HDLC uses keepalives to monitor the link state, as described in the “Keepalive Timer” section on page 378.

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:
Information About Configuring POS Interfaces

- 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” section on page 378.

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 on Cisco IOS XR Software” 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, page 379
- Configuring Optional POS Interface Parameters, page 382
- Modifying the Keepalive Interval on POS Interfaces, page 384

Bringing Up a POS Interface

This task describes the commands you can use to bring up a POS interface.

Prerequisites

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** `show interfaces` | (Optional) Displays configured interfaces.  
- Use this command to also confirm that the router recognizes the PLIM card. |
| **Example:**  
RP/0/RP0/CPU0:router# show interfaces | |
| **Step 2** `configure` | Enters global configuration mode. |
| **Example:**  
RP/0/RP0/CPU0:router# configure | |
| **Step 3** `interface pos interface-path-id` | Specifies the POS interface name and notation rack-slot/module/port, and enters interface configuration mode. |
| **Example:**  
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0 | |
| **Step 4** `ipv4 address ipv4_address/prefix` | 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. |
| **Example:**  
RP/0/RP0/CPU0:router (config)#ipv4 address 10.46.8.6/24 | |
| **Step 5** `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). |
| **Example:**  
RP/0/RP0/CPU0:router (config-if)# no shutdown | |
## Command or Action

### Step 6
- end
- commit

**Example:**
- RP/0/RP0/CPU0:router (config-if)# end
- RP/0/RP0/CPU0:router(config-if)# commit

**Purpose:** Saves configuration changes.
- When you issue the **end** command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?
  
  - 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

**Purpose:** Exits interface configuration mode and enters global configuration mode.

### Step 8
- exit

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

**Purpose:** Exits global configuration mode and enters EXEC mode.

### Step 9
- show interfaces
- configure
- interface pos interface-path-id
- no shut
- exit
- exit
- commit

**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

**Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.**

**Note** The configuration on both ends of the POS connection must match.
### Configuring POS Interfaces on Cisco IOS XR Software

**How to Configure a POS Interface**

To modify the default configuration of the POS interface you just brought up, see the “Configuring Optional POS Interface Parameters” section on page 382.

### Configuring Optional POS Interface Parameters

This task describes the commands you can use to modify the default configuration on a POS interface.

#### Prerequisites

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” section on page 379.

#### 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]`

### 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 382.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>interface pos interface-path-id</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0</td>
</tr>
<tr>
<td></td>
<td>Specifies the POS interface name and notation rack/slot/module/port, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>**encapsulation [hdlc</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# encapsulation hdlc</td>
</tr>
<tr>
<td></td>
<td>(Optional) Configures the interface encapsulation parameters and details such as HDLC or PPP.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The default encapsulation is hdlc.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**pos crc {16</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# pos crc 32</td>
</tr>
<tr>
<td></td>
<td>(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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> The default CRC is 32.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>mtu value</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# mtu 4474</td>
</tr>
<tr>
<td></td>
<td>(Optional) Configures the MTU value.</td>
</tr>
<tr>
<td></td>
<td>• The default value is 4474.</td>
</tr>
<tr>
<td></td>
<td>• The POS MTU range is 64–9216.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>end</strong> or <strong>commit</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router (config-if)# end or</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• When you issue the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
Configuring POS Interfaces on Cisco IOS XR Software

How to Configure a POS Interface

HC-384
Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router

What to Do Next

- To configure PPP authentication on POS interfaces where PPP encapsulation is enabled, see the Configuring PPP on Cisco IOS XR Software 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 384.

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.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7 exit</td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 8 exit</td>
<td>Exits global configuration mode and enters EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 9 show interfaces pos [interface-path-id]</td>
<td>(Optional) Displays general information for the specified POS interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show interface pos 0/3/0/0</td>
<td></td>
</tr>
</tbody>
</table>

What to Do Next

- To configure PPP authentication on POS interfaces where PPP encapsulation is enabled, see the Configuring PPP on Cisco IOS XR Software 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 384.

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.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7 exit</td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 8 exit</td>
<td>Exits global configuration mode and enters EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router (config)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 9 show interfaces pos [interface-path-id]</td>
<td>(Optional) Displays general information for the specified POS interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show interface pos 0/3/0/0</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisites

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” section on page 382.

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 type interface-path-id

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface pos interface-path-id</td>
<td>Specifies the POS interface name and notation rack/slot/module/port and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 3** keepalive {seconds [retry-count] | disable}
  or
  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. |
| Example: RP/0/RP0/CPU0:router(config-if)# keepalive 3 or RP/0/RP0/CPU0:router(config-if)# no keepalive |

- Use the keepalive disable command, the no keepalive, or the keepalive command with an argument of 0 to disable the keepalive feature entirely.
### Configuring POS Interfaces on Cisco IOS XR Software

**Configuration Examples for POS Interfaces**

This section provides the following configuration examples:

- **Bringing Up and Configuring a POS Interface with Cisco HDLC Encapsulation:** Example, page 386
- **Configuring a POS Interface with PPP Encapsulation:** Example, page 387

#### 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(config-if)# 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
```

---

**Step 4**

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

- 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show interfaces pos interface-path-id</code></td>
<td>(Optional) Verifies the interface configuration.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router# show interfaces POS 0/3/0/0
```

---

---
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
Additional References

These sections provide references related to POS interface configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td><em>Cisco IOS XR Master Commands List</em></td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference</em></td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using the</td>
<td><em>Cisco IOS XR Getting Started Guide</em></td>
</tr>
<tr>
<td>Cisco IOS XR software.</td>
<td></td>
</tr>
<tr>
<td>Cisco IOS XR AAA services configuration information</td>
<td><em>Cisco IOS XR System Security Configuration Guide</em> and</td>
</tr>
<tr>
<td></td>
<td><em>Cisco IOS XR System Security Command Reference</em></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference</em></td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRF.1.2</td>
<td>PVC User-to-Network Interface (UNI) Implementation Agreement - July 2000</td>
</tr>
<tr>
<td>ANSI T1.617 Annex D</td>
<td>—</td>
</tr>
<tr>
<td>ITU Q.933 Annex A</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1294</td>
<td>Multiprotocol Interconnect Over Frame Relay</td>
</tr>
<tr>
<td>RFC 1315</td>
<td>Management Information Base for Frame Relay DTEs</td>
</tr>
</tbody>
</table>
### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1490</td>
<td>Multiprotocol Interconnect Over Frame Relay</td>
</tr>
<tr>
<td>RFC 1586</td>
<td>Guidelines for Running OSPF Over Frame Relay Networks</td>
</tr>
<tr>
<td>RFC 1604</td>
<td>Definitions of Managed Objects for Frame Relay Service</td>
</tr>
<tr>
<td>RFC 2115</td>
<td>Management Information Base for Frame Relay DTEs Using SMIv2</td>
</tr>
<tr>
<td>RFC 2390</td>
<td>Inverse Address Resolution Protocol</td>
</tr>
<tr>
<td>RFC 2427</td>
<td>Multiprotocol Interconnect Over Frame Relay</td>
</tr>
<tr>
<td>RFC 2954</td>
<td>Definitions of Managed Objects for Frame Relay Service</td>
</tr>
</tbody>
</table>

### Technical Assistance

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

This module describes how to configure the Spatial Reuse Protocol (SRP) on supported Cisco Dynamic Packet Transport (DPT) interfaces on the Cisco CRS-1 Router.

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.2.2</td>
<td>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.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>This command was first supported on the 16-port OC-48c/STM-16c POS/DPT PLIM.</td>
</tr>
<tr>
<td>Release 3.5.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.6.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.7.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.8.0</td>
<td>Support for this feature was added on the Cisco CRS-1 Router for the following shared port adapters (SPAs):</td>
</tr>
<tr>
<td></td>
<td>• 1-port OC-192/STM-64 POS/RPR SPA XFP optics</td>
</tr>
<tr>
<td></td>
<td>• 4-port OC-48/STM-16 POS/RPR SPA</td>
</tr>
<tr>
<td></td>
<td>• 2-port OC-48/STM-16 POS/RPR SPA</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>No modification.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Configuring SRP Interfaces, page 392
- Information About Configuring SRP Interfaces, page 392
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. Figure 29 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 Figure 29, 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 29** 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, page 394** (required if using an SRP-supported PLIM)
- **Enabling SRP on an OC-48/STM-16 SPA Port, page 396** (required if using an SRP-supported OC-48/STM-16 SPA)
Enabling SRP on an OC-192/STM-64 SPA Port, page 398 (required if using an SRP-supported OC-192/STM-64 SPA)

• Creating a Basic SRP Configuration, page 401 (required)
• Configuring Intelligent Protection Switching (IPS), page 403 (optional)
• Configuring Modular Quality of Service CLI (MQC) with SRP, page 406 (optional)
• Adding a Node to the Ring, page 410 (optional)

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> hw-module port port-number-1 srp location instance</td>
<td>Enables SRP functionality on the first port.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 3</strong> hw-module port port-number-2 srp location instance</td>
<td>Enables SRP functionality on the second port.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module port 1 srp location 0/5/cpu0</td>
<td>Note: An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd.</td>
</tr>
<tr>
<td><strong>Step 4</strong> hw-module port port-number-3 srp location instance</td>
<td>Enables SRP functionality on the third port, for 16-port OC-48c/STM-16c POS/DPT PLIMs.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module port 2 srp location 0/5/cpu0</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 5</strong> hw-module port port-number-4 srp location instance</td>
<td>Enables SRP functionality on the fourth port, for 16-port OC-48c/STM-16c POS/DPT PLIMs.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module port 3 srp location 0/5/cpu0</td>
<td>Note: An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd.</td>
</tr>
<tr>
<td><strong>Note:</strong> 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.</td>
<td>Note: 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.</td>
</tr>
</tbody>
</table>
How to Configure an SRP Interface

Configuring SRP Interfaces on Cisco IOS XR Software

How to Configure an SRP Interface

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enables SRP functionality on the first port.</td>
</tr>
<tr>
<td>hw-module port port-number-1 srp location instance spa-bay number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>An SRP interface requires two consecutive physical ports for proper configuration. The first, lower-numbered port must be even, for example 0 or 2.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0 spa-bay 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables SRP functionality on the second port.</td>
</tr>
<tr>
<td>hw-module port port-number-2 srp location instance spa-bay number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# hw-module port 1 srp location 0/5/cpu0 spa-bay 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enables SRP functionality on the third port, for 4-port OC-48/STM-16 POS/RPR SPAs.</td>
</tr>
<tr>
<td>hw-module port port-number-3 srp location instance spa-bay number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>An SRP interface requires two consecutive physical ports for proper configuration. The first, lower-numbered port must be even; for example, 0 or 2.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# hw-module port 2 srp location 0/5/cpu0 spa-bay 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enables SRP functionality on the fourth port, for 4-port OC-48/STM-16 POS/RPR SPAs.</td>
</tr>
<tr>
<td>hw-module port port-number-4 srp location instance spa-bay number</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>An SRP interface requires two consecutive physical ports for proper configuration. The second, higher-numbered port must be odd.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# hw-module port 3 srp location 0/5/cpu0 spa-bay 2</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure an 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.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 6**  
end  
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**  
hw-module subslot subslot-id 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. |

Example:

| Example:  
RP/0/RP0/CPU0:router(config)# end  
RP/0/RP0/CPU0:router(config)# commit |  
Example:  
RP/0/RP0/CPU0:router# hw-module subslot 0/5/cpu0 reload |
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-1 shutdown
6. hw-module subslot subslot-id-2 shutdown
7. commit
8. no hw-module subslot subslot-id-1 shutdown
9. no hw-module subslot subslot-id-2 shutdown
10. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> hw-module port port-number-1 srp location instance spa-bay number</td>
<td>Enables SRP functionality on the SPA in the first bay.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0 spa-bay 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> hw-module port port-number-2 srp location instance spa-bay number</td>
<td>Enables SRP functionality on the SPA in the second bay.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module port 0 srp location 0/5/cpu0 spa-bay 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td>Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> hw-module subslot subslot-id shutdown</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module subslot 0/5/0 shutdown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shuts down the SPA in bay 0 of SRP location 0/5/cpu0.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6</td>
<td>hw-module subslot subslot-id shutdown</td>
<td>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 <code>hw-module subslot subslot-id reload</code> command to reload each SPA, because doing so can cause synchronization problems with the two SPAs and processes that comprise an SRP interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Shuts down the SPA in bay 1 of SRP location 0/5/cpu0.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# hw-module subslot 0/5/1 shutdown</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>commit</td>
<td>Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Returns the SPA in bay 0 of SRP location 0/5/cpu0 to the up state.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# commit</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>no hw-module subslot subslot-id shutdown</td>
<td>Returns the SPA in bay 1 of SRP location 0/5/cpu0 to the up state.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# no hw-module subslot 0/5/0 shutdown</td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>no hw-module subslot subslot-id shutdown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# no hw-module subslot 0/5/1 shutdown</td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>commit</td>
<td>Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# commit</td>
<td></td>
</tr>
</tbody>
</table>

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.
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.

- Configuring Single Ring Recovery on SRP Interfaces, page 416 (Optional)

### 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.

#### 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 mask`
6. `srp topology-timer value`
7. `no shutdown`
8. `end`
   or
9. `commit`
10. `show interfaces srp interface-path-id`
11. `show running-config`
## How to Configure an SRP Interface

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** show interfaces | (Optional) Displays configured interfaces.  
- Also use this command to confirm that the router recognizes the PLIM card. |
| Example: RP/0/RP0/CPU0:router# show interfaces | |
| **Step 2** configure | Enters global configuration mode. |
| Example: RP/0/RP0/CPU0:router# configure | |
| **Step 3** controller sonet interface-path-id 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.  
**Note** Refer to Configuring Clear Channel SONET Controllers on Cisco IOS XR Software for more detailed information on the SONET controller configuration. |
| 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 | |
| **Step 4** interface srp interface-path-id | Specifies the SRP interface name and notation rack/slot/module/port, and enters interface configuration mode. |
| Example: RP/0/RP0/CPU0:router(config)# interface srp 0/1/0/0 | |
| **Step 5** ipv4 address ip-address | Assigns an IP address and subnet mask to the interface. |
| Example: RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.1 255.255.255.224 | |
| **Step 6** srp topology-timer value | (Optional) Specifies how frequently topology discovery messages are sent around the ring to identify the current nodes on the SRP ring. |
| Example: RP/0/RP0/CPU0:router(config-if)# srp topology-timer 1 | |
| **Step 7** 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. |
| Example: RP/0/RP0/CPU0:router(config-if)# no shutdown | |
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 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.

Table 9 describes the IPS requests in the order of priority, from higher to lower.

---

**Table 9: IPS Requests**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>show running-config</code></td>
<td>(Optional) Displays the configuration information currently running on the router.</td>
</tr>
</tbody>
</table>

---

**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 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.

Table 9 describes the IPS requests in the order of priority, from higher to lower.
How to Configure an SRP Interface

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.

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`

---

### Table 9: Explanation of SRP IPS User Requests

<table>
<thead>
<tr>
<th>SRP IPS Request</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forced-switch</td>
<td>Adds a high-priority protection switch wrap on each end of a specified span by entering the user-configured <code>srp ips request forced-switch</code> command.</td>
</tr>
<tr>
<td>Manual-switch</td>
<td>Adds a low-priority protection switch wrap on each end of a specified span by entering the user-configured <code>srp request manual-switch</code> command.</td>
</tr>
</tbody>
</table>
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface srp interface-path-id</td>
<td>Specifies the SRP interface name in the notation rack/slot/module/port and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface srp 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> srp ips wtr-timer seconds</td>
<td>(Optional) Configures the amount of time in seconds that a wrap remains in place after the cause of the wrap is removed.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# srp ips wtr-timer 60</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> srp ips timer seconds</td>
<td>(Optional) Specifies the frequency of the transmission of IPS requests. The default is 1 second.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# srp ips timer 60 a</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>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 <code>srp ips timer</code> command.</td>
</tr>
<tr>
<td><strong>Step 5</strong> srp ips request forced-switch {a</td>
<td>b}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# srp ips request forced-switch a</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Use this command only as necessary, as it disables the node.</td>
</tr>
</tbody>
</table>
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 access-group-name
3. match mpls experimental topmost exp-value
4. exit

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>or commit</td>
<td>• When you issue the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>• Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# end</td>
</tr>
<tr>
<td></td>
<td>or RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td>Step 7</td>
<td></td>
</tr>
<tr>
<td>srp {request</td>
<td>remove} manual-switch {a</td>
</tr>
<tr>
<td></td>
<td>Note Use this command only as necessary.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# srp remove manual-switch a interface srp 0/1/0/0</td>
</tr>
<tr>
<td>Step 8</td>
<td></td>
</tr>
<tr>
<td>show srp ips interface srp interface-path-id</td>
<td>(Optional) Displays the IPS configuration on the SRP interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show srp ips interface srp 0/1/0/0</td>
</tr>
</tbody>
</table>
5. `class-map match-any access-group-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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> class-map match-any class-map-name</td>
<td>Enters class map configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# class-map match-any voice</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> match mpls experimental topmost exp-value</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cmap)# match mpls experimental topmost 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## How to Configure an SRP Interface

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> <code>exit</code></td>
<td>Exits the current submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-cmap)# exit</td>
<td></td>
</tr>
</tbody>
</table>

**Step 5** `class-map match-any class-map-name`

Enters class map configuration mode.
- Creates a class map to be used for matching packets to the class whose name you specify.
- 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.

**Example:** RP/0/RP0/CPU0:router(config)# class-map match-any ctrl

**Step 6** `match precedence precedence-value`

(Optional) Identifies IP precedence values as match criteria.
- The range is from 0 to 63.
- Reserved keywords can be specified instead of numeric values.

**Example:** RP/0/RP0/CPU0:router(config-cmap)# match precedence internet

**Step 7** `exit`

Exits the current submode.

**Example:** RP/0/RP0/CPU0:router(config-cmap)# exit

**Step 8** `policy-map policy-name`

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.

**Example:** RP/0/RP0/CPU0:router(config)# policy-map srp-policy

**Step 9** `class class-name`

Specifies the name of the class whose policy you want to create or change.

**Example:** RP/0/RP0/CPU0:router(config-pmap)# class voice

**Step 10** `police cir kbps`

Configures traffic policing.

**Note** 2000000 represents 10 percent of the interface line rate.

**Example:** RP/0/RP0/CPU0:router(config-pmap-c)# police cir 2000000

**Step 11** `set cos cos-value`

Sets the Layer 2 class of service (CoS) value of an outgoing packet.

**Example:** RP/0/RP0/CPU0:router(config-pmap-c)# set cos 4

**Step 12** `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.

**Example:** RP/0/RP0/CPU0:router(config-pmap-c)# priority

**Step 13** `exit`

Exits the current submode.

**Example:** RP/0/RP0/CPU0:router(config-pmap-c)# exit
### Command or Action

- **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:  
  `RP/0/RP0/CPU0:router(config-pmap-c)# exit`
  Exits the current submode.

- **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.
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. Figure 30 and Figure 31 show the physical configuration using a single DPT PLIM. Figure 32 and Figure 33 show the 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. interface srp interface-path-id
6. no srp ips request forced-switch {a | b}
### How to Configure an SRP Interface

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | configure
| Example: | Enters global configuration mode. |
| | RP/0/RP0/CPU0:router1# configure |
| **Step 2** | interface srp interface-path-id
| Example: | Specifies the SRP interface in the notation rack/slot/module/port for Node 1 and enters interface configuration mode. |
| | RP/0/RP0/CPU0:router1(config)# interface srp 0/1/0/0 |
| **Step 3** | srp ips request forced-switch {a | b}
| Example: | (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. |
| | RP/0/RP0/CPU0:router1(config-if)# srp ips request forced-switch a |
| **Step 4** | end or commit
| Example: | Saves configuration changes. |
| | RP/0/RP0/CPU0:router1(config-if)# end or RP/0/RP0/CPU0:router1(config-if)# commit |
| 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. |
| | • 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 SRP Interfaces on Cisco IOS XR Software

How to Configure an SRP Interface

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 Figure 33.

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 Figure 33.)

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
```

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 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 Figure 33.)

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 30  Four Routers on the SRP Ring (Each Router Using Two Physical Ports)
Figure 31 Adding a Router to an SRP Ring (Each Router Using Two Physical Ports)
Figure 32  SRP Ring Topology with Four Nodes

1  Cisco CRS-1 Router  2  Cisco 12000 Series Router

Figure 33  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.

Prerequisites

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 wrt-timer seconds
6. end
   or
   commit
7. show srp srr interface srp instance

DETAILED STEPS

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

Example:

```
RP/0/RP0/CPU0:router# configure
```
### Command or Action

**Step 2**  
`interface srp instance`

**Example:**
```
RP/0/RP0/CPU0:router(config)# interface srp 0/4/0/1
```

Enables 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 ring after the cause of the failure is removed. The default is 60 seconds.

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

**Example:**
```
RP/0/0/CPU0:router(config-if)# end  
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]:

  - 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, page 418
- Configuring Basic SRP: Example, page 418
- Configuring Modular QoS with SRP: Example, page 419

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
```
Configuring SRP Interfaces on Cisco IOS XR Software

Configuration Examples for SRP Interfaces

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). (See Figure 34.)

Note: Each SRP ring must be on a different subnet.
This configuration example shows the Cisco IOS commands used to configure SRP rings on the GSR+ A and GSR+ B routers in Figure 34.

**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
```

Figure 34  Two OC12 SRP Rings Connected to an OC48 SRP Ring

Note: These routers are members of the OC-12 SRP ring and the OC-48 SRP ring.
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 mrouting-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 mrouting-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 10.0.0.0 0.0.0.0 area 0
auto-cost reference-bandwidth 2488
!
ip classless

Additional References

These sections provide references related to SRP interface configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td><em>Cisco IOS XR Master Commands List</em></td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference</em></td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using the Cisco IOS XR software</td>
<td><em>Cisco IOS XR Getting Started Guide</em></td>
</tr>
<tr>
<td>Cisco IOS XR AAA services configuration information</td>
<td><em>Cisco IOS XR System Security Configuration Guide</em> and <em>Cisco IOS XR System Security Command Reference</em></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td><em>Cisco IOS XR Task ID Reference Guide</em></td>
</tr>
</tbody>
</table>
### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC-2892</td>
<td>The Cisco SRP MAC Layer Protocol</td>
</tr>
</tbody>
</table>

### Technical Assistance

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

This module describes the configuration of serial interfaces on the Cisco CRS Router. Before you configure a serial interface, you must configure the clear channel T3/E3 controller or channelized T1/E1 controller (DS0 channel) that is associated with that interface.

Feature History for Configuring Serial Controller Interfaces

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.4.1</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td></td>
<td>Support was added on the Cisco CRS-1 Router for the following hardware:</td>
</tr>
<tr>
<td></td>
<td>• Cisco CRS-1 SIP-800</td>
</tr>
<tr>
<td></td>
<td>• Cisco 2-Port and 4-Port Clear Channel T3/E3 SPA</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Configuring Serial Interfaces, page 425
- Information About Configuring Serial Interfaces, page 426
- How to Configure Serial Interfaces, page 433
- Configuration Examples for Serial Interfaces, page 450
- Additional References, page 453

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.
Information About Configuring Serial Interfaces

To configure serial interfaces, study the following concepts:

- **High-Level Overview: Serial Interface Configuration on Clear-Channel SPAs**, page 426
- **Cisco HDLC Encapsulation**, page 427
- **PPP Encapsulation**, page 427
- **Keepalive Timer**, page 429
- **Frame Relay Encapsulation**, page 430
- **Default Settings for Serial Interface Configurations**, page 431
- **Serial Interface Naming Notation**, page 432

High-Level Overview: Serial Interface Configuration on Clear-Channel SPAs

Table 10 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>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>Module</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Use the <code>hw-module subslot</code> command to set serial mode for the SPA to be T3, if necessary. <strong>Note</strong> By default, the 2-Port and 4-Port Clear Channel T3/E3 SPA is set to run in T3 mode.</td>
<td>“Configuring Clear Channel T3/E3 Controllers on Cisco IOS XR Software”</td>
<td>Setting the Card Type</td>
</tr>
<tr>
<td>2.</td>
<td>Configure the T3 controller.</td>
<td>“Configuring Clear Channel T3/E3 Controllers on Cisco IOS XR Software”</td>
<td>Setting the Card Type</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the serial interface that is associated with the T3 controller you configured in Step 2.</td>
<td>“Configuring Serial Interfaces on Cisco IOS XR Sotware”</td>
<td>“How to Configure Serial Interfaces”</td>
</tr>
</tbody>
</table>

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 on Cisco IOS XR Software” module in this manual.
Table 11 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 11: Overview: Configuring an E3 Serial Interface on a Clear Channel SPA

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>Module</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Use the <strong>hw-module subslot</strong> command to set serial mode for the SPA to be E3.</td>
<td>“Configuring Clear Channel T3/E3 Controllers on Cisco IOS XR Software”</td>
<td>Setting the Card Type</td>
</tr>
<tr>
<td>2.</td>
<td>Configure the E3 controller.</td>
<td>“Configuring Clear Channel T3/E3 Controllers on Cisco IOS XR Software”</td>
<td>Setting the Card Type</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the serial interface that is associated with the E3 controller you configured in Step 2.</td>
<td>Configuring Serial Interfaces on Cisco IOS XR Software</td>
<td>How to Configure Serial Interfaces</td>
</tr>
</tbody>
</table>

### 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**” section on page 429.

**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” section on page 429.

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 on Cisco IOS XR Software” 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 the following 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 on Cisco IOS XR Software” 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*. 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:
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 Table 12 are present. These default settings can be changed by configuration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration File Entry</th>
<th>Default Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keepalive</td>
<td>keepalive [disable]</td>
<td>keepalive 10 seconds</td>
</tr>
<tr>
<td>Note</td>
<td>no keepalive</td>
<td></td>
</tr>
<tr>
<td>Encapsulation</td>
<td>encapsulation [hdlc</td>
<td>hdlc</td>
</tr>
<tr>
<td></td>
<td>[ppp</td>
<td>frame-relay [ietf]]</td>
</tr>
</tbody>
</table>
### Serial Interface Default Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration File Entry</th>
<th>Default Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum transmission unit (MTU)</td>
<td><code>mtu bytes</code></td>
<td>1504 bytes</td>
</tr>
<tr>
<td>Cyclic redundancy check (CRC)</td>
<td>`crc [16</td>
<td>32]`</td>
</tr>
<tr>
<td>Data stream inversion on a serial interface</td>
<td><code>invert</code></td>
<td>Data stream is not inverted</td>
</tr>
<tr>
<td>Payload scrambling (encryption)</td>
<td><code>scramble</code></td>
<td>Scrambling is disabled.</td>
</tr>
<tr>
<td>Number of High-Level Data Link Control (HDLC) flag sequences to be inserted between the packets</td>
<td><code>transmit-delay</code></td>
<td>Default is 0 (disabled).</td>
</tr>
</tbody>
</table>

**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 the following 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.
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 on Cisco IOS XR Software” module in this document, you can configure the serial interfaces associated with that controller.

The following tasks describe how to configure a serial interface:

- Bringing Up a Serial Interface, page 433
- Configuring Optional Serial Interface Parameters, page 436
- Creating a Point-to-Point Serial Subinterface with a PVC, page 439
- Configuring Optional PVC Parameters, page 441
- Modifying the Keepalive Interval on Serial Interfaces, page 443
- How to Configure a Layer 2 Attachment Circuit, page 445
  - Creating a Serial Layer 2 Subinterface with a PVC, page 445
  - Configuring Optional Serial Layer 2 PVC Parameters, page 447

Bringing Up a Serial Interface

This task describes the commands used to bring up a serial interface.

Prerequisites

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** show interfaces | (Optional) Displays configured interfaces.  
- Use this command to also confirm that the router recognizes the PLIM card. |
| **Example:** RP/0/RP0/CPU0:router# show interfaces |
| **Step 2** configure | Enters global configuration mode. |
| **Example:** RP/0/RP0/CPU0:router# configure |
| **Step 3** interface serial interface-path-id | Specifies the serial interface name and notation rack/slot/module/port, and enters interface configuration mode. |
| **Example:** RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0 |
| **Step 4** ipv4 address ip-address | 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. |
| **Example:** RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.1 255.255.255.224 |
| **Step 5** 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). |
| **Example:** RP/0/RP0/CPU0:router (config-if)# no shutdown |
### Command or Action

**Step 6**

- `end`
- `commit`

**Example:**

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

Saves configuration changes.

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

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

Exits interface configuration mode and enters global configuration mode.

**Step 8**

- `exit`

**Example:**

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

Exits global configuration mode and enters EXEC mode.

**Step 9**

- `show interfaces`
- `configure`
- `interface serial interface-path-id`
- `no shut`
- `exit`

**Example:**

```plaintext
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
```

Repeat Step 1 through Step 8 to bring up the interface at the other end of the connection.

**Note**  
The configuration on both ends of the serial connection must match.
Configuring Serial Interfaces on Cisco IOS XR Software

How to Configure Serial Interfaces

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 436.

Configuring Optional Serial Interface Parameters

This task describes the commands used to modify the default configuration on a serial interface.

Prerequisites

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” section on page 433.

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 hdic-flags
9. end
   or
   commit
10. exit

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 10 show ipv4 interface brief</td>
<td>Verifies that the interface is active and properly configured. 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.”</td>
</tr>
<tr>
<td>Step 11 show interfaces serial interface-path-id</td>
<td>(Optional) Displays the interface configuration.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router # show ipv4 interface brief
```

```
RP/0/RP0/CPU0:router# show interfaces serial 0/1/0/0
```
# Configuring Serial Interfaces on Cisco IOS XR Software

## How to Configure Serial Interfaces

11. `exit`
12. `exit`
13. `show interfaces serial [interface-path-id]`

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface serial interface-path-id</td>
<td>Specifies the serial interface name and notation rack/slot/module/port, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> encapsulation [hdlc</td>
<td>ppp</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# encapsulation hdlc</td>
<td>Note The default encapsulation is hdlc.</td>
</tr>
<tr>
<td><strong>Step 4</strong> serial</td>
<td>(Optional) Enters serial submode to configure the serial parameters.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# serial</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> crc length</td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:ios(config-if-serial)# crc 32</td>
<td>Note The default is CRC length is 16.</td>
</tr>
<tr>
<td><strong>Step 6</strong> invert</td>
<td>(Optional) Inverts the data stream.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:ios(config-if-serial)# inverts</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> scramble</td>
<td>(Optional) Enables payload scrambling on the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:ios(config-if-serial)# scramble</td>
<td>Note Payload scrambling is disabled on the interface.</td>
</tr>
<tr>
<td><strong>Step 8</strong> transmit-delay hdlc-flags</td>
<td>(Optional) Specifies a transmit delay on the interface. Values can be from 0 to 128.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:ios(config-if-serial)# transmit-delay 10</td>
<td>Note Transmit delay is disabled by default (the transmit delay is set to 0).</td>
</tr>
</tbody>
</table>
## How to Configure Serial Interfaces

### Command or Action

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
|        | `end` or `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. |

| Example: | RP/0/RP0/CPU0:router (config-if)# end  
|          | or  
|          | RP/0/RP0/CPU0:router(config-if)# commit |

| Step 10 | `exit` | Exits serial configuration mode.  
| Example: | RP/0/RP0/CPU0:router(config-if-serial)# exit |

| Step 11 | `exit` | Exits interface configuration mode and enters global configuration mode.  
| Example: | RP/0/RP0/CPU0:router (config-if)# exit |

| Step 12 | `exit` | Exits global configuration mode and enters EXEC mode.  
| Example: | RP/0/RP0/CPU0:router (config)# exit |

| Step 13 | `show interfaces serial [interface-path-id]` | (Optional) Displays general information for the specified serial interface.  
| Example: | RP/0/RP0/CPU0:router# show interface serial 0/1/0/0 |

### 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 439.

- To configure PPP authentication on serial interfaces with PPP encapsulation, see the “Configuring PPP on Cisco IOS XR Software” module later in this manual.
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.

Prerequisites

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” section on page 433.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| configure                         | Example:
| RP/0/RP0/CPU0:router# configure   |                                              |
| Step 2                            | Enters serial subinterface configuration mode.|
| interface serial interface-path-id.subinterface point-to-point | Example:
| RP/0/RP0/CPU0:router (config)# interface serial 0/1/0/0.1 |
### Command or Action

<table>
<thead>
<tr>
<th>Step 3</th>
<th>ipv4 address ipv4_address/prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router (config-subif)#ipv4 address 10.46.8.6/24</td>
</tr>
</tbody>
</table>

Assigns an IP address and subnet mask to the subinterface.

<table>
<thead>
<tr>
<th>Step 4</th>
<th>pvc dlci</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router (config-subif)# pvc 20</td>
</tr>
</tbody>
</table>

Creates a serial permanent virtual circuit (PVC) and enters Frame Relay PVC configuration submode.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>end or commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router (config-subif)# end or RP/0/RP0/CPU0:router(config-subif)# commit</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>configure interface serial interface-path-id pvc dlci commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>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</td>
</tr>
</tbody>
</table>

Repeat Step 1 through Step 5 to bring up the serial subinterface and any associated PVC at the other end of the connection.

**Note** The DLCI (or PVC identifier) must match on both ends of the subinterface connection.

**Note** 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 436.
• 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.

**Prerequisites**

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” section on page 439.

**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. configure
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 configure the 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}

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**        | configure  
|                    | Enters global configuration mode. |
| **Example:**      | RP/0/RP0/CPU0:router# configure |
| **Step 2**        | interface serial interface-path-id.subinterface  
|                    | Enters serial subinterface configuration mode. |
| **Example:**      | RP/0/RP0/CPU0:router (config)# interface serial 0/1/0/0.1 |
## Command or Action

### 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
```
**Example:**
```
RP/0/RP0/CPU0:router (config-fr-vc)# end
```
Saves configuration changes.

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

  - 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
```
configure interface serial interface-path-id.subinterface
pvc dlci
encap [cisco | ietf]
commit
```
**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)# pvc 20
RP/0/RP0/CPU0:router (config-fr-vc)# encap cisco
RP/0/RP0/CPU0:router (config-fr-vc)# commit
```
Repeat Step 1 through Step 6 to bring up the serial subinterface and any associated PVC at the other end of the connection.

**Note** The configuration on both ends of the subinterface connection must match.
How to Configure Serial Interfaces

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.

Note

Cisco HDLC is enabled by default on serial interfaces.

Prerequisites

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” section on page 436.

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

Command or Action | Purpose
--- | ---
Step 8 | show frame-relay pvc dlci-number
Example: RP/0/RP0/CPU0:router# show frame-relay pvc 20
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) Verifies the configuration of specified serial interface.

(Optional) Displays the statistics and the configurations of the input and output policies that are attached to a subinterface.
### How to Configure Serial Interfaces

3. **keepalive** \{seconds | disable\}
   - or
   - no keepalive

4. end
   - or
   - commit

5. **show interfaces** type interface-path-id

#### DETAILED STEPS

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

**Example:**

RP/0/RP0/CPU0:router# configure

<table>
<thead>
<tr>
<th><strong>Step 2</strong> interface serial interface-path-id</th>
<th>Specifies the serial interface name and notation rack/slot/module/port and enters interface configuration mode.</th>
</tr>
</thead>
</table>

**Example:**

RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0

| **Step 3** keepalive \{seconds | disable\} | Specifies the number of seconds between keepalive messages. |
|--------------------------------------------|---------------------------------------------------------------|
| or no keepalive                            | • Use the **keepalive disable** command, the **no keepalive**, or the **keepalive** command with an argument of 0 to disable the keepalive feature. |

**Example:**

RP/0/RP0/CPU0:router(config-if)# keepalive 3
or
RP/0/RP0/CPU0:router(config-if)# no keepalive

- • 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.
### 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

### Creating a Serial Layer 2 Subinterface with a PVC

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

**Prerequisites**

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” section on page 433.
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>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2 interface serial interface-path-id.subinterface l2transport</td>
<td>Creates a subinterface and enters serial subinterface configuration mode for that subinterface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0.1 l2transport</td>
</tr>
<tr>
<td>Step 3 pvc vpi/vci</td>
<td>Creates a serial permanent virtual circuit (PVC) and enters serial Layer 2 transport PVC configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# pvc 5/20</td>
</tr>
</tbody>
</table>

Note Only one PVC is allowed per subinterface.
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.

Prerequisites

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” section on page 445.

What to Do Next

- To configure optional PVC parameters, see the “Configuring Optional Serial Layer 2 PVC Parameters” section on page 447.
- 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.
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface serial interface-path-id.subinterface l2transport</code></td>
<td>Enters serial subinterface configuration mode for a Layer 2 serial subinterface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface serial 0/1/0.1 l2transport</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>pvc dlci</code></td>
<td>Enters serial Frame Relay PVC configuration mode for the specified PVC.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# pvc 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> `encap [cisco</td>
<td>ietf]`</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-fr-vc)# encapsulation aal5</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> `service-policy {input</td>
<td>output} policy-map`</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router (config-subif)# service-policy output policy1</td>
<td></td>
</tr>
</tbody>
</table>
Step 6

Command or Action
end
or
commit

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

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

Example:
RP/0/RP0/CPU0:router(config-serial-l2transport-pvc)# end
or
RP/0/RP0/CPU0:router(config-serial-l2transport-pvc)# commit

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.

Step 8

Command or Action
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}

Purpose
(Optional) Displays the statistics and the configurations of the input and output policies that are attached to a subinterface.

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

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, page 450
- Configuring a Serial Interface with Frame Relay Encapsulation: Example, page 451
- Configuring a Serial Interface with PPP Encapsulation: Example, page 452

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#configure
RP/0/RP0/CPU0:router(config)#interface serial 0/3/0/0:0
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
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
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 runs, 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
```
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(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
```
**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
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
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 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
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
```
### Additional References

These sections provide references related to T3/E3 and T1/E1 controllers and serial interfaces.

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td><em>Cisco IOS XR Master Commands List</em></td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference</em></td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using Cisco IOS XR software</td>
<td><em>Cisco IOS XR Getting Started Guide</em></td>
</tr>
<tr>
<td>Cisco IOS XR AAA services configuration information</td>
<td><em>Cisco IOS XR System Security Configuration Guide</em> and <em>Cisco IOS XR System Security Command Reference</em></td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRF.1.2</td>
<td><em>PVC User-to-Network Interface (UNI) Implementation Agreement - July 2000</em></td>
</tr>
<tr>
<td>ANSI T1.617 Annex D</td>
<td>—</td>
</tr>
<tr>
<td>ITU Q.933 Annex A</td>
<td>—</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1294</td>
<td><em>Multiprotocol Interconnect Over Frame Relay</em></td>
</tr>
<tr>
<td>RFC 1315</td>
<td><em>Management Information Base for Frame Relay DTEs</em></td>
</tr>
<tr>
<td>RFC 1490</td>
<td><em>Multiprotocol Interconnect Over Frame Relay</em></td>
</tr>
</tbody>
</table>
## Additional References

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1586</td>
<td>Guidelines for Running OSPF Over Frame Relay Networks</td>
</tr>
<tr>
<td>RFC 1604</td>
<td>Definitions of Managed Objects for Frame Relay Service</td>
</tr>
<tr>
<td>RFC 2115</td>
<td>Management Information Base for Frame Relay DTEs Using SMIv2</td>
</tr>
<tr>
<td>RFC 2390</td>
<td>Inverse Address Resolution Protocol</td>
</tr>
<tr>
<td>RFC 2427</td>
<td>Multiprotocol Interconnect Over Frame Relay</td>
</tr>
<tr>
<td>RFC 2954</td>
<td>Definitions of Managed Objects for Frame Relay Service</td>
</tr>
</tbody>
</table>

### Technical Assistance

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

This module describes the configuration of Point-to-Point Protocol (PPP) on POS and serial interfaces on the Cisco CRS-1 Router.

This module describes how to perform these Point-to-Point Protocol (PPP) related tasks on POS and serial interfaces in Cisco IOS XR software:

- Enable and configure PPP authentication protocols
- Disable PPP authentication
- Modify optional PPP timeout and retry parameters
- Configure Multilink PPP (MLPPP)

Feature History for Configuring PPP Interfaces on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>PPP authentication was introduced on the Cisco CRS-1 Router.</td>
</tr>
</tbody>
</table>
| Release 4.0.1 | Support for MLPPP was added for the following SPAs:  
  - Cisco 1-Port Channelized OC-3/STM-1 SPA  
  - Cisco 2-Port and 4-Port Clear Channel T3/E3 SPA  
  - Cisco 4-Port OC-3c/STM-1 SPA  
  - Cisco 8-Port OC-3c/STM-1 SPA |

Contents

- Prerequisites for Configuring PPP, page 456
- Information About PPP, page 456
- How to Configure PPP, page 459
- Configuration Examples for PPP, page 486
- Additional References, page 492
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 Cisco IOS XR Software module in this manual.
  - To enable PPP encapsulation on a serial interface, see the Configuring Serial Interfaces on Cisco IOS XR Software module in this manual.

Information About PPP

To configure PPP and related features, you should understand the information in this section:

- PPP Authentication, page 456
- Multilink PPP, page 458

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 Reference* 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>The CHAP authenticator sends a challenge message to the peer.</td>
</tr>
<tr>
<td>Step 2</td>
<td>The peer responds with a value calculated through a one-way hash function.</td>
</tr>
<tr>
<td>Step 3</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

**Multilink PPP**

Multilink Point-to-Point Protocol (MLPPP) provides a method for combining multiple physical links into one logical link. The implementation combines multiple PPP interfaces into one multilink interface. MLPPP performs the fragmenting, reassembling, and sequencing of datagrams across multiple PPP links.

Link Fragmentation and Interleaving (LFI) is designed for MLPPP interfaces and is required when integrating voice and data on low-speed interfaces.

Link Fragmentation and Interleaving (LFI) provides stability for delay-sensitive traffic, such as voice or video, traveling on the same circuit as data. Voice is susceptible to increased latency and jitter when the network processes large packets on low-speed interfaces. LFI reduces delay and jitter by fragmenting large datagrams and interleaving them with low-delay traffic packets.

*Figure 35*  **Link Fragmentation Interleave**
Supported Cards

MLPPP is supported on the following line cards and SPAs:
- Cisco CRS multiservice line cards with SIP-800.
- 2-Port and 4-Port Channelized T3 SPAs (SPA-2XCT3/DS0, SPA-4XCT3/DS0)

LFI is supported on:
- Cisco 1-Port Channelized STM-1/OC-3 Shared Port Adapter

MLPPP Feature Summary

MLPPP in Cisco IOS XR 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 milliseconds.
- Minimum-active-links configuration option.
- LCP echo request/reply support over multilink interface.
- Full T1 and E1 framed and unframed links.

IPHC Over MLPPP

- The 8-Port Channelized T1/E1 SPA supports IPHC over MLPPP. For more information about IPHC and how to configure it, see the “Configuring Serial Interfaces on Cisco IOS XR Software” module in the Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router.

How to Configure PPP

This section includes the following procedures:
- Modifying the Default PPP Configuration, page 459
- Configuring PPP Authentication, page 463
- Disabling an Authentication Protocol, page 471
- Configuring Multilink PPP, page 476

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.
How to Configure PPP

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).

Prerequisites

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 Cisco IOS XR Software module in this manual.
- To enable PPP encapsulation on an interface, see the Configuring Serial Interfaces on Cisco IOS XR Software 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}`
# Configuring PPP on Cisco IOS XR Software

## How to Configure PPP

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface type interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ppp max-bad-auth retries</td>
<td>(Optional) Configures the number of authentication retries allowed on an interface after a PPP authentication failure.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# ppp max-bad-auth 3</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ppp max-configure retries</td>
<td>(Optional) Specifies the maximum number of configure requests to attempt (without response) before the requests are stopped.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# ppp max-configure 4</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ppp max-failure retries</td>
<td>(Optional) Configures the maximum number of consecutive Configure Negative Acknowledgments (CONFNAKs) permitted before a negotiation is terminated.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# ppp max-failure 3</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 6</th>
<th>ppp max-terminate number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ppp max-terminate 5</td>
</tr>
</tbody>
</table>

(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

<table>
<thead>
<tr>
<th>ppp timeout authentication seconds</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>ppp timeout retry seconds</th>
</tr>
</thead>
</table>

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.
### Configuring PPP Authentication

This section contains the following procedures:

- Enabling PAP, CHAP, and MS-CHAP Authentication, page 463
- Configuring a PAP Authentication Password, page 466
- Configuring a CHAP Authentication Password, page 468
- Configuring an MS-CHAP Authentication Password, page 470

### 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.

### Prerequisites

You must enable PPP encapsulation on the interface with the `encapsulation ppp` command, as described in the following modules:

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 9**
end or 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 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} | Verifies the PPP configuration for an interface or for all interfaces that have PPP encapsulation enabled. |

Example:

RP/0/RP0/CPU0:router# show ppp interfaces
serial 0/2/0/0

---

---
• To enable PPP encapsulation on a POS interface, see the Configuring POS Interfaces on Cisco IOS XR Software module in this manual.
• To enable PPP encapsulation on an interface, see the Configuring Serial Interfaces on Cisco IOS XR Software 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2 interface</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>type</td>
<td>Example:</td>
</tr>
<tr>
<td>interface-path-id</td>
<td>RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</td>
</tr>
<tr>
<td>Step 3 ppp</td>
<td>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.</td>
</tr>
<tr>
<td>authentication</td>
<td>• Replace the protocol argument with pap, chap, or ms-chap.</td>
</tr>
<tr>
<td>protocol [protocol]</td>
<td>• 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.</td>
</tr>
<tr>
<td>[protocol]] [list-name</td>
<td>• 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.</td>
</tr>
<tr>
<td>default]</td>
<td></td>
</tr>
</tbody>
</table>
Where To Go 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 466.
- If you enabled CHAP on an interface, configure a CHAP authentication password, as described in the “Configuring a CHAP Authentication Password” section on page 468.
- 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 470.
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.

Prerequisites

You must enable PAP authentication on the interface with the `ppp authentication` command, as described in the “Enabling PAP, CHAP, and MS-CHAP Authentication” section on page 463.

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

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

Example:

RP/0/RP0/CPU0:router# configure
### Command or Action

<table>
<thead>
<tr>
<th>Step 2</th>
<th><code>interface type interface-path-id</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Enters interface configuration mode.</td>
</tr>
</tbody>
</table>

| 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` |
| Purpose | 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. |

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

<table>
<thead>
<tr>
<th>Step 5</th>
<th><code>show running-config</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show running-config</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.</td>
</tr>
</tbody>
</table>
Configuring a CHAP Authentication Password

This task explains how to enable CHAP authentication and configure a CHAP password on a serial or POS interface.

Prerequisites

You must enable CHAP authentication on the interface with the ` PPP authentication` command, as described in the “Enabling PAP, CHAP, and MS-CHAP Authentication” section on page 463.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
</tbody>
</table>


### Configuring PPP on Cisco IOS XR Software

#### How to Configure PPP

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><code>interface type interface-path-id</code></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt; RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>`ppp chap password [clear</td>
<td>encrypted] password`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt; RP/0/RP0/CPU0:router(config-if)# ppp chap password clear xxxx</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter <code>clear</code> to select cleartext encryption, or <code>encrypted</code> if the password is already encrypted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Replace the <code>password</code> argument with a cleartext or already-encrypted password. This password is used to authenticate secure communications among a collection of routers.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The <code>ppp chap password</code> command is used for remote CHAP authentication only (when routers authenticate to the peer) and does not affect 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).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The CHAP secret password is used by the routers in response to challenges from an unknown peer.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>end</code>&lt;br&gt;or&lt;br&gt;<code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt; RP/0/RP0/CPU0:router(config-if)# end or RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• When you issue the <code>end</code> command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:&lt;br&gt;– Entering <code>yes</code> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>show running-config</code></td>
<td>Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt; RP/0/RP0/CPU0:router# show running-config</td>
<td></td>
</tr>
</tbody>
</table>
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.

Prerequisites

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” section on page 463.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface type interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ppp ms-chap password [clear</td>
<td>encrypted] password</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# ppp ms-chap password clear xxxx</td>
<td>The MS-CHAP secret password is used by the routers in response to challenges from an unknown peer.</td>
</tr>
</tbody>
</table>
Disabling an Authentication Protocol

This section contains the following procedures:

- Disabling PAP Authentication on an Interface, page 471
- Disabling CHAP Authentication on an Interface, page 473
- Disabling MS-CHAP Authentication on an Interface, page 474

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
# How to Configure PPP

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface type interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</td>
</tr>
<tr>
<td><strong>Step 3</strong> ppp pap refuse</td>
<td>Refuses Password Authentication Protocol (PAP) authentication from peers requesting it.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ppp pap refuse</td>
</tr>
<tr>
<td></td>
<td>• If outbound Challenge Handshake Authentication Protocol (CHAP) has been configured (using the <strong>ppp authentication</strong> command), CHAP will be suggested as the authentication method in the refusal packet.</td>
</tr>
<tr>
<td></td>
<td>• PAP authentication is disabled by default.</td>
</tr>
<tr>
<td><strong>Step 4</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# end  or RP/0/RP0/CPU0:router(config-if)# commit</td>
</tr>
<tr>
<td></td>
<td>• When you issue the <strong>end</strong> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
<tr>
<td><strong>Step 5</strong> show running-config</td>
<td>Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# show running-config</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface type interface-path-id</code></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>ppp chap refuse</code></td>
<td>Refuses CHAP authentication from peers requesting it. After you enter the <code>ppp chap refuse</code> command under the specified interface, all attempts by the peer to force the user to authenticate with the help of CHAP are refused.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# ppp chap refuse</td>
<td>• CHAP authentication is disabled by default.</td>
</tr>
<tr>
<td></td>
<td>• If outbound Password Authentication Protocol (PAP) has been configured (using the <code>ppp authentication</code> command), PAP will be suggested as the authentication method in the refusal packet.</td>
</tr>
</tbody>
</table>
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`

---

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

**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.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><code>interface type interface-path-id</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Refuses MS-CHAP authentication from peers requesting it. After you enter the <code>ppp ms-chap refuse</code> command under the specified interface, all attempts by the peer to force the user to authenticate with the help of MS-CHAP are refused.</td>
</tr>
<tr>
<td><code>ppp ms-chap refuse</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# ppp ms-chap refuse</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# end or RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.</td>
</tr>
<tr>
<td><code>show running-config</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show running-config</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Multilink PPP

This section contains the following procedures:

- Prerequisites, page 476
- Restrictions, page 476
- Configuring the Controller, page 477
- Configuring the Interfaces, page 480
- Configuring MLPPP Optional Features, page 482
- Removing an MLPPP member, page 484

Prerequisites

- Before configuring MLPPP, be sure you have one of the SPAs listed here installed. Both MLPPP and LFI are supported on the following SIPs and SPAs:
  - Cisco 1-Port Channelized STM-1/OC-3 SPA
  - Cisco 1-Port Channelized OC-12/STM-4 SPA
  - Cisco 2-Port and 4-Port Channelized T3 SPAs (SPA-2XCT3/DS0, SPA-4XCT3/DS0)
  - Cisco 8-Port Channelized T1/E1 SPA
  - Cisco 1-Port Channelized OC-12 Line Card (Supports MLPPP only)

Restrictions

MLPPP for Cisco IOS XR software has the following restrictions:

- Only full rate T1s are supported.
- All links in a bundle must belong to the same SPA.
- All links in a bundle must operate at the same speed.
- A maximum of 12 links per bundle is supported.
- A maximum of 28 bundles is supported on the 2-Port Channelized T3 SPA.
- A maximum of 56 bundles is supported on the 4-Port Channelized T3 SPA.
- A maximum of 224 bundles is supported per line card.
- All serial links in an MLPPP bundle inherit the value of the `mtu` command from the multilink interface. Therefore, you should not configure the `mtu` command on a serial interface before configuring it as a member of an MLPPP bundle. The Cisco IOS XR software blocks the following:
  - Attempts to configure a serial interface as a member of an MLPPP bundle if the interface is configured with a nondefault MTU value.
  - Attempts to change the `mtu` command value for a serial interface that is configured as a member of an MLPPP bundle.

In Cisco IOS XR software, multilink processing is controlled by a hardware module called the Multilink Controller, which consists of an ASIC, network processor, and CPU working in conjunction. The MgmtMultilink Controller makes the multilink interfaces behave like the serial interfaces of channelized SPAs.
Configuring the Controller

Perform this task to configure the controller.

SUMMARY STEPS

1. configure
2. controller type interface-path-id
3. mode type
4. clock source {internal | line}
5. exit
6. controller t1 interface-path-id
7. channel-group channel-group-number
8. timeslots range
9. exit
10. exit
11. controller mgmtmultilink interface-path-id
12. bundle bundle-id
13. end
   or
   commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller type interface-path-id</td>
<td>Enters controller configuration submode and specifies the controller name and instance identifier in <code>rack/slot/module/port</code> notation.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>controller t3 0/1/0/0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> mode type</td>
<td>Configures the type of multilinks to channelize; for example, 28 T1s.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>mode t1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> clock source {internal</td>
<td>line}</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td><code>clock source internal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits controller configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> controller t1 interface-path-id</td>
<td>Enters T1 configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>controller t1 0/1/0/0/1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> channel-group channel-group-number</td>
<td>Creates a T1 channel group and enters channel group configuration mode for that channel group. Channel group numbers can range from 0 to 23.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>channel-group 0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> timeslots range</td>
<td>Associates one or more DS0 time slots to a channel group and creates an associated serial subinterface on that channel group.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td><code>timeslots 1-24</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> timeslots range</td>
<td>The time slot range must be from 1 to 24 for the resulting serial interface to be accepted into a MLPPP bundle.</td>
</tr>
</tbody>
</table>

**Note** The default clock source is `internal`.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>exit</td>
<td>Exits channel group configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-t1-channel_group)# exit</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>exit</td>
<td>Exits T1 configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-t1)# exit</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>controller mgmtmultilink interface-path-id</td>
<td>Enters controller configuration submode for the management of multilink interfaces. Specify the controller name and instance identifier in rack/slot/module/port notation.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# controller mgmtmultilink 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>bundle bundle-id</td>
<td>Creates a multilink interface with the specified bundle ID.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mgmtmultilink)# bundle 20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-t3)# end or commit</td>
<td></td>
</tr>
</tbody>
</table>

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

  Uncommitted changes found, commit them before exiting(yes/no/cancel)?
  [cancel]:

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

Perform this task to configure the interfaces.

Restrictions

- All serial links in an MLPPP bundle inherit the value of the `mtu` command from the multilink interface. Therefore, you should not configure the `mtu` command on a serial interface before configuring it as a member of an MLPPP bundle. The Cisco IOS XR software blocks the following:
  - Attempts to configure a serial interface as a member of an MLPPP bundle if the interface is configured with a nondefault MTU value.
  - Attempts to change the `mtu` command value for a serial interface that is configured as a member of an MLPPP bundle.

SUMMARY STEPS

1. `configure`
2. `interface multilink interface-path-id`
3. `ipv4 address address/mask`
4. `multilink fragment-size bytes
   keepalive [interval | disable] [retry]`
5. `exit`
6. `interface type interface-path-id`
7. `encapsulation type`
8. `multilink group group-id`
9. `end`
   or
   `commit`
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface multilink interface-path-id</td>
<td>Specifies the multilink interface name and instance identifier in rack/slot/module/port/bundle-id notation, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# interface multilink 0/1/0/0/20</td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv4 address ip-address</td>
<td>Assigns an IP address and subnet mask to the interface in the format: A.B.C.D/prefix or A.B.C.D/mask</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 address 80.170.0.1/24</td>
</tr>
<tr>
<td><strong>Step 4</strong> multilink fragment-size bytes</td>
<td>(Optional) Specifies the size of the multilink fragments, such as 128 bytes. Some fragment sizes may not be supported. The default is no fragments.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# multilink fragment-size 350</td>
</tr>
<tr>
<td><strong>Step 5</strong> keepalive {interval</td>
<td>disable}[retry]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# keepalive disable</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>To connect with some Cisco IOS devices, multilink keepalives need to be disabled on both devices.</td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 7</strong> interface type interface-path-id</td>
<td>Specifies the interface name and instance identifier in rack/slot/module/port/1-number:channel-group notation, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# interface serial 0/1/0/0/1:0</td>
</tr>
</tbody>
</table>
How to Configure PPP

Perform this task to configure either of the following optional features:

- Minimum number of active links
- Multilink interleave

**Note** Minimum number active links must be configured at both endpoints.

**SUMMARY STEPS**

1. `configure`
2. `interface multilink interface-path-id`
3. `multilink`
4. `ppp multilink minimum-active links value`
5. `multilink interleave`
6. no shutdown
7. end
   or
commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface multilink interface-path-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# interface multilink 0/1/0/0/1</td>
</tr>
<tr>
<td>Specifies the multilink interface name and instance identifier in rack/slot/module/port/bundle-id notation, and enters interface configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>multilink</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# multilink</td>
</tr>
<tr>
<td>Enters interface multilink configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>ppp multilink minimum-active links value</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 12</td>
</tr>
<tr>
<td>(Optional) Specifies the minimum number of active links for the multilink interface.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>multilink interleave</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if-multilink)# multilink interleave</td>
</tr>
<tr>
<td>(Optional) Enables interleave on a multilink interface.</td>
<td></td>
</tr>
</tbody>
</table>
## How to Configure PPP

### Removing an MLPPP member

Perform this task to remove an MLPPP member link.

**SUMMARY STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>no shutdown</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config-if-multilink)# no shutdown</strong></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>end</strong> or <strong>commit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config-t3)# end</strong> or <strong>RP/0/RP0/CPU0:router(config-t3)# commit</strong></td>
</tr>
</tbody>
</table>

- **Step 6**
  - 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 7**
  - 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.

---

**Command or Action**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>controller type interface-path-id</td>
<td></td>
</tr>
<tr>
<td>shutdown</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>interface type interface-path-id</td>
<td></td>
</tr>
<tr>
<td>no multilink group group-id</td>
<td></td>
</tr>
<tr>
<td>encapsulation type</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
## Configuring PPP on Cisco IOS XR Software

### How to Configure PPP

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
<td>Enters controller configuration submode and specifies the controller name and instance identifier in rack/slot/module/port notation.</td>
</tr>
<tr>
<td>2</td>
<td>controller type interface-path-id</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# controller t1 0/4/2/0/11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>shutdown</td>
<td>Exits T1 controller configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-t1)#shutdown</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>exit</td>
<td>Exits T1 configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-t1)#exit</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)#interface serial 0/4/3/11:0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>no multilink group group-id</td>
<td>Removes the multilink group for this interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)#no multilink group 111</td>
<td></td>
</tr>
</tbody>
</table>
This section provides the following configuration examples:

- Configuring a POS Interface with PPP Encapsulation: Example, page 486
- Configuring a Serial Interface with PPP Encapsulation: Example, page 487
- Configuring MLPPP: Example, page 487
- Verifying Multilink PPP Configurations, page 488

### Configuring a POS Interface with PPP Encapsulation: Example

The following example shows how to create and configure a POS interface with PPP encapsulation:

```plaintext
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 chap password encrypted xxxx
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

### Command or Action Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> encapsulation type</td>
<td>Specifies the type of encapsulation; in this case, PPP.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# no encapsulation ppp</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>
| **Step 8** end or 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]:
| **Example:** RP/0/RP0/CPU0:router(config-if)# end or RP/0/0/CPU0:router(config-if)# 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. |
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/O/RPO/CPU0:router# configure
RP/O/RPO/CPU0:router(config)# interface POS 0/3/0/1
RP/O/RPO/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/O/RPO/CPU0:router# configure
RP/O/RPO/CPU0:router(config)# interface serial 0/3/0/0:0
RP/O/RPO/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/O/RPO/CPU0:router(config-if)# encapsulation ppp
RP/O/RPO/CPU0:router(config-if)# no shutdown
RP/O/RPO/CPU0:router(config-if)# ppp authentication ms-chap MIS-access
RP/O/RPO/CPU0:router(config-if)# ppp ms-chap password encrypted xxxx
RP/O/RPO/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

Configuring MLPPP: Example

```
RP/O/RPO/CPU0:router# configure
RP/O/RPO/CPU0:router(config)# controller t3 0/1/0/0/1
RP/O/RPO/CPU0:router(config)# mode t1
RP/O/RPO/CPU0:router(config-t3)# clock source internal
RP/O/RPO/CPU0:router(config-t3)# exit
RP/O/RPO/CPU0:router(config)# controller t1 0/1/0/0/1/1
RP/O/RPO/CPU0:router(config-t1)# channel-group 0
RP/O/RPO/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/O/RPO/CPU0:router(config-t1-channel_group)# exit
RP/O/RPO/CPU0:router(config-t1)# exit
RP/O/RPO/CPU0:router(config)# controller mgmtmultilink 0/1/0/0
RP/O/RPO/CPU0:router(config-mgmtmultilink)# bundle 20
RP/O/RPO/CPU0:router(config-t3)# commit
RP/O/RPO/CPU0:router(config-t3)# exit

RP/O/RPO/CPU0:router(config)# interface multilink 0/1/0/0/20
RP/O/RPO/CPU0:router(config-if)# ipv4 address 80.170.0.1/24
RP/O/RPO/CPU0:router(config-if)# multilink fragment-size 128
RP/O/RPO/CPU0:router(config-if)# keepalive disable
RP/O/RPO/CPU0:router(config-if)# exit
RP/O/RPO/CPU0:router(config)# interface serial 0/1/0/0/1
RP/O/RPO/CPU0:router(config-if)# encapsulation ppp
RP/O/RPO/CPU0:router(config-if)# multilink group 20
RP/O/RPO/CPU0:router(config-t3)# commit
RP/O/RPO/CPU0:router(config-t3)# exit

RP/O/RPO/CPU0:router(config)# interface multilink 0/1/0/0/1
RP/O/RPO/CPU0:router(config-if)# multilink
RP/O/RPO/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 10
RP/O/RPO/CPU0:router(config-if-multilink)# multilink interleave
RP/O/RPO/CPU0:router(config-if-multilink)# no shutdown
RP/O/RPO/CPU0:router(config-t3)# commit
```
Verifying Multilink PPP Configurations

Use the following show commands to verify and troubleshoot your multilink configurations:

- `show multilink interfaces`: Examples, page 488
- `show ppp interfaces multilink`: Example, page 490
- `show ppp interface serial`: Example, page 491
- `show imds interface multilink`: Example, page 491

**show multilink interfaces: Examples**

```
RP/0/RP0/CPU0:Router# show multilink interfaces Serial 0/4/3/1/10:0
Serial0/4/3/1/10:0 is up, line protocol is up
  Encapsulation: PPP
  Multilink group id: 6
  Member status: ACTIVE

RP/0/RP0/CPU0:Router# show multilink interfaces Multilink 0/4/3/0/3
Multilink0/4/3/0/3 is up, line protocol is up
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: PPP
  Member Links: 1 active, 1 inactive
    - Serial0/4/3/1/5:0 is up, line protocol is up
      Encapsulation: PPP
      Multilink group id: 3
      Member status: ACTIVE
    - Serial0/4/3/1/6:0 is administratively down, line protocol is administratively down
      Encapsulation: PPP
      Multilink group id: 3
      Member status: INACTIVE : LCP has not been negotiated

  Fragmentation Statistics
  Input Fragmented packets 0            Input Fragmented bytes 0
  Output Fragmented packets 0           Output Fragmented bytes 0
  Input Unfragmented packets 0          Input Unfragmented bytes 0
  Output Unfragmented packets 0         Output Unfragmented bytes 0
  Input Reassembled packets 0           Input Reassembled bytes 0

RP/0/5/CPU0:Mav-IOX-Rahul#sho multilink interfaces Serial 0/4/3/1/10:0
Serial0/4/3/1/10:0 is up, line protocol is up
  Encapsulation: PPP
  Multilink group id: 6
  Member status: ACTIVE

RP/0/RP0/CPU0:Router# show multilink interfaces
Multilink0/4/3/0/1 is up, line protocol is up
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: FR
  Member Links: 1 active, 1 inactive
    - Serial0/4/3/1/2:0: INACTIVE : Down (Member link idle)
    - Serial0/4/3/1/1:0: ACTIVE : Up
```
Multilink0/4/3/0/10 is up, line protocol is down
Fragmentation: disabled
Interleave: disabled
Encapsulation: PPP
Member Links: 0 active, 0 inactive
Fragmentation Statistics
Input Fragmented packets 0  Input Fragmented bytes 0
Output Fragmented packets 0  Output Fragmented bytes 0
Input Unfragmented packets 0  Input Unfragmented bytes 0
Output Unfragmented packets 0  Output Unfragmented bytes 0
Input Reassembled packets 0  Input Reassembled bytes 0

Multilink0/4/3/0/100 is administratively down, line protocol is administratively down
Fragmentation: disabled
Interleave: disabled
Encapsulation: PPP
Member Links: 0 active, 0 inactive
Fragmentation Statistics
Input Fragmented packets 0  Input Fragmented bytes 0
Output Fragmented packets 0  Output Fragmented bytes 0
Input Unfragmented packets 0  Input Unfragmented bytes 0
Output Unfragmented packets 0  Output Unfragmented bytes 0
Input Reassembled packets 0  Input Reassembled bytes 0

Multilink0/4/3/0/2 is up, line protocol is up
Fragmentation: disabled
Interleave: disabled
Encapsulation: FR
Member Links: 2 active, 0 inactive
- Serial0/4/3/1/4:0: ACTIVE : Up
- Serial0/4/3/1/3:0: ACTIVE : Up

Multilink0/4/3/0/3 is up, line protocol is up
Fragmentation: disabled
Interleave: disabled
Encapsulation: PPP
Member Links: 1 active, 1 inactive
- Serial0/4/3/1/5:0: ACTIVE
- Serial0/4/3/1/6:0: INACTIVE : LCP has not been negotiated
Fragmentation Statistics
Input Fragmented packets 0  Input Fragmented bytes 0
Output Fragmented packets 0  Output Fragmented bytes 0
Input Unfragmented packets 0  Input Unfragmented bytes 0
Output Unfragmented packets 0  Output Unfragmented bytes 0
Input Reassembled packets 0  Input Reassembled bytes 0

Multilink0/4/3/0/4 is up, line protocol is up
Fragmentation: disabled
Interleave: disabled
Encapsulation: PPP
Member Links: 2 active, 0 inactive
- Serial0/4/3/1/8:0: ACTIVE
- Serial0/4/3/1/7:0: ACTIVE
Fragmentation Statistics
Input Fragmented packets 0  Input Fragmented bytes 0
Output Fragmented packets 0  Output Fragmented bytes 0
Input Unfragmented packets 0  Input Unfragmented bytes 0
Output Unfragmented packets 0  Output Unfragmented bytes 0
Input Reassembled packets 0  Input Reassembled bytes 0

Multilink0/4/3/0/5 is up, line protocol is up
Fragmentation: disabled
Interleave: enabled
Encapsulation: PPP
Member Links: 1 active, 0 inactive
- Serial0/4/3/1/9:0: ACTIVE
Fragmentation Statistics
Input Fragmented packets 0          Input Fragmented bytes 0
Output Fragmented packets 0         Output Fragmented bytes 0
Input Unfragmented packets 0       Input Unfragmented bytes 0
Output Unfragmented packets 0      Output Unfragmented bytes 0
Input Reassembled packets 0        Input Reassembled bytes 0

Multilink0/4/3/0/6 is up, line protocol is up
Fragmentation: disabled
Interleave: enabled
Encapsulation: PPP
Member Links: 1 active, 0 inactive
- Serial0/4/3/1/10:0: ACTIVE
Fragmentation Statistics
Input Fragmented packets 0          Input Fragmented bytes 0
Output Fragmented packets 0         Output Fragmented bytes 0
Input Unfragmented packets 0       Input Unfragmented bytes 0
Output Unfragmented packets 0      Output Unfragmented bytes 0
Input Reassembled packets 0        Input Reassembled bytes 0

Multilink0/4/3/0/7 is up, line protocol is down
Fragmentation: disabled
Interleave: enabled
Encapsulation: PPP
Member Links: 0 active, 1 inactive
- Serial0/4/3/1/11:0: INACTIVE : LCP has not been negotiated
Fragmentation Statistics
Input Fragmented packets 0          Input Fragmented bytes 0
Output Fragmented packets 0         Output Fragmented bytes 0
Input Unfragmented packets 0       Input Unfragmented bytes 0
Output Unfragmented packets 0      Output Unfragmented bytes 0
Input Reassembled packets 0        Input Reassembled bytes 0

Multilink0/4/3/0/8 is up, line protocol is down
Fragmentation: disabled
Interleave: enabled
Encapsulation: PPP
Member Links: 0 active, 1 inactive
- Serial0/4/3/1/12:0: INACTIVE : LCP has not been negotiated
Fragmentation Statistics
Input Fragmented packets 0          Input Fragmented bytes 0
Output Fragmented packets 0         Output Fragmented bytes 0
Input Unfragmented packets 0       Input Unfragmented bytes 0
Output Unfragmented packets 0      Output Unfragmented bytes 0
Input Reassembled packets 0        Input Reassembled bytes 0

show ppp interfaces multilink: Example

RP/0/RP0/CPU0:Router# show ppp interfaces multilink 0/3/1/0/1

Multilink 0/3/1/0/1 is up, line protocol is up
LCP: Open
    Keepalives disabled
IPCP: Open
    Local IPv4 address: 1.1.1.2
    Peer IPv4 address: 1.1.1.1
Multilink
Member Links: 2 active, 1 inactive (min-active 1)
- Serial0/3/1/0/0:0: ACTIVE
- Serial0/3/1/1/0: ACTIVE
- Serial0/3/1/0/2:0: INACTIVE : LCP has not been negotiated

show ppp interface serial: Example

RP/0/RP0/CPU0:Router# show ppp interface Serial 0/3/1/0/0:0
Serial 0/3/1/0/0:0 is up, line protocol is up
  LCP: Open
    Keepalives disabled
    Local MRU: 1500 bytes
    Peer MRU: 1500 bytes
    Local Bundle MRU: 1596 bytes
    Peer Bundle MRU: 1500 bytes
    Local Endpoint Discriminator: 1b61950e3e9ce8172c8289df0000003900000001
    Peer Endpoint Discriminator: 7d046cd8390a4519087aefb90000003900000001
  Authentication
    Of Peer: <None>
    Of Us:   <None>
  Multilink
    Multilink group id: 1
    Member status: ACTIVE

show imds interface multilink: Example

RP/0/RP0/CPU0:Router# show imds interface Multilink 0/3/1/0/1
IMDS INTERFACE DATA (Node 0x0)
Multilink0_3_1_0_1 (0x04001200)
-----------------------
flags: 0x0001002f    type: 55 (IFT_MULTILINK)    encap: 52 (ppp)
state: 3 (up)     mtu: 1600    protocol count: 3
control parent: 0x04000800     data parent: 0x00000000
-----------------------
protocol      capsulation     state        mtu
--------------- -------------------- --------------- --------
  12 (ipv4) 26 (ipv4)            3 (up)          1500
  47 (ipcp)            3 (up)          1500
  16 (ppp_ctrl)
  53 (ppp_ctrl)            3 (up)          1500
  0 (Unknown)
  139 (c_shim)            3 (up)          1600
  52 (ppp)            3 (up)          1504
  56 (queue_fifo)            3 (up)          1600
  60 (txm_nopull)            3 (up)          1600
Additional References

The following sections provide references related to PPP encapsulation.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using Cisco IOS XR software</td>
<td>Cisco IOS XR Getting Started Guide</td>
</tr>
<tr>
<td>Cisco IOS XR AAA services configuration information</td>
<td>Cisco IOS XR System Security Configuration Guide and Cisco IOS XR System Security Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC-1661</td>
<td>The Point-to-Point Protocol (PPP)</td>
</tr>
<tr>
<td>RFC-1994</td>
<td>PPP Challenge Handshake Authentication Protocol (CHAP)</td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/support">http://www.cisco.com/support</a></td>
</tr>
</tbody>
</table>
This module describes the configuration and management of 802.1Q VLAN interfaces on the Cisco CRS Router.

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 on Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.2</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>• Support was added for VLAN commands on bundled Ethernet interfaces.</td>
</tr>
<tr>
<td></td>
<td>• Support was added for the dot1q native vlan command on Cisco CRS-1 Router shared port adapters (SPAs).</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>• The Layer 2 Virtual Private Network (L2VPN) feature was first supported on Ethernet interfaces on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td></td>
<td>• Support was added on for the 8-Port 1-Gigabit Ethernet SPA.</td>
</tr>
<tr>
<td>Release 5.1.1</td>
<td>• The dot1q vlan id command was replaced with encapsulation dot1q command for 802.1q VLAN sub-interfaces.</td>
</tr>
<tr>
<td></td>
<td>• The show vlan interface command was replaced with show ethernet tags command.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Configuring 802.1Q VLAN Interfaces, page 496
- Information About Configuring 802.1Q VLAN Interfaces, page 496
- How to Configure 802.1Q VLAN Interfaces, page 499
- Configuration Examples for VLAN Interfaces, page 509
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, page 496
- 802.1Q Tagged Frames, page 496
- CFM on 802.1Q VLAN Interfaces, page 497
- Subinterfaces, page 497
- Subinterface MTU, page 497
- Native VLAN, page 497
- VLAN Subinterfaces on Ethernet Bundles, page 498
- Layer 2 VPN on VLANs, page 498

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 Cisco IOS XR Software module:

- Ethernet CFM, page 119
- Configuring Ethernet CFM, page 152
- Ethernet CFM Service Configuration: Example, page 187
- Ethernet CFM Show Command: Examples, page 188

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** 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 on page 504.

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 it 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.
How to Configure 802.1Q VLAN Interfaces

This section contains the following procedures:

- Configuring 802.1Q VLAN Subinterfaces, page 499
- Configuring Native VLAN, page 502
- Configuring an Attachment Circuit on a VLAN, page 504
- Removing an 802.1Q VLAN Subinterface, page 507

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 of this module.

Note

The `dot1q vlan` command was replaced with the `encapsulation dot1q` command for 802.1Q VLAN subinterfaces from Cisco IOS XR Release 5.1.x onwards. The `show vlan interface` command was replaced with `show ethernet tags` command from Cisco IOS XR Release 5.1.x onwards.

SUMMARY STEPS

1. configure
2. interface { GigabitEthernet | TenGigE | Bundle-Ether } interface-path-id.subinterface
3. dot1q vlan vlan-id
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 ethernet tags [ { GigabitEthernet | TenGigE | Bundle-Ether } interface-path-id ] [location instance]
9. show vlan trunks [brief] [location instance] [ { GigabitEthernet | TenGigE | Bundle-Ether } interface-path-id ] [summary]
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**  
configure | Enters global configuration mode. |
| **Example:**  
RP/0/RP0/CPU0:router# configure | |
| **Step 2**  
interface {GigabitEthernet | TenGigE | Bundle-Ether | } interface-path-id.subinterface | 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 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. |
| **Example:**  
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10 | |
| **Step 3**  
encapsulation dot1q vlan-id | 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:  
  `encapsulation dot1q vlan-id`  
• To configure a Q-in-Q AC, use the following syntax:  
  `encapsulation dot1q vlan-id second-dot1q vlan-id` |
| **Example:**  
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 100 | |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4** ipv4 address ip-address mask | 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. |
| **Example:**  
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24 | |
| **Step 5** exit | (Optional) Exits the subinterface configuration mode.  
- The *exit* command is not explicitly required. |
| **Example:**  
RP/0/RP0/CPU0:router(config-subif)# exit | |
| **Step 6** Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces. | — |
| **Step 7** end or 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. |
| **Example:**  
RP/0/RP0/CPU0:router(config)# end  
or  
RP/0/RP0/CPU0:router(config)# commit | |
### 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`
5. `show ethernet trunk bundle-ether instance` (Optional)
6. `show vlan trunks [brief] [location instance] [{GigabitEthernet | TenGigE | Bundle-Ether | fastethernet} interface-path-id] [summary]`

---

#### Command or Action

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>show ethernet tags [type</code></td>
<td>(Optional) Displays the interface configuration.</td>
</tr>
<tr>
<td></td>
<td><code>interface-path-id] [location</code></td>
<td>• To display the configuration for a particular port, use the <code>location</code></td>
</tr>
<tr>
<td></td>
<td><code>instance]</code></td>
<td>keyword.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>• To display the configuration for the specified interface or subinterface, use the <code>interface</code> keyword.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show vlan interface 5</td>
<td>(Optional) Displays the interface configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• To display the configuration for a particular port, use the <code>location</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>keyword.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• To display the configuration for the specified interface or subinterface, use the <code>interface</code> keyword.</td>
</tr>
</tbody>
</table>

---

#### Command or Action

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>show vlan trunks [brief] [location instance]</code></td>
<td>(Optional) Displays summary information about each of the VLAN trunk interfaces.</td>
</tr>
<tr>
<td></td>
<td>[{GigabitEthernet</td>
<td>TenGigE</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>• brief—Displays a brief summary.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show vlan trunk summary</td>
<td>• summary—Displays a full summary.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• location—Displays information about the VLAN trunk interface on the given port.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• interface—Displays information about the specified interface or subinterface.</td>
</tr>
</tbody>
</table>
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
</tbody>
</table>
| Step 2 | `interface {GigabitEthernet | TenGigE | Bundle-Ether} interface-path-id` | 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. |
| **Example:** | RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4 | |
| Step 3 | `dot1q native vlan number` | 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). |
| **Example:** | RP/0/RP0/CPU0:router(config-if)# dot1q native vlan 1 | |
How to Configure 802.1Q VLAN Interfaces

Command or Action | Purpose
--- | ---
**Step 4** | Saves configuration changes.
end or 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 5** | (Optional) Displays summary information about each of the interface configuration VLAN trunk interfaces.
show ethernet trunk bundle-ether instance show vlan trunks [brief] [location instance] [[GigabitEthernet | TenGigE | Bundle-Ether] interface-path-id] [summary] | The Ethernet bundle instance range is from 1 through 65535.

• 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 an Attachment Circuit on a VLAN**

Use the following procedure to configure an attachment circuit on a VLAN.

**Note**

The **dot1q vlan** command was replaced with the **encapsulation dot1q** command for 802.1Q VLAN subinterfaces from Cisco IOS XR Release 5.1.x onwards.

**SUMMARY STEPS**

1. configure
2. `interface {GigabitEthernet | TenGigE | Bundle-Ether} interface-path-id.subinterface l2transport`
3. `encapsulation dot1q vlan-id`
4. `l2protocol {cdp | pvst | stp | vtp} {forward | tunnel} {experimental bits | drop}`
5. `end` or `commit`
6. `show interfaces {GigabitEthernet | TenGigE} interface-path-id`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> `interface {GigabitEthernet</td>
<td>TenGigE</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0.1 l2transport</td>
<td></td>
</tr>
</tbody>
</table>

- 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.

- Naming notation is `instance.subinterface`, and a period between arguments is required as part of the notation.

**Note** You must include the `l2transport` keyword in the command string; otherwise, the configuration creates a Layer 3 subinterface rather than an AC.
### How to Configure 802.1Q VLAN Interfaces

#### Step 3

**Command or Action**

```
encapsulation dot1q vlan-id
```

**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:
  
  ```
  encapsulation dot1q vlan-id
  ```

- To configure a Q-in-Q AC, use the following syntax:
  
  ```
  encapsulation dot1q vlan-id second-dot1q vlan-id
  ```

- To configure a Q-in-Any AC, use the following syntax:
  
  ```
  encapsulation dot1q vlan-id second-dot1q any
  ```

#### Note

#### Step 4

**Command or Action**

```
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.
Configuring 802.1Q VLAN Interfaces on Cisco IOS XR Software

How to Configure 802.1Q VLAN Interfaces

HC-507

Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco CRS Router

What to Do Next

• To configure a Point-to-Point pseudo-wire cross connect on the AC, see the Implementing MPLS Layer 2 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.

Note

The show vlan interface command was replaced with show ethernet tags command from Cisco IOS XR Release 5.1.x onwards.

SUMMARY STEPS

1. configure

Step 5

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if-l2)# end or RP/0/RP0/CPU0:router(config-if-l2)# commit</td>
<td></td>
</tr>
</tbody>
</table>

Step 6

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# show interfaces TenGigE 0/3/0/0.1</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure 802.1Q VLAN Interfaces

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 ethernet tags** [type interface-path-id] [location instance]

6. **show vlan trunks** [brief] [location instance] [[GigabitEthernet | TenGigE | Bundle-Ether |] interface-path-id] [summary]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>no interface</strong> {GigabitEthernet</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Removes the subinterface, which also automatically deletes all the configuration applied to the subinterface.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10</td>
<td>• Replace the <strong>interface-path-id</strong> argument with one of the following instances:</td>
</tr>
<tr>
<td></td>
<td>– 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.</td>
</tr>
<tr>
<td></td>
<td>– Ethernet bundle instance. Range is from 1 through 65535.</td>
</tr>
<tr>
<td></td>
<td>• Replace the <strong>subinterface</strong> argument with the subinterface value. Range is from 0 through 4095.</td>
</tr>
<tr>
<td></td>
<td>Naming notation is <strong>instance.subinterface</strong>, and a period between arguments is required as part of the notation.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Repeat Step 2 to remove other VLAN subinterfaces.</td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4**

```
end
or
commit
```

**Example:**

```
RP/0/RP0/CPU0:router(config)# end
or
RP/0/RP0/CPU0:router(config)# commit
```

**Purpose**

Saves configuration changes.

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

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
```

- 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 tags [GigabitEthernet | TenGigE | Bundle-Ether] interface-path-id | location instance]
```

**Example:**

```
RP/0/RP0/CPU0:router# show vlan trunk summary
```

(Optional) Displays the interface configuration.

- To display the configuration for a port, use the `location` keyword.
- To display the configuration for the specified interface or subinterface, use the `interface` keyword.

**Step 6**

```
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.

## Configuration Examples for VLAN Interfaces

This section contains the following example:

**VLAN Subinterfaces: Example, page 509**

### VLAN Subinterfaces: Example

The following example shows how to create three VLAN subinterfaces at one time:
Configuring 802.1Q VLAN Interfaces on Cisco IOS XR Software

Configuration Examples for VLAN Interfaces

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.1
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 10
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 10.0.10.1/24
RP/0/RP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.2
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 20
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 10.0.20.1/24
RP/0/RP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.3
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 30
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 10.0.30.1/24
RP/0/RP0/CPU0:router(config-subif)# commit
RP/0/RP0/CPU0:router(config)# exit
RP/0/RP0/CPU0:router# show vlan trunks summary
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 ethernet tags
interface     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 brief
interface        encapsulations         intf-state
Te0/2/0/4        802.1Q (Ether)         up
```

The following example shows how to create two VLAN subinterfaces on an Ethernet bundle:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface bundle-ether 2
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.2.1/24
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# interface bundle-ether 2.1
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 10
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 192.168.100.1/24
RP/0/RP0/CPU0:router(config-subif)# exit
RP/0/RP0/CPU0:router(config)# interface bundle-ether 2.2
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 20
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 192.168.200.1/24
RP/0/RP0/CPU0:router(config-subif)# exit
RP/0/RP0/CPU0:router(config)# commit

The following example shows how to create a basic dot1Q AC:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0.1
RP/0/RP0/CPU0:router(config-subif)# l2transport
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 20
RP/0/RP0/CPU0:router(config-subif)# commit
RP/0/RP0/CPU0:router(config-subif)# exit
RP/0/RP0/CPU0:router(config)# exit
```

The following example shows how to create a Q-in-Q AC:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0.2
RP/0/RP0/CPU0:router(config-subif)# l2transport
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 20 second-dot1q 10
```
The following example shows how to create a Q-in-Any AC:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.3
RP/0/RP0/CPU0:router(config-subif)# l2transport
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q vlan 30 second-dot1q any
RP/0/RP0/CPU0:router(config-subif)# commit
RP/0/RP0/CPU0:router(config-subif)# exit
RP/0/RP0/CPU0:router(config)# exit
```

### Additional References

These sections provide references related to VLAN interface configuration.

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td><em>Cisco IOS XR Master Commands List</em></td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference</em></td>
</tr>
<tr>
<td>Initial system bootup and configuration information for a router using the Cisco IOS XR Software</td>
<td><em>Cisco IOS XR Getting Started Guide</em></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td><em>Cisco IOS XR Interface and Hardware Component Command Reference</em></td>
</tr>
</tbody>
</table>
## Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature</td>
<td>—</td>
</tr>
</tbody>
</table>

## MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no applicable MIBs for this module.</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

## RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

## Technical Assistance

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

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**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced on the Cisco CRS Router.</td>
</tr>
</tbody>
</table>

**Contents**

- Prerequisites for Configuring Tunnel Interfaces, page 512
- Information About Configuring Tunnel Interfaces, page 512
- How to Configure Tunnel Interfaces, page 514
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, page 512
- Virtual Interface Naming Convention, page 512
- Tunnel-IPSec Overview, page 513
- Tunnel-IPSec Naming Convention, page 513
- Crypto Profile Sets, page 513
- How to Configure Tunnel Interfaces, page 514

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:

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
</table>


Information About Configuring Tunnel Interfaces

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 “Additional References” section on page 517. 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.
How to Configure Tunnel Interfaces

This section contains the following procedures:

- Configuring Tunnel-IPSec Interfaces, page 514 (Required)

Configuring Tunnel-IPSec Interfaces

This task explains how to configure Tunnel-IPSec interfaces.

Prerequisites

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 Cisco IOS XR System Security Configuration Guide.

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. `interface tunnel-ipsec identifier`
3. `profile profile-name`
4. `tunnel source {ip-address \ interface-id}`
5. `tunnel destination {ip-address \ tunnel-id}`
6. `end`
   or
   `commit`
7. `show ip route`
# Configuring Tunnel Interfaces on Cisco IOS XR Software

## How to Configure Tunnel Interfaces

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>interface tunnel-ipsec identifier</code></td>
<td>Identifies the IPSec interface to which the crypto profile will be attached and enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface tunnel-ipsec 30</td>
<td></td>
</tr>
</tbody>
</table>
| 3    | `profile profile-name` | 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. |
|      | **Example:** RP/0/RP0/CPU0:router(config-if)# profile user1 |
| 4    | `tunnel source (ip-address | interface-id)` | Specifies the tunnel source IP address or interface ID.  
   - This command is required for both static and dynamic profiles. |
|      | **Example:** RP/0/RP0/CPU0:router(config-if)# tunnel source Ethernet0/1/1/2 |
| 5    | `tunnel destination {ip-address | tunnel-id}` | (Optional) Specifies the tunnel destination IP address.  
   - This command is not required if the profile is dynamic. |
|      | **Example:** RP/0/RP0/CPU0:router(config-if)# tunnel destination 192.168.164.19 |
Configuration Examples for Tunnel Interfaces

This section contains the following example:

Tunnel-IPSec: Example, page 516

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(config-if)# configure
RP/0/RP0/CPU0:router(config)# crypto ipsec transform-set tset1
RP/0/RP0/CPU0:router(config-transform-set 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(config-if)# configure
RP/0/RP0/CPU0:router(config)# crypto ipsec profile user1
RP/0/RP0/CPU0:router(config-user1)# match sampleacl 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
```
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 Cisco IOS XR System Security Configuration Guide.

Additional References

These sections provide references related to tunnel interface configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Information about IPSec and crypto profiles</td>
<td>Cisco IOS XR System Security Configuration Guide</td>
</tr>
<tr>
<td>Information about MPLS Traffic Engineering, including configuring a tunnel interface for MPLS-TE</td>
<td>Cisco IOS XR Multiprotocol Label Switching Configuration Guide</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no applicable MIBs for this module</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature</td>
<td>—</td>
</tr>
</tbody>
</table>

Technical Assistance

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

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.5.2</td>
<td>This feature was introduced on the Cisco CRS-1 Router.</td>
</tr>
<tr>
<td></td>
<td>Support was added for the 1-Port 10-Gigabit Ethernet LAN/WAN-PHY</td>
</tr>
<tr>
<td></td>
<td>Shared Port Adapter (SPA-1X10GE-WL-V2).</td>
</tr>
<tr>
<td>Release 3.6.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.7.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.8.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>Support was added for user configuration of WAN-PHY Signal Failure (SF)</td>
</tr>
<tr>
<td></td>
<td>and Signal Degrade (SD) Bit Error Rate (BER) reporting and thresholds.</td>
</tr>
<tr>
<td>Release 4.0.0</td>
<td>Support was added for the following physical layer interface modules (PLIMs):</td>
</tr>
<tr>
<td></td>
<td>• 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (14X10GBE-WL-XFP) (with the</td>
</tr>
<tr>
<td></td>
<td>Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)</td>
</tr>
<tr>
<td></td>
<td>• 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)</td>
</tr>
<tr>
<td>Release 4.0.1</td>
<td>Support was added for the following PLIMs:</td>
</tr>
<tr>
<td></td>
<td>• 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP)</td>
</tr>
<tr>
<td></td>
<td>• 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Configuring LAN/WAN-PHY Controller Interfaces, page 520
- Information About the LAN/WAN-PHY Controllers, page 520
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 is 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 at this URL:


- 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 physically 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.

- Configuring LAN-PHY Mode, page 522
- Configuring WAN-PHY Signal Failure and Signal Degrade Bit Error Rates, page 526

**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.

### Prerequisites

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]`
# Configuring LAN/WAN-PHY Controllers on Cisco IOS XR Software

## How to Configure LAN/WAN-PHY Controllers

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> show controllers wanphy interface-path-id [alarms</td>
<td>all</td>
</tr>
<tr>
<td><strong>Example:</strong> 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</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> controller wanphy interface-path-id</td>
<td>Specifies the LAN/WAN-PHY controller name in the notation rack/slot/module/port and enters wanphy configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> lanmode on</td>
<td>Configures LAN-PHY mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-wanphy)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> commit</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-wanphy)# commit</td>
<td></td>
</tr>
</tbody>
</table>

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

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
How to Configure LAN/WAN-PHY Controllers

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> hw-module subslot interface-path-id reload</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# hw-module subslot 0/6/1 reload</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> show controllers wanphy interface-path-id [alarms</td>
<td>all</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# show controllers wanphy 0/6/1/0 all</code></td>
<td></td>
</tr>
<tr>
<td><code>Tue Jan  5 23:28:37.738 PST</code></td>
<td></td>
</tr>
<tr>
<td><code>Interface: wanphy0_6_1_0</code></td>
<td></td>
</tr>
<tr>
<td><code>Configuration Mode: LAN Mode</code></td>
<td></td>
</tr>
</tbody>
</table>

### 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:

```bash
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.

### Prerequisites

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`
5. `hw-module subslot interface-path-id reload`
6. `show controllers wanphy interface-path-id [alarms | all | registers]`
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>controller wanphy interface-path-id</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Specifies the LAN/WAN-PHY controller name in the notation rack/slot/module/port and enters wanphy configuration mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>wanmode on</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Configures WAN-PHY mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-wanphy)# wanmode on</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>end</strong> or <strong>commit</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-wanphy)# end or commit</td>
<td></td>
</tr>
<tr>
<td>• When you issue the <strong>end</strong> command, the system prompts you to commit changes:</td>
<td></td>
</tr>
<tr>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>

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. Table 1 lists the configured SF and SD threshold values and the corresponding Hold on Times.

### Table 1  SF and SD Configured Threshold Value Hold on Times

<table>
<thead>
<tr>
<th>Configured Threshold Value</th>
<th>Hold on Time (max) in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>10e-3</td>
<td>13</td>
</tr>
<tr>
<td>10e-4</td>
<td>13</td>
</tr>
<tr>
<td>10e-5</td>
<td>13</td>
</tr>
<tr>
<td>10e-6</td>
<td>13</td>
</tr>
<tr>
<td>10e-7</td>
<td>20</td>
</tr>
<tr>
<td>10e-8</td>
<td>110</td>
</tr>
<tr>
<td>10e-9</td>
<td>1010</td>
</tr>
</tbody>
</table>

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.

### Prerequisites

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`
### How to Configure LAN/WAN-PHY Controllers

8. **end**
   or
   **commit**

9. **show controllers wanphy interface-path-id [alarms | all | registers]**

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** show controllers wanphy interface-path-id [alarms | all | registers] | Displays the configuration mode of the LAN/WAN-PHY controller.  
**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 |
| **Step 2** configure | Enters global configuration mode.  
**Example:**  
RP/0/RP0/CPU0:router# configure |
| **Step 3** controller wanphy interface-path-id | Specifies the LAN/WAN-PHY controller name in the notation rack/slot/module/port and enters wanphy configuration mode.  
**Example:**  
RP/0/RP0/CPU0:router(config)# controller wanphy 0/6/1/0 |
| **Step 4** report sd-ber | Enables signal degrade (sd) bit error rate (ber) reporting.  
**Note** By default sd-ber reporting is disabled.  
**Example:**  
RP/0/RP0/CPU0:router(config-wanphy)# report sd-ber |
| **Step 5** report sf-ber disable | Disables signal fault (sf) bit error rate (ber) reporting.  
**Note** By default sf-ber reporting is enabled.  
**Example:**  
RP/0/RP0/CPU0:router(config-wanphy)# report sf-ber disable |
| **Step 6** threshold sd-ber range | 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).  
**Example:**  
RP/0/RP0/CPU0:router(config-wanphy)# threshold sd-ber 7 |
| **Step 7** threshold sf-ber range | 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).  
**Example:**  
RP/0/RP0/CPU0:router(config-wanphy)# threshold sf-ber 4 |
### Command or Action

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>or commit</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RP0/CPU0:router(config-dwdm)# end
or
RP/0/RP0/CPU0:router(config-dwdm)# 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.

### Command or Action

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show controllers wanphy interface-path-id [alarms</td>
<td>all</td>
</tr>
</tbody>
</table>

**Example:**

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,
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
  LOF = 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,
```

Additional References

The following sections provide references related to LAN/WAN-PHY controller configuration.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>
## Configuring LAN/WAN-PHY Controllers on Cisco IOS XR Software

### Related Topic

<table>
<thead>
<tr>
<th>Document Title</th>
<th>Initial system bootup and configuration information for a router using Cisco IOS XR software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR Getting Started Guide</td>
<td></td>
</tr>
<tr>
<td>Cisco IOS XR AAA services configuration information</td>
<td></td>
</tr>
<tr>
<td>Cisco IOS XR System Security Configuration Guide and Cisco IOS XR System Security Command Reference</td>
<td></td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.3ae</td>
<td>10 Gigabit Ethernet</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHER-WIS (defined in RFC3637)</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC3637</td>
<td>Definitions of Managed Objects for the Ethernet WAN Interface Sublayer</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/support">http://www.cisco.com/support</a></td>
</tr>
</tbody>
</table>
Configuring the Satellite Network Virtualization (nV) System on the Cisco CRS Router

This module describes the configuration of the Satellite Network Virtualization system on the Cisco CRS Router.

### Table 2 Feature History for Configuring Satellite nV System

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.3.1</td>
<td>• Support for Cisco CRS-3 Router with Cisco CRS-3 Modular Services Line Card as host was included.</td>
</tr>
<tr>
<td>Release 4.3.2</td>
<td>• Multi-chassis and back to back support on Cisco CRS-3 Router was included.</td>
</tr>
</tbody>
</table>

### Contents

- Prerequisites for Configuration, page 533
- Overview of Satellite nV Switching System, page 534
- Overview of Port Extender Model, page 536
- Implementing a Satellite nV System, page 540
- Upgrading and Managing Satellite nV Software, page 551
- Configuration Examples for Satellite nV System, page 557
- Additional References, page 559

### Prerequisites for Configuration

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 the Satellite nV system, you must have these hardware and software installed in your chassis:
• Host — Cisco CRS-3 Modular Services Line Card with fixed PLIM (14x10GE, 20x10GE, and MSC140). The Line card that hosts the Satellite nV device can be a Cisco CRS Multi Chassis and Back to Back system.
• Satellite — Cisco ASR9000v.
• Software — Cisco IOS XR Software Release 4.3.1 or later with hfr-asr9000v-nV-px.pie.

Overview of Satellite nV Switching System

The Satellite Network Virtualization (nV) service or the Satellite Switching System enables you to configure a topology in which one or more satellite switches complement one or more Cisco CRS-3 Routers, to collectively realize a single virtual switching system. In this system, the satellite switches act under the management control of the routers. The complete configuration and management of the satellite chassis and features is performed through the control plane and management plane of the Cisco CRS-3 Router, which is referred to as the host.

Interconnection between the Cisco CRS-3 Router and its satellites is through standard Ethernet interfaces or bundle ethernet interfaces coming from a single modular services line card. All bundle members must be connected to the same satellite device.

When the Satellite nV service was introduced in Cisco IOS XR Software Release 4.3.x, Cisco ASR 9000v was used as the satellite device. It had four 10 Gigabit ports that were used as Interchassis Links (ICL).

In general, the type of interface used on the host is decided on the basis of the satellite device used. See Figure 1.

![Figure 1 Satellite nV Switching System](image)

This type of architecture can be realized in a carrier Ethernet transport network, with the satellite switches used as either access switches, or pre-aggregation and aggregation switches. These switches feed into the Cisco CRS-3 Router where more advanced Layer 2 and Layer 3 services are provisioned.
You can also utilize this model in a Fiber To The Business (FTTB) network application, where business internet and VPN services are offered on a commercial basis. Further, it can also be used in other networks, such as wireless or Radio Access Network (RAN) backhaul aggregation networks.

Benefits of Satellite nV System

The Satellite nV system offers these benefits:

1. **Extended port scalability and density** - You can create a virtual line card with more than 100 physical Gigabit Ethernet ports per slot. There is a significant increase of Ethernet port density in the resulting logical Cisco CRS-3 Router. For example, a single 14-port or 20-port Ten Gigabit Ethernet line card on the Cisco CRS-3 Router could integrate multiple satellite switches with a maximum of 100 Gig Ethernet ports per line card. This is beneficial because the Cisco CRS-3 Router has a per-slot non blocking capacity of up to 400 Gbps (with appropriate RSPs) and there is no other way of physically fitting hundreds of gigabit ethernet ports/ SFPs on the face plate of a single Cisco CRS-3 line card. As a result, in order to utilize the full capacity of an Cisco CRS-3 line card, it is necessary to physically separate out the ethernet ports, while maintaining logical management control. This would appear as if all ports were physically on a single large line card of the Cisco CRS-3 Router.

2. **Reduced cost** - All the core-routing capabilities and application features of the Cisco IOS XR software are available on low cost access switches.

3. **Reduced operating expense** - You can seamlessly upgrade software images, and also manage the chassis and services from a common point. This includes a single logical router view, single point of applying CLI or XML interface for the entire system of switches, a single point of monitoring the entire system of switches and a single point of image management and software upgrades for the entire system.

4. **Enhanced feature consistency** - All the features on the regular GigE ports of Cisco CRS-3 Router are also available on the access ports of a satellite access switch in a functionally identical and consistent manner. The typical application of a satellite system would be in the access and aggregation layers of a network. By integrating the access switches along with the aggregation or core switch, you can ensure that there are no feature gaps between the access switch and the aggregation or core switch. All features, such as carrier ethernet features, QoS and OAM, function consistently, from access to core, because of this integrated approach.

5. **Improved feature velocity** - With the satellite solution, every feature that is implemented on the Cisco CRS-3 Router becomes instantly available at the same time in the access switch, resulting in an ideal feature velocity for the edge switch.

6. **Better resiliency** - The nV satellite solution enables better multi-chassis resiliency, as well as better end-to-end QoS. For more information on QoS capabilities, see *Cisco IOS XR Quality of Service Configuration Guide for the Cisco CRS Router*.
Overview of Port Extender Model

In the Port Extender Satellite switching system, a satellite switch is attached to its host through physical ethernet ports.

The parent Cisco CRS-3 Router is referred as the host in this model. From a management or a provisioning point of view, the physical access ports of the satellite switch are equivalent to the physical ethernet ports on the Cisco CRS-3 Router. You do not need a specific console connection for managing the Satellite Switching System, except for debugging purposes. The interface and chassis level features of the satellite are visible in the control plane of Cisco IOS XR Software running on the host. This allows the complete management of the satellites and the host as a single logical router.

In this model, a single Cisco CRS-3 Router hosts two satellite switches, SAT1 and SAT2, to form an overall virtual switching system; this is shown by the dotted line surrounding the Cisco CRS-3 Router, SAT1, and SAT2 in Figure 2.

This structure effectively appears as a single logical Cisco CRS-3 Router to the external network. External access switches A1, A2 connect to this overall virtual switch by physically connecting to SAT1 and SAT2 using normal ethernet links. The links between the satellite switches and the Cisco CRS-3 Router are ethernet links, and are referred as ICLs (Inter-Chassis Links). The Cisco CRS-3 Router is referred as the host in this system. When there is congestion on the interchassis links, an inbuilt QoS protection mechanism is available for the traffic.

Note

SAT1, SAT2, and the host Cisco CRS-3 Router need not be located in the same geographic location. This means that the ICLs need not be of nominal length for only intra-location or intra-building use. The ICLs may be tens, hundreds, or even thousands of miles in length, thereby creating a logical satellite switch spanning a large geography. This distance depends on the pluggables used on the CRS host and Satellite ICL port.
Features Supported in the Satellite nV System

This section provides details of the features of a satellite nV system.

Satellite System Physical Topology

The satellite system supports the point-to-point hub and spoke physical topology for the ICLs between satellite switches and the host Cisco CRS-3 Router. This topology allows a physical Ethernet MAC layer connection from the satellite to the Cisco CRS-3 Router. This can be realized using a direct Ethernet over Fiber or Ethernet over Optical transport (such as Ethernet over a SONET/ SDH/ CWDM/ DWDM network).

This topology also allows a satellite switch to be geographically at a separate location, other than that of the host Cisco CRS-3 Router.

Inter-Chassis Link Redundancy Modes and Load Balancing

The Satellite system supports these redundancy modes:

- **Non-redundant inter-chassis links mode** - In this mode, there is no link level redundancy between inter-chassis links of a satellite.

- **Redundant inter-chassis links mode** - In this mode, the link level redundancy between inter-chassis links are provided using a single link aggregation (LAG) bundle.

In the redundant ICL mode, the load balancing of traffic between members of the IC bundle is done using a simple hashing function based on the satellite access port ID, and not based on the flow based hash using L2 or L3 header contents from the packets. This ensures that a single ICL is used by all packets for a given satellite access port. As a result, the actions applied for QoS and other features consider all the packets belonging to a single satellite access port.

Note

Both the Access Bundles and ICL bundles can co-exist, but not concurrently.

For more details on QoS application and configuration on ICLs, see *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco CRS Router*.

Satellite Discovery and Control Protocols

A Cisco proprietary discovery and control protocol is used between the satellite switches and the host devices, to handle discovery, provisioning, and monitoring of the satellite devices from the host Cisco CRS-3 Satellite System in-band over the ICLs. The Satellite Discovery And Control (SDAC) Protocol provides the behavioural, semantic, and syntactic definition of the relationship between a satellite device and its host.

Satellite Discovery and Control Protocol IP Connectivity

The connectivity for the SDAC protocol is provided through a normal in-band IP routed path over the ICLs using private and public IP addresses appropriate for the carrier’s network.

You can configure a management IP address on the host CLI for each satellite switch and corresponding IP addresses on the ICLs. You can select addresses from the private IPv4 address space (for example, 10.0.0.0/8 or 192.1.168.0/24) in order to prevent any conflict with normal service level IPv4 addresses.
being used in the IPv4 FIB. You can also configure a private VRF that is used for only satellite management traffic, so that the IP addresses assigned to the satellites can be within this private VRF. This reduces the risk of address conflict or IP address management complexity compared to other IP addresses and VRFs that are used on the router.

**Quality of Service**

Most Layer-2, Layer-3 QoS and ACL features are supported on Satellite Ethernet interfaces that are similar to normal physical Ethernet interfaces, with the exception of any ingress policy with a queueing action. However, for QoS, there may be some functional differences in the behavior because in the Cisco IOS XR Software Release 4.3.1, user-configured MQC policies are applied on the Cisco CRS-3 Router, and not on the satellite switch interfaces. For more detailed information on QoS policy attributes, features, and their configuration, see *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco CRS Router*.

---

**Note**

User-configured QoS policies are independent of any default port level QoS that are applied in order to handle IC link congestion and oversubscription scenarios. In addition to the default port-level QoS applied on the satellite system ports, there is also some default QoS applied on the host side, to the ingress and egress traffic from and to the Satellite Ethernet ports.

---

**Time of Day Synchronization**

The Time of Day parameter on the satellite switch is synchronized with the time of day on the host. This ensures that time stamps on debug messages and other satellite event logs are consistent with the host, and with all satellite switches across the network. This is achieved through the SDAC Discovery Protocol from the host to the satellite switch when the ICLs are discovered.

---

**Satellite Chassis Management**

The chassis level management of the satellite is done through the host because the satellite switch is a logical portion of the overall virtual switch. This ensures that service providers get to manage a single logical device with respect to all aspects including service-level, as well as box-level management. This simplifies the network operations. These operations include inventory management, environmental sensor monitoring, and fault/alarm monitoring for the satellite chassis through the corresponding CLI, SNMP, and XML interfaces of the host.

---

**Note**

The Satellite nV System hardware features, support for SFPs, and compatible topologies are described in the *Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide*. 
Restrictions of the Satellite nV System

These are some of the software restrictions of the Cisco CRS-3 Satellite System:

- The inter-chassis link redundancy is supported only through the static EtherChannel, and not through LACP based link bundles. Minimum and maximum link commands are not applicable when ICL is a bundle.
- If a satellite system is operating in redundant ICL mode, then you cannot configure link bundles of any form (with or without LACP) on the access ports of that same satellite switch.
- If a satellite system is operating in redundant ICL mode, then Ethernet OAM features are not supported on the access ports of that satellite.
- These features, protocols, and topologies are not supported on the Cisco CRS-3 Satellite System:
  - L2VPN
  - QinQ
  - TE tunnel over Satellite interface
  - Pseudowire Headend
  - GRE over satellite interface
  - L2TPv3
  - Multicast over satellite interface
  - Satellite interface as core facing interface
  - 801.1ad/1ah encapsulation
  - HSRP and VRRP
  - HW DBA based netflow
- If a satellite system is operating in redundant ICL mode, then Cisco Discovery Protocol(CDP) and Link Layer Discovery Protocol(LLDP) are not supported on the access ports of that satellite.
- You cannot connect the same satellite box to more than one Cisco CRS-3 Modular Services Line Card.
- Both the access link bundles and ICL bundles can co-exist, but not concurrently.
- BFD echo mode is not supported on Satellite Gigabit Ethernet links. BFD Asynchronous mode is supported on the Satellite Gigabit Ethernet links that are not part of a bundle. When Satellite links are part of the Access bundle, only BFD over Logical Bundles (BLB) is supported.
- Adding non-ICL links (normal TenGigE links) to ICL bundle is not supported. This configuration is not rejected, but the behavior is unpredictable.
- Bundling of satellite and non-satellite interfaces is not supported.
- Bundling of satellite interfaces from different satellite boxes is not supported.
- Only Cisco CRS-3 Modular Services Line Card with fixed PLIM (14x10GE and 20x10GE) can be used to interconnect with the Satellite boxes.
- All bundle members must be from same satellite. The maximum number of bundle members is restricted to 44.
- ISSU and MDR are not supported on the satellite.
Implementing a Satellite nV System

The Interface Control Plane Extender (ICPE) infrastructure has a mechanism to provide the Control Plane of an interface physically located on the Satellite device in the local Cisco IOS XR software. After this infrastructure is established, the interfaces behave like other physical ethernet interfaces on the router.

The ICPE configuration covers these functional areas, which are each required to set up full connectivity with a Satellite device:

- Defining the Satellite nV System, page 540
- Configuring the Host IP Address, page 543
- Configuring the Inter-Chassis Links and IP Connectivity, page 544
- Plug and Play Satellite nV Switch Turn up: (Rack, Plug, and Go installation), page 549

Defining the Satellite nV System

Each satellite that is to be attached to Cisco IOS XR software must be configured on the host, and also be provided with a unique identifier. In order to provide suitable verification of configuration and functionality, the satellite type, and its capabilities must also be specified.

Further, in order to provide connectivity with the satellite, an IP address must be configured, which will be pushed down to the satellite through the Discovery protocol, and allows Control protocol connectivity.

This task explains how to define the satellite system by assigning an ID and basic identification information.

SUMMARY STEPS

1. configure
2. nv
3. satellite Satellite ID
4. serial-number string (Optional)
5. description string (Optional)
6. type type
7. ipv4 address address
8. end
   or
   commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>nv</code></td>
<td>Enters the nV configuration submode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>nv</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>satellite Sat ID</code></td>
<td>Declares a new satellite that is to be attached to the host and enters the satellite configuration submode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>Satellite IDs ranges from 100 to 239.</td>
</tr>
<tr>
<td></td>
<td><code>satellite &lt;100-239&gt;</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>serial-number string</code></td>
<td>(Optional) Serial number is used for satellite authentication.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>serial-number CAT1521B1BB</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>description string</code></td>
<td>(Optional) Specifies any description string that is associated with a satellite such as location and so on.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>description Milpitas Building12</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>type Type_name</code></td>
<td>Defines the expected type of the attached satellite.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>type asr9000v</code></td>
<td></td>
</tr>
</tbody>
</table>
### Implementing a Satellite nV System

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 7**

_{ipv4 address address_}

**Example:**
RP/0/RP0/CPU0:router(config-nV)# ipv4 address 10.22.1.2

Specifies the IP address to assign to the satellite. ICPE sets up a connected route to the specified IP address through all configured ICLs.

| **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.
  - 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 Host IP Address

This procedure gives you the steps to configure a host IP address on a loopback interface.

SUMMARY STEPS

1. `configure`
2. `interface Loopback0`
3. `ipv4 address 8.8.8.8 255.255.255.255`
4. `end`
   or
   `commit`

DETAILED STEPS

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

Example:

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

Step 2 `interface Loopback0` | Specifies the loopback address for the interface.

Example:

```
RP/0/RP0/CPU0:router(config)# interface loopback0
```
Configuring the Inter-Chassis Links and IP Connectivity

Inter-Chassis Links (ICLs) need to be explicitly configured, in order to indicate which satellite is expected to be connected. You must also specify the access port, that is down-stream GigE ports, which crosslink up to the Host through the configured ICL.

In order to establish connectivity between the host and satellite, suitable IP addresses must be configured on both sides. The satellite IP address is forwarded through the Discovery protocol. The configuration is described in the section, Defining the Satellite nV System, page 540.

---

**Note**

This configuration shows the use of the global default VRF. The recommended option is to use a private VRF for nV IP addresses as shown in the Satellite Management using private VRF subsection under Satellite System Configuration: Example.

This procedure shows the configuration of ICL in non-redundant mode.

**SUMMARY STEPS**

1. `configure`
2. `interface interface_name`
3. `description`
4. **ipv4 point-to-point** (optional)
5. **ipv4 unnumbered Loopback0** (optional)
6. **nv**
7. **satellite-fabric-link satellite id**
8. **remote-ports interface-type**
9. **end**

**or**

**commit**

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface interface-name</td>
<td>The supported inter-chassis link interface types are limited by the connectivity provided on the supported satellites. GigabitEthernet, TenGigE, and Bundle-Ether interfaces are the only supported ICL types.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface TenGigE0/2/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> description</td>
<td>Specifies the description of the supported inter-chassis link interface type.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-interface)# description To Sat5 1/46</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv4 point-to-point</td>
<td>(Optional) Configures the IPv4 point to point address.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-interface)# ipv4 point-to-point</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ipv4 unnumbered loopback0</td>
<td>(Optional) Configures the IPv4 loopback address on the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-interface)# interface unnumbered loopback0</td>
<td></td>
</tr>
</tbody>
</table>
## Configuring Inter-Chassis Links and IP Connectivity in Redundant ICL mode

This procedure describes the configuration of ICL in redundant mode.

### SUMMARY STEPS

1. `configure`
2. `interface bundle-ether id`
3. `description`
4. `nv`
5. `satellite-fabric-link satellite id`
6. `remote-ports interface-type`

### Command or Action | Purpose
--- | ---
**Step 6** nv | Enters the Network Virtualization configuration mode.  
**Example:**  
RP/0/RP0/CPU0:router(config-if)# nv

**Step 7** satellite-fabric-link satellite <id> | Specifies that the interface is an ICPE inter-chassis link.  
**Example:**  
RP/0/RP0/CPU0:router(config-int-nv)# satellite-fabric-link satellite 200

**Step 8** remote-ports interface-type | Configures the remote satellite ports 0 to 43.  
**Example:**  
RP/0/RP0/CPU0:router(config-int-nv)# remote-ports GigabitEthernet 0/0/0-43

**Step 9** end  
**or**  
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.
7. `interface interface-type`
8. `bundle id id mode on`
9. `end` 
or `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface bundle-ether id</code></td>
<td>Specifies the supported inter-chassis link interface type.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>description</code></td>
<td>Specifies the description of the supported inter-chassis link interface type.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-interface)# description To Sat5 1/46</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>nv</code></td>
<td>Enters the Network Virtualization configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# nv</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>satellite-fabric-link satellite &lt;id&gt;</code></td>
<td>Specifies that the interface is an ICPE inter-chassis link.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-int-nv)# satellite-fabric-link satellite 200</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>remote-ports interface-type</code></td>
<td>Configures the remote satellite ports 0 to 43.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-int-nv)# remote-ports GigabitEthernet 0/0/0-43</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>interface interface_type</code></td>
<td>Configures the specified interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-interface)# interface TenGigE0/0/6</td>
<td></td>
</tr>
</tbody>
</table>
## Implementing a Satellite nV System

### Step 8

**Command or Action**

- `bundle id mode on`

**Example:**

```
RP/0/RP0/CPU0:router(config-interface)#
bundle id 100 mode on
```

**Purpose**

Specifies the bundle id and activates it.

### Step 9

**Command or Action**

- `end`
- `commit`

**Example:**

```
RP/0/RP0/CPU0:router(config)# end
or
RP/0/RP0/CPU0:router(config)# commit
```

**Purpose**

Saves configuration changes.

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

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

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

---

**Note**

For information on QoS configuration on ICLs, see *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco CRS Router*.

---

### Auto-IP

The Auto IP feature improves the plug-and-play set up of an nV satellite system. With the Auto IP feature, IP connectivity to the satellite is automatically provisioned. As a result:

- The nV Satellite Loopback interface is created on the host
- Loopback interface is given an IP address from a private satellite VRF
- Satellite fabric links are unnumbered to the loopback interface
- The IP address assigned to satellite is auto-generated from the satellite VRF

In the case of Auto IP, you do not need to provide IP address on the nv satellite global configuration and on the ICL. In the case of manual IP, you need to provide IP address on the nV satellite global configuration and on ICL.

The auto-IP feature assigns an IP address in the format 10.x.y.1 automatically, where:

- x is the top (most significant) 8 bits of the satellite ID
- y is the bottom 8 bits (the rest) of the satellite ID
Configuration Example for Auto IP

```
configure
nv satellite 150
type asr9000v

configure
interface TenGigabitEthernet 0/1/0/0
nv satellite-fabric-link satellite 150
remote-ports GigabitEthernet 0/0/0-5
```

- You do not have to assign IP for the satellite and 10.0.<satellite-id>.1 is assigned automatically.
- nV-Loopback0 is created automatically. nVSatellite VRF is created automatically, and assigned to nV-Loopback0. 10.0.0.1/32 is assigned to nV-Loopback0.
- nV-Loopback0 is referenced by the Ten Gigabit Ethernet interface automatically when it is made as an ICL.

There is no CLI to enable Auto IP. If you do not configure manual IP, this will be invoked.

**Note** You can also override the Auto IP feature by using the standard IP configuration.

Configuring the Satellite nV Access Interfaces

The access GigabitEthernet interfaces on the satellite are represented locally in Cisco IOS XR Software using interfaces named GigabitEthernet similar to other non-satellite GigabitEthernet interfaces. The only difference is that the rack ID used for a satellite access GigabitEthernet interface is the configured satellite ID for that satellite.

These interfaces support all features that are normally configurable on GigabitEthernet interfaces (when running over a physical ICL), or Bundle-Ether interfaces (when running over a virtual ICL).

Plug and Play Satellite nV Switch Turn up: (Rack, Plug, and Go installation)

1. Unpack the satellite rack, stack, and connect to the power cord.
2. Plug in the qualified optics of correct type into any one or more of the SFP+ slots and appropriate qualified optics into SFP+ or XFP slots on the host. Connect through the SMF/MMF fiber.

**Note** Connect the 10GigE fibers from the host to any of the 10G SFP+ ports on the satellite device in any order.

**Note** The Satellite nV service can use the Cisco CRS-3 Router as host. The Cisco ASR 9000v can be used as satellite device.
3. Configure the satellite nV system through CLI or XML on the host 10GigE ports. Configure the host for nV operations as described in the sections Defining the Satellite nV System, Configuring the Host IP Address, and Configuring the Inter-Chassis Links and IP Connectivity.

4. Power up the chassis of the satellite device.

Note

For power supply considerations of ASR 9000v, refer to the Appendix C, Cisco ASR 9000 and Cisco CRS Satellite Systems (ASR 9000v) of the Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide online.

5. You can check the status of the satellite chassis based on these chassis error LEDs on the front face plate.
   - If the Critical Error LED turns ON, then it indicates a serious hardware failure.
   - If the Major Error LED turns ON, then it indicates that the hardware is functioning well but unable to connect to the host.
   - If the Critical and Major LEDs are OFF, then the satellite device is up and running and connected to the host.
   - You can do satellite ethernet port packet loopback tests through the host, if needed, to check end to end data path.

Note

When the satellite software requires an upgrade, it notifies the host. You can do an inband software upgrade from the host, if needed. Use the show nv satellite status on the host to check the status of the satellite.
Upgrading and Managing Satellite nV Software

Satellite software images are bundled inside a PIE and the PIE name is dependent on the type of satellite, such as hfr-asr9000v-nV-px.pie within the Cisco CRS-3 Router package. The Cisco IOS XR software production SMU tool can be used to generate patches for the satellite image in the field to deliver bug fixes or minor enhancements without requiring a formal software upgrade.

This section provides the commands to manage the satellite nV Software.

Prerequisites

You must have installed the satellite installation procedure using the Plug and Play Satellite installation procedure. For more information, see Plug and Play Satellite nV Switch Turn up: (Rack, Plug, and Go installation).

Installing a Satellite

To download and activate the software image on the satellite, use the `install nv satellite satellite ID / all transfer/activate` commands. The `transfer` command downloads the image to the satellite. When the `transfer` command is followed by the `activate` command, the software is activated on the satellite.

Example

```
RP/0/RP0/CPU0: sat-host# install nv satellite 100 transfer
Install operation initiated successfully.
RP/0/RP0/CPU0: sat-host#RP/0/RP0/CPU0: May 3 20:12:46.732 : icpe_gco[1146]: %PKT_INFRA-ICPE_GCO-6-TRANSFER_DONE : Image transfer completed on Satellite 100

RP/0/RP0/CPU0: sat-host# install nv satellite 100 activate
Install operation initiated successfully.
```

Note

If the `activate` command is run directly, then the software image is transferred to the satellite and also activated.

Example

```
RP/0/RP0/CPU0: sat-host# install nv satellite 101 activate
Install operation initiated successfully.
RP/0/RP0/CPU0: sat-host#RP/0/RP0/CPU0: May 3 20:06:33.276 : icpe_gco[1146]: %PKT_INFRA-ICPE_GCO-6-TRANSFER_DONE : Image transfer completed on Satellite 101
RP/0/RP0/CPU0: May 3 20:06:33.449 : icpe_gco[1146]: %PKT_INFRA-ICPE_GCO-6-INSTALL_DONE : Image install completed on Satellite 101
RP/0/RP0/CPU0: May 3 20:06:33.510 : invmgr[254]: %PLATFORM-INV-6-OIROUT : OIR: Node 101 removed
```
For the satellite image upgrade to work, you must ensure that the management-plane CLI is not configured on the Cisco CRS-3 Router. If it is configured, then you need to add this exception for each of the 10GigE interfaces, which are the satellite ICLs.

You can include the exception using this CLI:

```
control-plane
management-plane
inband
!..
interface TenGigE0/0/0/5  <=== To enable TFTP on nV satellite ICL
allow TFTP
```

If you do not include this exception, then the image download and upgrade fails.

### Monitoring the Satellite Software

To perform a basic status check, use the `show nv satellite status brief` command.

```
RP/0/RP0/CPU0:router# show nv satellite status brief

Sat-ID  Type      IP Address    MAC address     State
------  --------  ------------  --------------  --------------------------------
100     asr9000v  101.102.103.105  dc7b.9426.1594  Connected (Stable)
200     asr9000v  101.102.103.106  0000.0000.0000  Halted; Conflict: no links configured
220     asr9000v  194.168.9.9    0000.0000.0000  Halted; Conflict: satellite has no type configured
```

Note To check if an upgrade is required on satellite, run the `show nv satellite status satellite satellite_id` command.

**Example**

```
RP/0/RP0/CPU0:router# show nv satellite status satellite 100

Satellite 100
----------
  State: Connected (Stable)
  Type: asr9000v
  Description: sat-test
  MAC address: dc7b.9426.47e4
  IPv4 address: 100.1.1.1
  Configured Serial Number: CAT1521B1BB
  Received Serial Number: CAT1521B1BB
  Remote version: Compatible (latest version)
  ROMMON: 125.0 (Latest)
  FPGA: 1.13 (Latest)
  IOS: 200.8 (Latest)
  Configured satellite fabric links:
    TenGigE0/2/0/6
    ----------
  State: Satellite Ready
```
Port range: GigabitEthernet0/0/0-9
TenGigE0/2/0/13
------------------
State: Satellite Ready
Port range: GigabitEthernet0/0/30-39
TenGigE0/2/0/9
------------------
State: Satellite Ready
Port range: GigabitEthernet0/0/10-19

Note
In this example output, Remote version, ROMMON, FPGA, and IOS must show the latest version. If it does not, an upgrade is required on the satellite. The version numbers displayed are the installed version on the ASR 90000v. If a version number is displayed, instead of latest key word in the above output, that would correspond to the ASR9000v image bundles in the satellite pie.

Monitoring the Satellite Protocol Status

• To check the status of the satellite discovery protocol, use the show nv satellite protocol discovery command.

RP/0/RP0/CPU0:router# show nv satellite protocol discovery brief

<table>
<thead>
<tr>
<th>Interface</th>
<th>Sat-ID</th>
<th>Status</th>
<th>Discovered links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Te0/1/0/0</td>
<td>100</td>
<td>Satellite Ready</td>
<td>Te0/1/0/0</td>
</tr>
<tr>
<td>Te0/1/0/1</td>
<td>100</td>
<td>Satellite Ready</td>
<td>Te0/1/0/1</td>
</tr>
</tbody>
</table>

(Or)

RP/0/RP0/CPU0:router# show nv satellite protocol discovery interface TenGigE 0/1/0/0

Satellite ID: 100
Status: Satellite Ready
Remote ports: GigabitEthernet0/0/0-15
Host IPv4 Address: 101.102.103.104
Satellite IPv4 Address: 101.102.103.105
Vendor: cisco, ASR9000v-DC-E
Remote ID: 2
Remote MAC address: dc7b.9426.15c2
Chassis MAC address: dc7b.9426.1594

• To check the status of the satellite control protocol status, use the show nv satellite protocol control command.

RP/0/RP0/CPU0:router# show nv satellite protocol control brief

<table>
<thead>
<tr>
<th>Sat-ID</th>
<th>IP Address</th>
<th>Protocol state</th>
<th>Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>101.102.103.105</td>
<td>Connected</td>
<td>Ctrl, If-Ext L1, If-Ext L2, X-link, Soft Reset, Inventory, EnvMon, Alarm</td>
<td></td>
</tr>
</tbody>
</table>

RP/0/RSP0/CPU0:shanghai# sh nv satellite protocol control

Satellite 100

--------
IP address: 101.102.103.105
Status: Connected
Channels:
  Control
Monitoring the Satellite Inventory

You can use the `show inventory chassis`, `show inventory fans`, `show environment temperatures` commands in the admin configuration mode to monitor the status of satellite inventory.

```
RP/0/RP0/CPU0:router(admin)# show inventory chassis
NAME: "module 0/RSP0/CPU0", DESCR: "ASR9K Fabric, Controller, 4G memory"
PID: A9K-RSP-4G, VID: V02, SN: FCI143781GJ
... NAME: "fantray SAT100/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V00, SN: CAT1507B228

NAME: "module SAT100/0/CPU0", DESCR: "ASR-9000v GE-SFP Line Card"
PID: ASR-9000v, VID: N/A, SN:
NAME: "module mau GigabitEthernet100/0/CPU0/8", DESCR: "CISCO-AVAGO"
PID: SFP-GE-S, VID: V01, SN: AGM1424F08N

NAME: "module mau TenGigE100/0/CPU0/3", DESCR: "CISCO-FINISAR"
PID: SFP-10G-SR, VID: V02, SN: FNS144502Y3

NAME: "power-module SAT100/PM0/SP", DESCR: "ASR-9000v Power Module"
PID: ASR-9000v, VID: N/A, SN:
NAME: "Satellite Chassis ASR-9000v ID 100", DESCR: "ASR9000v"
PID: ASR-9000v-AC-A, VID: V00, SN: CAT12345678

RP/0/RP0/CPU0:router(admin)# show inventory fans
NAME: "fantray SAT100/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V01, SN: CAT1531B4TC

NAME: "fantray SAT101/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V01, SN: CAT1542B0LJ

NAME: "fantray SAT102/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V01, SN: CAT1531B477

RP/0/RP0/CPU0: sat-host (admin)# show inventory | b GigabitEthernet100/
NAME: "module mau GigabitEthernet100/0/CPU0/0", DESCR: "CISCO-FINISAR 
PID: SFP-GE-S, VID: , SN: FNS11350LS5E  
NAME: "module mau GigabitEthernet100/0/CPU0/1", DESCR: "CISCO-FINISAR 
PID: SFP-GE-S, VID: V01, SN: FNS0934M290  
NAME: "module mau GigabitEthernet100/0/CPU0/2", DESCR: "CISCO-FINISAR 
PID: SFP-GE-S, VID: , SN: FNS12280LS9  

RP/0/RP0/CPU0:router(admin)# show environment temperatures

<table>
<thead>
<tr>
<th>R/S/I</th>
<th>Modules</th>
<th>Sensor</th>
<th>(deg C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/RSP0/*</td>
<td>host</td>
<td>Inlet0</td>
<td>33.1</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>46.9</td>
</tr>
<tr>
<td>0/RSP1/*</td>
<td>host</td>
<td>Inlet0</td>
<td>32.1</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>45.9</td>
</tr>
<tr>
<td>0/0/*</td>
<td>host</td>
<td>Inlet0</td>
<td>37.3</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>52.3</td>
</tr>
<tr>
<td>0/1/*</td>
<td>spa0</td>
<td>InletTemp</td>
<td>34.0</td>
</tr>
<tr>
<td></td>
<td>spa0</td>
<td>Hotspot</td>
<td>34.5</td>
</tr>
<tr>
<td></td>
<td>spa1</td>
<td>LocalTemp</td>
<td>38.0</td>
</tr>
<tr>
<td></td>
<td>spa1</td>
<td>Chan1Temp</td>
<td>36.0</td>
</tr>
<tr>
<td></td>
<td>spa1</td>
<td>Chan2Temp</td>
<td>39.0</td>
</tr>
<tr>
<td></td>
<td>spa1</td>
<td>Chan3Temp</td>
<td>39.0</td>
</tr>
<tr>
<td></td>
<td>spa1</td>
<td>Chan4Temp</td>
<td>48.0</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Inlet0</td>
<td>36.1</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>64.0</td>
</tr>
<tr>
<td>0/2/*</td>
<td>host</td>
<td>Inlet0</td>
<td>39.2</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>54.6</td>
</tr>
<tr>
<td>0/3/*</td>
<td>host</td>
<td>Inlet0</td>
<td>41.3</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>48.5</td>
</tr>
<tr>
<td>0/FT0/*</td>
<td>host</td>
<td>Inlet0</td>
<td>42.3</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>36.1</td>
</tr>
<tr>
<td>0/FT1/*</td>
<td>host</td>
<td>Inlet0</td>
<td>40.4</td>
</tr>
<tr>
<td></td>
<td>host</td>
<td>Hotspot0</td>
<td>35.8</td>
</tr>
<tr>
<td>SAT100/FT0/*</td>
<td>host</td>
<td>Hotspot0</td>
<td>53.0</td>
</tr>
<tr>
<td>SAT101/FT0/*</td>
<td>host</td>
<td>Hotspot0</td>
<td>56.0</td>
</tr>
<tr>
<td>SAT102/FT0/*</td>
<td>host</td>
<td>Hotspot0</td>
<td></td>
</tr>
</tbody>
</table>
Reloading the Satellite Device

In order to reload the satellite device, use the `hw-module satellite satellite id/all reload` command.

Example

```
RP/0/RP0/CPU0# hw-module satellite 101 reload
Reload operation completed successfully.
RP/0/RP0/CPU0: May 3 20:26:51.883 : invmgr[254]: %PLATFORM-INV-6-OIROUT : OIR: Node 101 removed
```

Port Level Parameters Configured on a Satellite

These are the port-level parameters that can be configured on a satellite nV system:

- Admin state (shut and no shut)
- Ethernet MTU
- Ethernet MAC Address.
- Ethernet link auto-negotiation that includes,
  - Half and full duplex
  - Link speed
  - Flow control
- Static configuration of auto-negotiation parameters such as speed, duplex, and flow control
- Carrier delay
- Layer-1 packet loopback which includes,
  - Line loopback
  - Internal loopback
- All satellite access port features on Cisco ASR 9000 Series Router.

Loopback Types on Satellite Ports

There are two types of loopback interfaces that can be configured on satellite ports. They are,

- Line Loopback
- Internal Loopback

These illustrations show how the loopback interface types function on a satellite.
You can specify the type of loopback to be used, as specified in this example:

```
Interface GigabitEthernet 100/0/0/0
loopback line | internal
```

### Configuration Examples for Satellite nV System

This section contains these examples:

- Satellite System Configuration: Example, page 557
  - Satellite Global Configuration, page 557
  - ICL (satellite-fabric-link) Interface Configuration, page 558
  - Satellite Interface Configuration, page 558
  - Satellite Management using private VRF, page 558

### Satellite System Configuration: Example

This example shows a sample configuration for setting up the connectivity of a Satellite System.

### Satellite Global Configuration

The satellite ID, type, serial number, description, and satellite IP address are configured in the satellite global configuration submode:

```
nv
satellite 100
  type asr9000v
  serial-number CAT1521B1BB
  description milpitas bldg20
  ipv4 address 10.0.0.100
  !
  !
```
ICL (satellite-fabric-link) Interface Configuration

On the interface connected to the satellite (TenGig or Bundle interface), the ports associated with the satellite ID must be specified. All fabric links connected to the same satellite must use the same (host) IPv4 address. The same or different host IPv4 addresses can be used for the same host to connect to different satellites.

interface Loopback1000
ipv4 address 10.0.0.1 255.0.0.0
interface TenGigE0/2/1/0
description To Sat5 1/46
ipv4 point-to-point
ipv4 unnumbered Loopback1000
nv
    satellite-fabric-link satellite 200
    remote-ports GigabitEthernet 0/0/0-30
!
!
!

These examples illustrate using IP addresses from the global VRF of the router for satellite management traffic. As discussed Satellite Discovery and Control Protocol IP Connectivity section, this can also be done using a private VRF, to prevent IP address conflict with the global VRF. In this case, the loopback interface and the ICL interfaces in the examples must be assigned to the private VRF dedicated for satellite management traffic.

Satellite Interface Configuration

The Satellite interface can be used as any other regular GigabitEthernet interfaces:

interface GigabitEthernet200/0/0/0
ip address 99.0.0.1 255.255.255.0
!
!
interface GigabitEthernet200/0/0/2
bundle id 100 mode active
!
!

Satellite Management using private VRF

You can use a special private VRF instead of the global default routing table, to configure the loopback interface and ICLs used for satellite management traffic. IP addresses in this VRF will not conflict with any other addresses used on the router.

router(config)# vrf NV_MGMT_VRF
router(config)# address ipv4 unicast

router(config)# interface Loopback 1000
router(config)# vrf NV_MGMT_VRF
router(config)# ipv4 address 10.0.0.1 / 24

router(config)# interface TenGigE 0/1/0/3
router(config)# vrf NV_MGMT_VRF
router(config)# ipv4 point-to-point
router(config)# ipv4 unnumbered Loopback 1000
router(config)# nv
router(config-nv)# satellite-fabric-link satellite 200
router(config-nv)# remote-ports GigabitEthernet 0/0/28-39
router(config)# nv satellite 200
router(config)# ipv4 address 10.0.0.2 / 24

Additional References

These sections provide references to related documents.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Satellite System software upgrade and downgrade on Cisco IOS XR Software</td>
<td>Cisco IOS XR Getting Started Guide for the Cisco CRS Router</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Satellite QoS configuration information for the Cisco IOS XR software</td>
<td>Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco CRS Router</td>
</tr>
<tr>
<td>Bidirectional Forwarding Detection features on the satellite system</td>
<td>Cisco  Routing Configuration Guide</td>
</tr>
<tr>
<td>Multicast features on the satellite system</td>
<td>Cisco  Multicast Configuration Guide</td>
</tr>
<tr>
<td>Broadband Network Gateway features on the satellite system</td>
<td>Cisco  Broadband Network Gateway Configuration Guide</td>
</tr>
<tr>
<td>AAA related information and configuration on the satellite system</td>
<td>Cisco  System Security Configuration Guide</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Configuring AAA Services on Cisco IOS XR Software module of Cisco IOS XR System Security Configuration Guide</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no applicable MIBs for this module.</td>
<td>To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>N.A</td>
</tr>
</tbody>
</table>

### Technical Assistance

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

HC Cisco IOS XR Interface and Hardware Component Configuration Guide
IC Cisco IOS XR IP Addresses and Services Configuration Guide
MCC Cisco IOS XR Multicast Configuration Guide
MNC Cisco IOS XR System Monitoring Configuration Guide
MPC Cisco IOS XR MPLS Configuration Guide
QC Cisco IOS XR Modular Quality of Service Configuration Guide
RC Cisco IOS XR Routing Configuration Guide
SBC Cisco IOS XR Session Border Controller Configuration Guide
SC Cisco IOS XR System Security Configuration Guide
SMC Cisco IOS XR System Management Configuration Guide
VFC Cisco IOS XR Virtual Firewall Configuration Guide

A

action capabilities-conflict command HC-118
action discovery-timeout command HC-118
aggregate none command HC-172
ais-shut command HC-321
aps group command HC-319
aps submode
channel command HC-319
interface loopback command HC-319
See aps group command
atm address-registration command HC-52
atm ilmi-config disable command HC-54
atm ilmi-keepalive command HC-52
ATM interface
address registration and callback functions HC-52
ATM overview HC-15
attaching a vc-class HC-48, HC-49, HC-51
bringing up HC-22, HC-59
configuring
OAM AIS/RDI HC-45
OAM F5 loopback HC-45
OAM management HC-45

optional parameters HC-24, HC-59
PVC HC-29
traffic shaping HC-46
VC class HC-44, HC-62
creating
interface HC-28
PVC HC-28
subinterface HC-27, HC-28

F4 OAM overview HC-16

ILMI
disabling HC-53, HC-54
enabling HC-51, HC-52
overview HC-17
prerequisites HC-14
vc-class, attaching HC-47
VC class-mapping overview HC-16
verifying
ILMI configuration HC-53, HC-55

B

bert pattern command HC-345
buckets archive command HC-172
bundle command HC-479
Bundle-Ether command HC-227
bundle id command HC-227, HC-231
bundle-POS HC-252

C
cablelength command HC-329, HC-335, HC-340
card type
changing to E3 mode HC-330
changing to T3 mode HC-330
channel command HC-319
channel-group command HC-342, HC-478
Channelized SONET, configuring HC-540, HC-543
CHAP
 defined HC-378
 enabling HC-463
 password, configuring HC-468
 ppp HC-428, HC-457
 refusing HC-473
class command HC-408, HC-409
class-int command HC-48
class-map command HC-407, HC-408
class map submode
 match mpls experimental topmost command HC-407
 match precedence command HC-408
class-vc command HC-51
clock source command HC-329
configure command HC-396
clock source command HC-395
controller mgmtmultilink command HC-479
controller sonet command HC-402
control packets HC-392
crc command HC-377, HC-432

default settings
 flow control, Management Ethernet HC-71
 mac-address (Management Ethernet) HC-71
 speed (Management Ethernet) HC-71
delay trigger command HC-314
dot1q native vlan command HC-502
dot1q vlan command HC-500, HC-504
duplex command HC-74
Dynamic Packet Transport (DPT), Cisco HC-391

E
E1 controller
 card type, setting HC-330
 default configuration values HC-329
E3 controller
 cable length, setting HC-329
 card type, setting HC-330
 clock source, default value HC-329
 configuring
 clear channel HC-332
 clear channel serial HC-333
 national reserved bits HC-329
E3 configuration mode HC-333
 frame type HC-329, HC-335
E3 mode, changing to T3 mode HC-330
 encapsulation command HC-45, HC-377, HC-431
 end command HC-396
Ethernet interface
 configuring
 flow control HC-84
 IP address and subnet mask HC-96
 MAC accounting HC-84, HC-99
 MAC address HC-85, HC-96
 MTU HC-85, HC-96
 configuring MAC accounting HC-84
 configuring the MAC address HC-84
 default settings
 flow control HC-84
 MAC accounting HC-84
 MAC address HC-85
 mtu HC-85
 displaying
 Ethernet interfaces HC-97
 MAC accounting statistics HC-99
displaying Ethernet interfaces HC-97
 enabling
 flow-control HC-96
 Layer 2 transport mode HC-100
Gigabit Ethernet standards  HC-86
   IEEE 802.3ab 1000BASE-T Gigabit Ethernet  HC-87
   IEEE 802.3ae 10 Gbps Ethernet  HC-87
   IEEE 802.3 Physical Ethernet Infrastructure  HC-87
   IEEE 802.3z 1000 Mbps Gigabit Ethernet  HC-87

Layer 2 VPN
   configuring Layer 2 protocol tunneling and data units  HC-101, HC-505
   Layer 2 protocol tunneling and data units, configuring  HC-101, HC-505
   overview  HC-85
   VLAN support  HC-498
using the l2protocol command  HC-101, HC-505
using the mac-accounting command  HC-84
using the mac address command  HC-85

VLANs
   802.1Q frames tagging  HC-496
assigning a VLAN AC  HC-500, HC-504
configuring native VLAN  HC-502
displaying VLAN interfaces  HC-251, HC-253, HC-503
Layer 2 VPN attachment circuit, configuring  HC-503
MTU inheritance  HC-497
native VLAN, configuring  HC-502
native VLAN description  HC-497
overview  HC-496
subinterface, configuring  HC-499
subinterface, removing  HC-506
subinterface overview  HC-497
using the dot1q vlan command  HC-500, HC-504
using the interface command  HC-502
using the show vlan interfaces command  HC-251, HC-253, HC-503
VLAN interfaces, displaying  HC-501
VLAN trunks, displaying  HC-501
exit command  HC-408, HC-409

F

Fast Ethernet interface
   auto-negotiation  HC-85
   configuring
      duplex operation  HC-85
      MAC accounting  HC-85
      MTU  HC-85
default settings
      auto-negotiation  HC-85
      duplex operation  HC-85
      interface speed  HC-85
      MAC accounting  HC-85
      mtu  HC-85
fdl command  HC-342
flow-control command  HC-84, HC-96

Frame Relay
   serial interfaces  HC-430
   framing command  HC-329

H

HDLC  HC-377
hw-module node reload command  HC-396
hw-module port command  HC-395
hw-module subslot cardtype command  HC-330, HC-331

I

IEEE 802.3ad standard  HC-217
if preconfiguration submode, ipv4 address command  HC-9
if submode
   bundle id command  HC-231, HC-232, HC-239, HC-244, HC-252
   controller sonet command  HC-321
duplex command  HC-74
interface command  HC-320
ip address command  HC-249, HC-250
ipv4 address command  HC-72
keepalive command  HC-320, HC-481
mtu command  HC-72
no shutdown command  HC-72, HC-231, HC-250, HC-252
pos crc command  HC-320
See interface command
See interface preconfigure command
show srp ips interface srp command  HC-406
speed command  HC-75
srp interface srp request manual-switch command  HC-406
srp ips request forced-switch command  HC-405, HC-411, HC-412
srp ips timer command  HC-405
srp ips wr-timer command  HC-405
srp topology-timer command  HC-402
tunnel destination command  RC-515
tunnel source command  RC-515
ILMI, on ATM interfaces  HC-51
interface Bundle-Ether command  HC-230, HC-236, HC-249
interface Bundle-POS command  HC-227
interface command  HC-28, HC-502
for Ethernet interfaces  HC-504
for VLAN subinterfaces  HC-504
Link Bundling  HC-252
PPP  HC-417
interface loopback command  HC-319
interface POS command  HC-244
interface preconfigure command  HC-9
interfaces
Link Bundling  HC-209
configuring  HC-228
link switchover  HC-228
prerequisites  HC-212
tunnel  RC-512
interface submode, service-policy command  HC-410
invert command  HC-437
ip address command
bundle-POS  HC-249, HC-250
IPS (Intelligent Protection Switching)  HC-403
ipv4 address command  HC-9, HC-28, HC-227, HC-249
Fast Ethernet  HC-72

K
keepalive command  HC-377, HC-385
keepalive timer
description  HC-378
monitoring
POS link state  HC-378

L
l2protocol command  HC-101, HC-505
l2transport command  HC-100
L2VPN
See Layer 2 VPN  HC-85
See Layer 2 VPN  HC-85
LACP (Link Aggregation Control Protocol)  HC-216
Layer 2 VPN
configuring
Layer 2 protocol tunneling and data units  HC-101, HC-505
Layer 2 transport mode  HC-100
configuring Layer 2 protocol tunneling and data units  HC-101, HC-505
overview  HC-85
LCP (Link Control Protocol)  HC-377, HC-427
link bundling
configuring VLAN bundles  HC-498
link bundling, configuring VLAN bundles  HC-498
link switchover  HC-228
Local Management Interface (LMI)  HC-430
loopback, naming convention  HC-299
loopback command  HC-315
Index

M

mac accounting command HC-99
mac-accounting command HC-84, HC-85
mac address command HC-85, HC-96
management ethernet interface, configuring HC-70
match mpls experimental topmost command HC-407
match precedence command HC-408
mdl string command HC-340
mdl transmit command HC-329, HC-340
mip auto-create command HC-123
mode command HC-333
MQC (Modular Quality of Service), configuring with SRP HC-406
MS-CHAP authentication
  disabling HC-474
  enabling HC-463, HC-464
password, configuring HC-470
ppp HC-378, HC-458, HC-470, HC-475
viewing HC-465
mtu command HC-85, HC-377, HC-432
Multi-Chassis Link Aggregation
  Access Network Redundancy Model HC-219
multilink command HC-483, HC-485
multilink fragment-size command HC-481
multilink interleave command HC-483
Multiprotocol Label Switching control processor (MPLSCP) HC-378, HC-428

N

naming conventions
  loopback HC-299
  null interface HC-299
  preconfigure HC-7
national bits command HC-329, HC-336
negotiation auto command HC-85, HC-97
Network Control Protocols (NCPs) HC-377, HC-427
no interface command HC-506
nonstop forwarding HC-228
no shutdown command (warning) HC-7
bundle-POS HC-250, HC-252
Fast Ethernet HC-72
null interface
  configuring HC-300
displaying HC-301
  naming convention HC-299

O

oam ais-rdi command HC-45
oam-pvc manage command HC-45
oam retry command HC-45
overhead command HC-315

P

PAP authentication
  defined HC-457
  disabling HC-471
  enabling HC-463, HC-464, HC-466, HC-467
  ppp HC-378, HC-466
  refusing HC-472
  viewing HC-465
path command HC-316, HC-323
path scrambling command HC-321
ping ethernet cfm command HC-171
Point-to-Point protocol
  See PPP
police command HC-408
policy-map command HC-408
policy map submode
  class command HC-408, HC-409
  police command HC-408
  policy-map command HC-408
  priority command HC-408, HC-409
set cos command  HC-408, HC-409

POS (Packet-over-SONET)
See  POS interface

POS interface

bringing up  HC-379
configuring
CRC value  HC-377, HC-383
capsulation type  HC-377
interface encapsulation  HC-383, HC-482
keepalive timer  HC-377, HC-384, HC-385, HC-481
MTU  HC-377, HC-383
optional parameters  HC-382
PPP authentication  HC-378
default settings
CRC  HC-377
capsulation  HC-377
keepalive  HC-377
mtu  HC-377

HDLC encapsulation
description
overview  HC-377

interface configuration mode
interface command  HC-481
interface multilink command  HC-481, HC-483
interface pos command  HC-380, HC-383

PPP encapsulation
description  HC-375
overview  HC-377

PPP

CHAP
authentication  HC-457
enabling  HC-463, HC-464
password, configuring  HC-468, HC-469
refusing  HC-473
viewing  HC-465
configuring  HC-416
default configuration, modifying  HC-459, HC-461
interfaces, displaying  HC-463

MS-CHAP
disabling  HC-474, HC-475
enabling  HC-464
authenticaion  HC-463
password, configuring  HC-470
ppp  HC-428, HC-458
viewing  HC-465
overview  HC-465

PAP
authentication  HC-428, HC-466
disabling  HC-471
enabling  HC-463, HC-464, HC-466, HC-467
refusing  HC-472
viewing  HC-465

POS configuration example  HC-486
POS interface  HC-375, HC-377
prerequisites  HC-456

serial configuration example  HC-487
serial interface  HC-427

ppp authentication command  HC-378, HC-428, HC-457, HC-464
ppp chap password command  HC-469
ppp chap refuse command  HC-473
ppp max-bad-auth command  HC-461
ppp max-configure command  HC-461
ppp max-failure command  HC-461
ppp max-terminate command  HC-462
ppp ms-chap password command  HC-470
ppp ms-chap refuse command  HC-475
ppp multilink minimum-active links command  HC-483
ppp pap refuse command  HC-472
ppp pap sent-username command  HC-466, HC-467
ppp timeout authentication command  HC-462
ppp timeout retry command  HC-462

preconfiguration
advantages  HC-7
directory  HC-5
naming conventions  HC-7
restriction to physical interface  HC-5
priority command HC-408, HC-409
profile command RC-515
PVC
   ATM interface HC-29
   ATM subinterface HC-27, HC-50
pvc command HC-28

R
rings (outer and inner) HC-392
RP, preconfiguration directory HC-5

S
scramble command HC-437
serial interface
   configuring
      CRC HC-437
      interface encapsulation HC-437
      IP address and subnet mask HC-434
      keepalive timer HC-444
      transmit delay HC-437
      data stream, inverting HC-437
      default settings
      CRC HC-432
      encapsulation HC-431
      keepalive HC-431
      mtu HC-432
      link state HC-427, HC-428, HC-429
      payload scrambling, enabling HC-437
      PPP encapsulation HC-427
      prerequisites HC-425
service-policy command HC-410
set cos command HC-408, HC-409
SF (Signal Fail) HC-404
show aps command HC-322
show atm ilmi-status command HC-53, HC-55
show bundle Bundle-Ether command HC-233, HC-240, HC-253
show bundle Bundle-POS command HC-245
show controllers command HC-334
show controller sonet command HC-317
show interfaces command HC-95, HC-450
   for Ethernet interfaces HC-97, HC-101
show ip route command RC-516
show lacp bundle Bundle-Ether command HC-233
show lacp bundle Bundle-POS command HC-245
show mac accounting command HC-99
show ppp interfaces command HC-463, HC-465
show running-config command HC-403
show srp ips interface srp command HC-406
show version command HC-95
show vlan command HC-251, HC-253, HC-501, HC-503
show vlan interface command HC-507
show vlan trunks command HC-241
SLARP (Serial Line Address Resolution Protocol) HC-379, HC-427, HC-429
SONET (Synchronous Optical Network)
   APS HC-317
      description HC-309
      fast reroute (FFR) HC-322
SONET APS (SONET Automatic Protection Switching) HC-317
SONET controller
   configuring HC-313
   frame type HC-315
   sonet submode
      ais-shut command HC-321
      clock source command HC-314, HC-321, HC-402
      delay trigger command HC-314, HC-323
      framing command HC-315
      loopback command HC-315
      overhead command HC-315
      path command HC-316, HC-323
      path scrambling command HC-321
See controller sonet command
SPAN HR-283
Spatial Reuse Protocol (SRP) HC-391
basic configuration HC-401
definition HC-391
MQC, configuring HC-406
node, adding to ring HC-410
PLIM, enabling on HC-394
SPA, enabling on HC-396, HC-398
speed command HC-85
srp interface srp request manual-switch command HC-406
srp ips request forced-switch command HC-405, HC-411, HC-412
srp ips timer command HC-405
srp ips wr-timer command HC-405
srp topology-timer command HC-402
SSH (Secure Shell) RC-513
SSH (Secure Shell), not needed with IPSec RC-513
SSL (Secure Socket Layer) RC-513
SSL (Secure Socket Layer), not needed with IPSec RC-513
switched port analyzer HR-283
switchover HC-228

T

T1 controller
ANSI T1.403 or AT&T TR54016 performance reports HC-342
card type, setting HC-330
clock source HC-342
default configuration values HC-329
DS0 timeslots, associating HC-342, HC-478
frame type HC-342
T1 channel group, creating HC-342, HC-478
T1 configuration mode HC-342, HC-478
T3 controller
cable length, setting HC-329
card type, setting HC-330
clock source
configuring HC-335, HC-339, HC-478
default value HC-329
configuring HC-337
BERT HC-345
clear channel E3 controller HC-333
clear channel T3 controller HC-337
MDL messages HC-329
frame type HC-329, HC-340
modifying
default E3 controller HC-335
default T3 controller HC-339
T3 mode, changing to E3 mode HC-330
timeslots command HC-342, HC-478
traffic filtering HC-300
tunnel destination command RC-515
tunnel source command RC-515

V

vc-class
ATM interfaces HC-44, HC-47
ATM PVCs HC-49
ATM subinterfaces HC-48
configuring HC-62
creating HC-62
vc-class command HC-44
virtual interface
active/standby RPs HC-8, HC-301
naming convention HC-299, RC-512
null interface definition HC-300
switchover HC-299
VLANs
802.1Q frames tagging HC-496
assigning a VLAN AC HC-500, HC-504
configuring
bundles HC-498
IP address and subnet mask HC-500
native VLAN HC-502
subinterfaces HC-499
configuring bundles HC-498
configuring native VLAN HC-502
displaying
  VLAN interfaces HC-501, HC-507
  VLAN trunks HC-501, HC-507
displaying VLAN interfaces HC-251, HC-253, HC-503
Layer 2 VPN
  attachment circuit, configuring HC-503
Layer 2 VPN support HC-498
MTU inheritance HC-498
native VLAN description HC-497
overview HC-496
subinterface overview HC-497
using the dot1q vlan command HC-500, HC-504
using the show vlan interfaces command HC-251, HC-253, HC-503
VLAN subinterface, removing HC-506

W

wait-to-restore (WTR) HC-403
wavelength division multiplexing (WDM) HC-392

Y

yellow command HC-342