Interface and Hardware Component Configuration Guide for Cisco NCS 6000 Series Routers, IOS XR Release 7.2.x

First Published: 2020-08-01

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020 Cisco Systems, Inc. All rights reserved.

CONTENTS

PREFACE

Preface xi

Changes to This Document xi

Obtaining Documentation and Submitting a Service Request xi

CHAPTER 1

New and Changed Interface and Hardware Component Features 1

Interface and Hardware Component Features Added or Modified in IOS XR Release 7.2.x 1

CHAPTER 2

Preconfiguring Physical Interfaces 3

Prerequisites for Preconfiguring Physical Interfaces 4

Information About Preconfiguring Physical Interfaces 4

Physical Interface Preconfiguration Overview 4

Benefits of Interface Preconfiguration 4

Use of the Interface Preconfigure Command 5

Active and Standby RPs and Virtual Interface Configuration 5

How to Preconfigure Physical Interfaces 6

Configuration Examples for Preconfiguring Physical Interfaces 7

Preconfiguring an Interface: Example 7

Additional References 8

CHAPTER 3

Advanced Configuration and Modification of the Management Ethernet Interface 11

Prerequisites for Configuring Management Ethernet Interfaces 12

Information About Configuring Management Ethernet Interfaces 12

Default Interface Settings 12

How to Perform Advanced Management Ethernet Interface Configuration 13

Configuring a Management Ethernet Interface 13

IPv6 Stateless Address Auto Configuration on Management Interface 15

Configuring the Duplex Mode for a Management Ethernet Interface 16
Configuring the Speed for a Management Ethernet Interface 18
Modifying the MAC Address for a Management Ethernet Interface 19
Verifying Management Ethernet Interface Configuration 20
Configuration Examples for Management Ethernet Interfaces 21
Configuring a Management Ethernet Interface: Example 21
Additional References 22

CHAPTER 4 **Configuring Ethernet Interfaces 23**
Prerequisites for Configuring Ethernet Interfaces 23
Information About Configuring Ethernet 24
Default Configuration Values for Gigabit Ethernet and 10-Gigabit Ethernet 24
Default Configuration Values for Fast Ethernet 24
Layer 2 VPN on Ethernet Interfaces 25
Gigabit Ethernet Protocol Standards Overview 26
IEEE 802.3 Physical Ethernet Infrastructure 26
IEEE 802.3ab 1000BASE-T Gigabit Ethernet 26
IEEE 802.3z 1000 Mbps Gigabit Ethernet 26
IEEE 802.3ae 10 Gbps Ethernet 26
IEEE 802.3ba 100 Gbps Ethernet 26
MAC Address 27
MAC Accounting 27
Ethernet MTU 27
Flow Control on Ethernet Interfaces 27
802.1Q VLAN 28
VRRP 28
HSRP 28
Duplex Mode on Fast Ethernet Interfaces 29
Fast Ethernet Interface Speed 29
Link Autonegotiation on Ethernet Interfaces 29
Link Layer Discovery Protocol (LLDP) 30
LLDP Frame Format 31
LLDP Operation 31
Supported LLDP Functions 32

Unsupported LLDP Functions 32
Carrier Delay on Ethernet Interfaces 32
How to Configure Ethernet 33
Configuring Ethernet Interfaces 33
Configuring Gigabit Ethernet Interfaces 33
What to Do Next 36
What to Do Next 36
Configuring MAC Accounting on an Ethernet Interface 36
Configuring a L2VPN Ethernet Port 38
What to Do Next 40
Configuring LLDP 40
LLDP Default Configuration 40
Enabling LLDP Globally 41
Configuring Global LLDP Operational Characteristics 41
Disabling Transmission of Optional LLDP TLVs 41
Disabling LLDP Receive and Transmit Operations for an Interface 41
Verifying the LLDP Configuration 43
Configuration Examples for Ethernet 44
Configuring an Ethernet Interface: Example 44
Configuring MAC-Accounting: Example 44
Configuring a Layer 2 VPN AC: Example 45
Configuring LLDP: Examples 45
Where to Go Next 45
Additional References 46

CHAPTER 5 **Configuring Ethernet OAM** 47
Prerequisites for Configuring Ethernet OAM 47
Information About Configuring Ethernet OAM 47
Ethernet Link OAM 47
Neighbor Discovery 48
Link Monitoring 48
MIB Retrieval 48
Miswiring Detection (Cisco-Proprietary) 48
SNMP Traps 49

How to Configure Ethernet OAM 49
Configuring Ethernet Link OAM 49
Configuring an Ethernet OAM Profile 49
Attaching an Ethernet OAM Profile to an Interface 55
Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration 56
Verifying the Ethernet OAM Configuration 58
Configuration Examples for EOAM Interfaces 58
Configuring an Ethernet OAM Profile Globally: Example 58
Configuring Ethernet OAM Features on an Individual Interface: Example 59
Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example 59
Clearing Ethernet OAM Statistics on an Interface: Example 60
Enabling SNMP Server Traps on a Router: Example 60
Where to Go Next 60
Additional References 61

CHAPTER 6 **Configuring Link Bundling** 63
Prerequisites for Configuring Link Bundling 63
Prerequisites for Configuring Link Bundling 64
Information About Configuring Link Bundling 64
Link Bundling Overview 64
Features and Compatible Characteristics of Link Bundles 65
Link Aggregation Through LACP 66
IEEE 802.3ad Standard 66
LACP Short Period Time Intervals 67
Load Balancing 68
VLANs on an Ethernet Link Bundle 68
Link Bundle Configuration Overview 69
Nonstop Forwarding During RP Switchover 69
Link Switchover 69
Bundle Fast Convergence 70
BFC Functionality 70
Condition for BFC 70
Sample BFC Data 70
How to Configure Link Bundling 71

Configuring Ethernet Link Bundles 71
Configuring EFP Load Balancing on an Ethernet Link Bundle 76
Configuring VLAN Bundles 78
Configuring the Default LACP Short Period Time Interval 84
Configuring Custom LACP Short Period Time Intervals 86
Configuration Examples for Link Bundling 91
Example: Configuring an Ethernet Link Bundle 91
Example: Configuring a VLAN Link Bundle 92
Example: Configuring EFP Load Balancing on an Ethernet Link Bundle 92
Examples: Configuring LACP Short Periods 92
Additional References 93

CHAPTER 7 **Configuring Traffic Mirroring** 95
Overview of Traffic Mirroring 95
ERSPAN 96
ERPAN with UDF 97
Traffic Mirroring Terminology 97
Characteristics of the Source Port 98
Characteristics of the Monitor Session 98
Characteristics of the Destination Port 98
Configure Traffic Mirroring 99

CHAPTER 8 **Configuring Virtual Loopback and Null Interfaces** 103
Prerequisites for Configuring Virtual Interfaces 103
Information About Configuring Virtual Interfaces 103
Virtual Loopback Interface Overview 104
Null Interface Overview 104
Virtual Management Interface Overview 104
Active and Standby RPs and Virtual Interface Configuration 104
How to Configure Virtual Interfaces 105
Configuring Virtual Loopback Interfaces 105
Configuring Null Interfaces 106
Configuring Virtual IPv4 Interfaces 108
Configuration Examples for Virtual Interfaces 109

Configuring a Loopback Interface: Example 109
Configuring a Null Interface: Example 109
Configuring a Virtual IPv4 Interface: Example 110

CHAPTER 9 **Configuring 802.1Q VLAN Interfaces 111**
Prerequisites for Configuring 802.1Q VLAN Interfaces 111
Information About Configuring 802.1Q VLAN Interfaces 112
802.1Q VLAN Overview 112
802.1Q Tagged Frames 112
Subinterfaces 112
Subinterface MTU 112
Native VLAN 113
VLAN Sub-interfaces on Ethernet Bundles 113
How to Configure 802.1Q VLAN Interfaces 113
Configuring 802.1Q VLAN Subinterfaces 113
Configuring 802.1Q VLAN Subinterfaces 113
Configuring an Attachment Circuit on a VLAN 115
What to Do Next 118
Removing an 802.1Q VLAN Subinterface 118
118
Configuration Examples for VLAN Interfaces 119
VLAN Subinterfaces: Example 120
Additional References 121

CHAPTER 10 **Configuring GRE Tunnels 123**
Configuring Static GRE 123
IP-in-IP De-capsulation 124
Single Pass GRE Encapsulation Allowing Line Rate Encapsulation 128
Configuration 128

CHAPTER 11 **Configuring Tunnel Interfaces 139**
Prerequisites for Configuring Tunnel Interfaces 140
Information About Configuring Tunnel Interfaces 140
Tunnel Interfaces Overview 140

Virtual Interface Naming Convention 140
Tunnel-IPSec Overview 140
Tunnel-IPSec Naming Convention 141
Crypto Profile Sets 141
How to Configure Tunnel Interfaces 141
Configuring Tunnel-IPSec Interfaces 141
Configuration Examples for Tunnel Interfaces 143
Tunnel-IPSec: Example 144
Where to Go Next 144

CHAPTER 12 **Configuring Dense Wavelength Division Multiplexing Controllers** 145
Prerequisites for Configuring DWDM Controller Interfaces 145
Information About the DWDM Controllers 146
Information about IPoDWDM 146
How to Configure DWDM Controllers 147
Configuring the Optical Parameters 147
Configuring G.709 Parameters 149
What to Do Next 151
Forward Error Correction (FEC) 151
Configuring FEC on an Interface 152
Configuring IPoDWDM 153
Configuring the Optical Layer DWDM Ports 153
Configuring the Administrative State of DWDM Optical Ports 154
Configuration Examples 156
Turning On the Laser: Example 156
Turning Off the Laser: Example 157
IPoDWDM Configuration: Examples 157
Optical Layer DWDM Port Configuration: Examples 157
Administrative State of DWDM Optical Ports Configuration: Examples 157
Additional References 157

CHAPTER 13 **Configuring IP-in-IP Decapsulation** 159
IP-in-IP Decapsulation 159

Preface

Note This product has reached end-of-life status. For more information, see the End-of-Life and End-of-Sale Notices.

The *Interface and Hardware Component Configuration Guide for Cisco NCS 6000 Series Routers* provides information and procedures related to router interface and hardware configuration.

The preface contains the following sections:

- Changes to This Document, on page xi
- Obtaining Documentation and Submitting a Service Request, on page xi

Changes to This Document

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

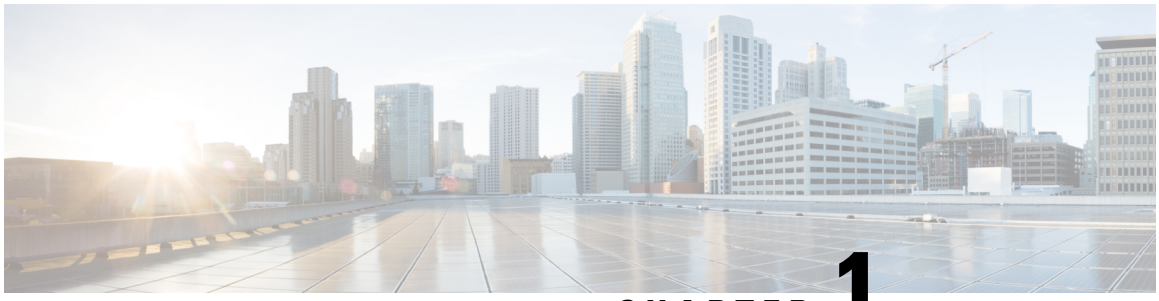
Table 1: Changes to This Document

Date	Summary
August 2020	Initial release of this document
January 2021	Republished for Release 7.2.2

Obtaining Documentation and Submitting a Service Request

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

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



CHAPTER 1

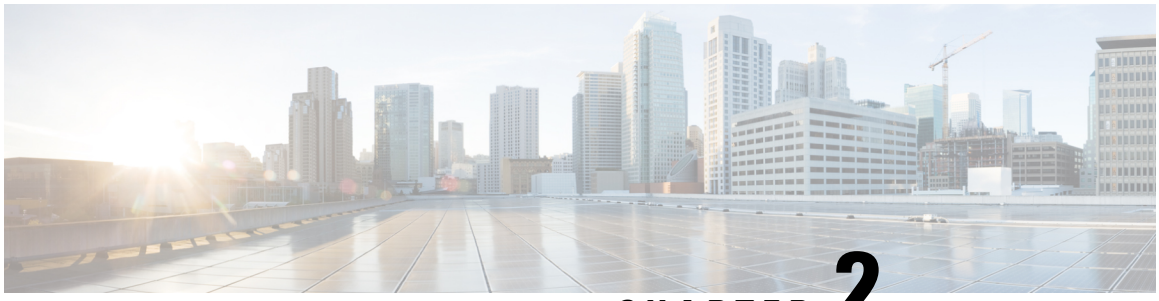
New and Changed Interface and Hardware Component Features

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

- Interface and Hardware Component Features Added or Modified in IOS XR Release 7.2.x, on page 1

Interface and Hardware Component Features Added or Modified in IOS XR Release 7.2.x

Feature	Description	Introduced/Changed in Release	Where Documented
IPv6 over GRE support	Generic Routing Encapsulation (GRE) is a tunneling protocol that provides a simple generic approach to transport packets of one protocol over another protocol by means of encapsulation using an IPv6 address.	Release 7.2.1	See chapter *Configuring GRE Tunnels*.
Forward Error Correction (FEC)	FEC support allows you to enable or disable RS-FEC (Reed-Solomon Codec for Ethernet IEEE 802.3 Clause 91) for optics.	Release 7.2.1	See chapter *Configuring DWDM* .



CHAPTER 2

Preconfiguring Physical Interfaces

This module describes the preconfiguration of physical interfaces.

Preconfiguration is supported for these types of interfaces and controllers:

- 10-Gigabit Ethernet
- Management Ethernet
- Serial
- SONET controllers and channelized SONET controllers

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

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

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

Note Only physical interfaces can be preconfigured.

Feature History for Preconfiguring Physical Interfaces

Release	Modification
Release 5.0.0	Ethernet interface preconfiguration was introduced.

- Prerequisites for Preconfiguring Physical Interfaces, on page 4
- Information About Preconfiguring Physical Interfaces, on page 4
- How to Preconfigure Physical Interfaces, on page 6
- Configuration Examples for Preconfiguring Physical Interfaces, on page 7
- Additional References, on page 8

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

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

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

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

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

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

Benefits of Interface Preconfiguration

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

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

Use of the Interface Preconfigure Command

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

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

Active and Standby RPs and Virtual Interface Configuration

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

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

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

After switchover , the virtual interfaces will all be present on the standby (now active) . 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 NCS 6000 Series 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. Do one of the following:
   - **ipv4 address** *ip-address subnet-mask*
   -
   - **ipv4 address** *ip-address*/*prefix*
4. Configure additional interface parameters, as described in this manual in the configuration chapter that applies to the type of interface that you are configuring.
5. Do one of the following:
   - **end**
   -
   - **commit** best-effort
6. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
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	Do one of the following: • **ipv4 address** *ip-address subnet-mask* • • **ipv4 address** *ip-address*/*prefix* **Example:**	Assigns an IP address and mask to the interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if-pre)# ipv4 address 192.168.1.2/32`	
Step 4	Configure additional interface parameters, as described in this manual in the configuration chapter that applies to the type of interface that you are configuring.	

Step 5	Do one of the following:
	• **end**
	•
	• **commit** best-effort
	Example:
	`RP/0/RP0/CPU0:router(config-if-pre)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-if-pre)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit best-effort** command to save the configuration changes to the running configuration file and remain within the configuration session. The **commit best-effort** command merges the target configuration with the running configuration and commits only valid changes (best effort). Some configuration changes might fail due to semantic errors.

Step 6	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	(Optional) Displays the configuration information currently running on the router.

Configuration Examples for Preconfiguring Physical Interfaces

This section contains the following example:

Preconfiguring an Interface: Example

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

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface preconfigure
 ten
GigE 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
```

Additional References

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

Related Documents

Related Topic	Document Title
Interface configuration commands	Interface and Hardware Component Command Reference for Cisco NCS 6000 Series Routers
Initial system bootup and configuration information	
Information about user groups and task IDs	*Cisco IOS XR Task ID Reference Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

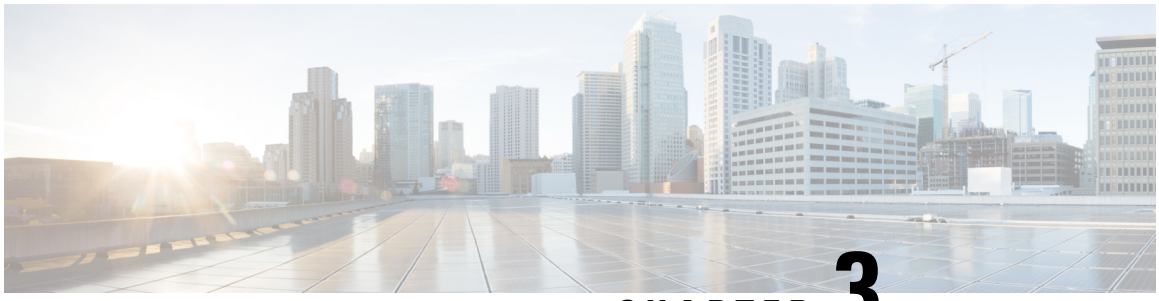
MIBs	MIBs Link
There are no applicable MIBs for this module.	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 3

Advanced Configuration and Modification of the Management Ethernet Interface

This module describes the configuration of Management Ethernet interfaces.

Before you can use Telnet to access the router through the LAN IP address, you must set up a Management Ethernet interface and enable Telnet servers, as described in the *Configuring General Router Features* module of the *System Setup and Software Installation Guide for Cisco NCS 6000 Series Routers* . This module describes how to modify the default configuration of the Management Ethernet interface after it has been configured, as described in *System Setup and Software Installation Guide for Cisco NCS 6000 Series Routers*.

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

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

Feature History for Configuring Management Ethernet Interfaces

Release	Modification
Release 5.0.0	This feature was introduced.

- Prerequisites for Configuring Management Ethernet Interfaces, on page 12
- Information About Configuring Management Ethernet Interfaces, on page 12
- How to Perform Advanced Management Ethernet Interface Configuration, on page 13
- Configuration Examples for Management Ethernet Interfaces, on page 21
- Additional References, on page 22

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, ensure that these tasks and conditions are met:

- You have performed the initial configuration of the Management Ethernet interface.
- You know how to apply the generalized interface name specification *rack/slot/module/port* .

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

Information About Configuring Management Ethernet Interfaces

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

Default Interface Settings

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

Table 2: Management Ethernet Interface Default Settings

Parameter	Default Value	Configuration File Entry
Speed in Mbps	Speed is autonegotiated.	**speed** [**10** | **100** | **1000**] To return the system to autonegotiate speed, use the **no speed** [**10** | **100** | **1000**] command.
Duplex mode	Duplex mode is autonegotiated.	**duplex** {**full** | **half**} To return the system to autonegotiated duplex operation, use the **no duplex {full | half}** command, as appropriate.
MAC address	MAC address is read from the hardware burned-in address (BIA).	**mac-address** address To return the device to its default MAC address, use the **no mac-address** address command.

How to Perform Advanced Management Ethernet Interface Configuration

This section contains the following procedures:

Configuring a Management Ethernet Interface

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

Note You do not need to perform this task if you have already set up the Management Ethernet interface to enable telnet servers.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:routerconfigure terminal`	Enters XR configuration mode.
Step 2	**interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack* /*slot* /*module* /*port* . The example indicates port 0 on the card that is installed in slot 0.
Step 3	**ipv4 address** *ip-address mask*	Assigns an IP address and subnet mask to the interface.

	Command or Action	Purpose

	Example:
	`RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224`
		• Replace *ip-address* with the primary IPv4 address for the interface.
		• Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways:
		  • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.
		  • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.

Step 4	**mtu** *bytes* **Example:** `RP/0/RP0/CPU0:router(config-if# mtu 1448`	(Optional) Sets the maximum transmission unit (MTU) byte value for the interface. The default is 1514. • The default is 1514 bytes. • The range for the Management Ethernet interface Interface **mtu** value is 64 to 1514 bytes. • The range for the Bundle Interface **mtu** value is 64 to 9500 bytes.
Step 5	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration, which removes the forced administrative down on the interface, enabling it to move to an up or down state.

Step 6	Do one of the following:
	• **end**
	• or
	• **commit**
	• or
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	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 XR EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

IPv6 Stateless Address Auto Configuration on Management Interface

The IPv6 Stateless Address Auto Configuration (SLAAC) is used to automatically assign IPv6 addresses to the host interfaces. This functionality can be used when the exact addresses used by the host need not be specific, as long as they are unique and properly routable. SLAAC helps in automating provisioning of the router.

IPv6 auto configuration is disabled by default. Perform this task to enable IPv6 SLAAC. The SLAAC functionality has to be enabled from the Management interface.

SUMMARY STEPS

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **ipv6 address autoconfig**
4. **show ipv6 interfaces** *interface-path-id*

DETAILED STEPS

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface MgmtEth** *interface-path-id*

Example:

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

Enters interface configuration mode on the specified Management Ethernet interface. Use the notation *rack*/*slot*/*module*/*port*.

This example indicates port 0 on the RP card that is installed in slot 0.

Step 3 **ipv6 address autoconfig**

Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv6 address autoconfig
```

Enable IPv6 stateless address auto configuration on the management port.

Step 4 **show ipv6 interfaces** *interface-path-id*

Example:

```
RP/0/RP0/CPU0:router# show ipv6 interfaces gigabitEthernet 0/2/0/0
```

Displays statistics for interfaces on the router. You can see in the output shown below that IPv6 is enabled and the IPv6 addresses has been auto configured.

Example

This example displays how to enable IPv6 SLAAC auto configuration on management interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:router(config)# ipv6 address autoconfig
RP/0/RP0/CPU0:router# show ipv6 interfaces MgmtEth 0/RP0/CPU0/1

MgmtEth0/RSP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::cecc:ccff:fecc:cccc
  Global unicast address(es):
    25::cecc:ccff:fecc:cccc, subnet is 25::/64
    2020:abcd:1234:1001:cecc:ccff:fecc:cccc, subnet is 2020:abcd:1234:1001::/64
    20::cecc:ccff:fecc:cccc, subnet is 20::/64
    447::cecc:ccff:fecc:cccc, subnet is 447::/64
    448::cecc:ccff:fecc:cccc, subnet is 448::/64
    13::cecc:ccff:fecc:cccc, subnet is 13::/64
    3457::cecc:ccff:fecc:cccc, subnet is 3457::/64
    19::cecc:ccff:fecc:cccc, subnet is 19::/64
  Joined group address(es): ff02::1:ffcc:cccc ff02::2 ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 2
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

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. Do one of the following:
   - **end**
   - or
   - **commit**
   - or

DETAILED STEPS

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

Step 4	Do one of the following:
	• **end**
	• or
	• **commit**
	• or
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	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 XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

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

Configuring the Speed for a Management Ethernet Interface

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

SUMMARY STEPS

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **speed** {**10** | **100** | **1000**}
4. Do one of the following:
   - **end**
   - or
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0`	Enters interface configuration mode and specifies the Management Ethernet interface name and instance.
Step 3	**speed** {**10**	**100**	**1000**} **Example:** `RP/0/RP0/CPU0:router(config-if)# speed 100`	Configures the interface speed parameter. **speed** options are or Mbps. **Note** The default Management Ethernet interface speed is autonegotiated. **Note** To return the system to the default autonegotiated speed, use the **no speed** command.
Step 4	Do one of the following: • **end** • or • **commit**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	Example:
	`RP/0/RP0/CPU0:router(config-if)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR 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. Do one of the following:
   - **end**
   - or
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface MgmtEth** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0`	Enters interface configuration mode and specifies the Management Ethernet interface name and instance.

	Command or Action	Purpose
Step 3	**mac-address** address **Example:** `RP/0/RP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD`	Configures the MAC layer address of the Management Ethernet interface. **Note** To return the device to its default MAC address, use the **no mac-address** address command.

Step 4	Do one of the following:
	• **end**
	• or
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	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 XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying Management Ethernet Interface Configuration

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

Configuration Examples for Management Ethernet Interfaces

This section provides the following configuration examples:

Configuring a Management Ethernet Interface: Example

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

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/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

MMgmtEth0/RP0/CPU0/0 is up, line protocol is up
Hardware is Management Ethernet, address is 0011.93ef.e8ea (bia 0011.93ef.e8ea
)
Description: Connected to Lab LAN
Internet address is 172.29.52.70/24
MTU 1514 bytes, BW 100000 Kbit
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set,
ARP type ARPA, ARP timeout 04:00:00
Last clearing of "show interface" counters never
5 minute input rate 3000 bits/sec, 7 packets/sec
5 minute output rate 0 bits/sec, 1 packets/sec
30445 packets input, 1839328 bytes, 64 total input drops
0 drops for unrecognized upper-level protocol
Received 23564 broadcast packets, 0 multicast packets
0 runts, 0 giants, 0 throttles, 0 parity
57 input errors, 40 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
171672 packets output, 8029024 bytes, 0 total output drops
Output 16 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
1 carrier transitions


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

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by the feature.	—

MIBs

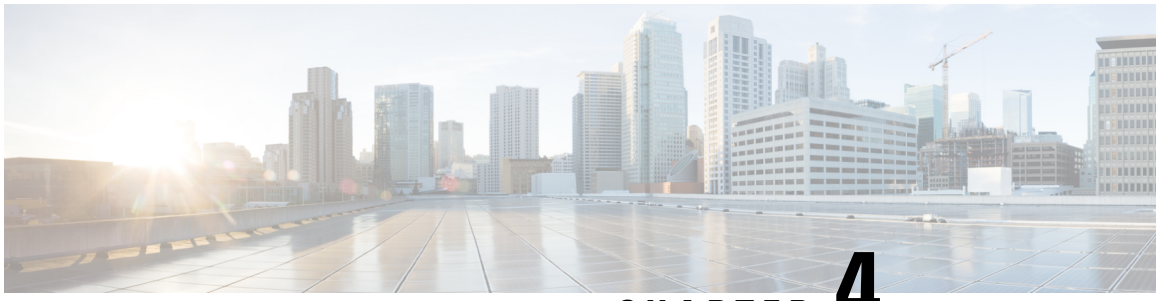
MIBs	MIBs Link
There are no applicable MIBs for this module.	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
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.	http://www.cisco.com/support



CHAPTER 4

Configuring Ethernet Interfaces

This module describes the configuration of Ethernet interfaces on the Cisco NCS 6000 Series Routers.

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

Feature History for Configuring Ethernet Interfaces

Release	Modification
Release 5.0.0	This feature was introduced.
Release 5.0.1	Support for Source MAC accounting was included.
Release 5.2.1	Support for Link Layer Discovery Protocol (LLDP) was included.

- Prerequisites for Configuring Ethernet Interfaces, on page 23
- Information About Configuring Ethernet , on page 24
- How to Configure Ethernet, on page 33
- Configuration Examples for Ethernet , on page 44
- Where to Go Next, on page 45
- Additional References, on page 46

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, ensure that these tasks and conditions are met:

- Know the interface IP address.
- You know how to apply the specify the generalized interface name with the generalized notation *rack/slot/module/port* .

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 NCS 6000 Series Router supports 10-Gigabit Ethernet (10 Gbps), and 100-Gigabit Ethernet (100 Gbps) interfaces.

Default Configuration Values for Gigabit Ethernet and 10-Gigabit Ethernet

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

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

Table 3: Gigabit Ethernet and 10-Gigabit Ethernet Modular Services Card Default Configuration Values

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

Default Configuration Values for Fast Ethernet

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

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

Table 4: Fast Ethernet Default Configuration Values

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

Layer 2 VPN on Ethernet Interfaces

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

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 network.

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

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

Gigabit Ethernet Protocol Standards Overview

The Gigabit Ethernet interfaces support the following protocol standards:

These standards are further described in the sections that follow.

IEEE 802.3 Physical Ethernet Infrastructure

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

MAC Address

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

MAC Accounting

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

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

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

Ethernet MTU

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

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

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

Note IPv6 does not support fragmentation.

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

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

Flow Control on Ethernet Interfaces

The flow control used on 10-Gigabit Ethernet interfaces consists of periodically sending flow control pause frames. It is fundamentally different from the usual full- and half-duplex flow control used on standard

management interfaces. Flow control can be activated or deactivated for ingress traffic only. It is automatically implemented for egress traffic.

802.1Q VLAN

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

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

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

VRRP

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

For more information on VRRP, see the *Implementing VRRP* module of *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* module of *IP Addresses and Services Configuration Guide.*

Duplex Mode on Fast Ethernet Interfaces

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

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

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

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

Fast Ethernet Interface Speed

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

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

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

Link Autonegotiation on Ethernet Interfaces

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

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

This table 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 5: Relationship Between duplex and speed Commands

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

Link Layer Discovery Protocol (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 acquire knowledge about, other Cisco devices connected to the network.

To support non-Cisco devices, and to allow for interoperability between other devices, the Cisco NCS 6000 Series Router also supports the IEEE 802.1AB Link Layer Discovery Protocol (LLDP).

LLDP is a neighbor discovery protocol that is used by network devices to advertise information about themselves, to other devices on the network. This protocol runs over the Data Link Layer, which permits 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 that is 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 identities can be advertised using this protocol.

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

- Port Description

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

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

LLDP Frame Format

LLDP frames use the IEEE 802.3 format, which consists of these 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 these basic formats:

- TLV Header (16 bits), which includes these 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 sending device, which is enabled for transmitting LLDP information, and which sends periodic advertisements of information in LLDP frames to a receiving device.

Devices are identified using a combination of 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 these additional operational characteristics:

- LLDP operates 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 these events occur:
  - The packet update frequency, specified by the **lldp timer** command, is reached. The default is 30.
  - A change in the values of the managed objects occurs from the local system's LLDP MIB.
  - 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 these actions on these TLV characteristics:

- • Interprets a TTL value of 0 as a request to automatically purge the information about 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.
  • If the TTL is non-zero, copy of an unknown organizationally-specific TLV is maintained, for later access through network management.

Supported LLDP Functions

The Cisco NCS 6000 Series Router supports these LLDP functions:

- IPv4 and IPv6 management addresses—In general, both IPv4 and IPv6 addresses are 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 is populated with an address from another interface. The advertised LLDP IP address is implemented according to this priority order of IP addresses for interfaces on the Cisco NCS 6000 Series Router Locally configured address
  - MgmtEth0/RP0/CPU0/0
  - MgmtEth0/RP0/CPU0/1
  - MgmtEth0/RP1/CPU0/0
  - MgmtEth0/RP1/CPU0/1
  - Loopback address

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

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

Unsupported LLDP Functions

- These LLDP functions are not supported: LLDP-MED organizationally unique extension—Interoperability, however, 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; Certain optional TLVs, however, can be disabled globally.

Carrier Delay on Ethernet Interfaces

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

Carrier Delay up suppresses short line flaps where the line is down, then comes up, then goes down again. A line that was previously down must be up longer than the duration specified for the **carrier-delay up** command

before the system is informed that the interface has come up. All flaps that are shorter than the duration specified for the **carrier-delay up** command are suppressed.

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

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

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

How to Configure Ethernet

This section provides the following configuration procedures:

Configuring Ethernet Interfaces

This section provides the following configuration procedures:

Configuring Gigabit Ethernet Interfaces

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show version** **Example:** `RP/0/RP0/CPU0:router# show version`	(Optional) Displays the current software version, and can also be used to confirm that the router recognizes the modular services card.
Step 2	**show interfaces** [**TenGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show TenGigE 0/1/0/0`	(Optional) Displays the configured interface and checks the status of each interface port. Possible interface types for this procedure are: • HundredGigE • TenGigE
Step 3	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters XR configuration mode.
Step 4	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interfaceTenGigE 0/1/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port* . Possible interface types for this procedure are: • HundredGigE • TenGigE **Note** The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.

Step 5	**ipv4 address** *ip-address mask*
	Example:
	`RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224`
		Assigns an IP address and subnet mask to the interface.
		• Replace *ip-address* with the primary IPv4 address for the interface.
		• Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways:
		  • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.
		  • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.

	Command or Action	Purpose
Step 6	**flow-control** {**bidirectional**| **egress** | **ingress**} **Example:** `RP/0/RP0/CPU0:router(config-if)# flow control ingress`	(Optional) Enables the sending and processing of flow control pause frames. • **egress**—Enables the sending of flow control pause frames in egress. • **ingress**—Enables the processing of received pause frames on ingress. • **bidirectional**—Enables the sending of flow control pause frames in egress and the processing of received pause frames on ingress.
Step 7	**mtu** *bytes* **Example:** `RP/0/RP0/CPU0:router(config-if)# mtu 1448`	(Optional) Sets the MTU value for the interface. • The default is 1514 bytes for normal frames and 1518 bytes for 802.1Q tagged frames. • The range for Gigabit Ethernet and 10-Gigabit Ethernet mtu values is 64 bytes to 65535 bytes.
Step 8	**mac-address** *value1.value2.value3* **Example:** `RP/0/RP0/CPU0:router(config-if)# mac address 0001.2468.ABCD`	(Optional) Sets the MAC layer address of the Management Ethernet interface. • The values are the high, middle, and low 2 bytes, respectively, of the MAC address in hexadecimal. The range of each 2-byte value is 0 to ffff.
Step 9	**negotiation auto** **Example:** `RP/0/RP0/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 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	Do one of the following: • **end** • • **commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	Command or Action	Purpose
	Example: **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode. • Entering **no** exits the configuration session and returns the router to XR 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** [**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, on page 36 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, on page 38 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 attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or Quality of Service (QoS), to the Ethernet interface, refer to the appropriate configuration guide.

Configuring MAC Accounting on an Ethernet Interface

This task explains how to configure MAC accounting on an Ethernet interface. MAC accounting has special show commands, which are illustrated in this procedure. Otherwise, the configuration is the same as configuring a basic Ethernet interface, and the steps can be combined in one configuration session. See "Configuring Gigabit Ethernet Interfaces, on page 33" in this module for information about configuring the other common parameters for Ethernet interfaces.

SUMMARY STEPS

1. **configure**
2. **interface** [**TenGigE** | **fastethernet**] *interface-path-id*
3. **ipv4 address** *ip-address mask*

4. **mac-accounting** {**egress** | **ingress**}
5. Do one of the following:
   - **end**
   -
   - **commit**
6. **show mac-accounting** *type* **location** *instance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface** [**TenGigE**	**fastethernet**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (**?**) online help function.

Step 3	**ipv4 address** *ip-address mask*
	Example:
	`RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224`
		Assigns an IP address and subnet mask to the interface.
		• Replace *ip-address* with the primary IPv4 address for the interface.
		• Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways:
		  • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.
		  • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.

Step 4	**mac-accounting** {**egress**	**ingress**} **Example:** `RP/0/RP0/CPU0:router(config-if)# mac-accounting egress`	Generates accounting information for IP traffic based on the source and destination MAC addresses on LAN interfaces. • To disable MAC accounting, use the **no** form of this command.
Step 5	Do one of the following:	Saves configuration changes.

	Command or Action	Purpose

	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	Example:
	`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 XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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

Configuring a L2VPN Ethernet Port

Use the following procedure to configure an L2VPN Ethernet port.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface** [**TenGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interfaceTenGigE 0/1/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port* . Possible interface types for this procedure are: • TenGigE
Step 3	**l2transport** **Example:** `RP/0/RP0/CPU0:router(config-if)# l2transport`	Enables Layer 2 transport mode on a port and enter Layer 2 transport configuration mode.
Step 4	**l2protocol** {**cdp** | **pvst** | **stp** | **vtp**} {[**forward** | **tunnel**][**experimental** *bits*]|**drop**} **Example:** `RP/0/RP0/CPU0:router(config-if-l2)# l2protocol stp tunnel` **Example:**	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	Do one of the following: • **end** • or • **commit** • or **Example:** `RP/0/RP0/CPU0:router(config-if-l2)# end` **Example:**	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 XR EXEC mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if-l2)# commit`	• Entering **no** exits the configuration session and returns the router to XR 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** [**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 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 these configuration topics for LLDP:

LLDP Default Configuration

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

Table 6: LLDP Default Configuration

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

Enabling LLDP Globally

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

You can override this default operation at the interface to disable receive or transmit operations. For more information about how to selectively disable LLDP receive or transmit operations for an interface, see the Disabling LLDP Receive and Transmit Operations for an Interface, on page 41.

To enable LLDP globally, complete these steps:

Configuring Global LLDP Operational Characteristics

The LLDP Default Configuration, on page 40 section 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 these steps:

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 these steps:

Disabling LLDP Receive and Transmit Operations for an Interface

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

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

SUMMARY STEPS

1. **configure**
2. **interface GigabitEthernet 0/2/0/0**
3. **lldp**
4. **receive disable**
5. **transmit disable**
6. Do one of the following:
   - **end**
   - or
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface GigabitEthernet 0/2/0/0** **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port* . Possible interface types for this procedure are: • GigabitEthernet • TenGigE
Step 3	**lldp** **Example:** `RP/0/RP0/CPU0:router(config-if)# lldp`	(Optional) Enters LLDP configuration mode for the specified interface.
Step 4	**receive disable** **Example:** `RP/0/RP0/CPU0:router(config-lldp)# receive disable`	(Optional) Disables LLDP receive operations on the interface.
Step 5	**transmit disable** **Example:** `RP/0/RP0/CPU0:router(config-lldp)# transmit disable`	(Optional) Disables LLDP transmit operations on the interface.

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

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

Verifying the LLDP Configuration

This section describes how to 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 this 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, this 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 this example:

```
RP/0/RP0/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 these commands:

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

Command	Description
show lldp traffic	Displays statistics for LLDP traffic

Configuration Examples for Ethernet

This section provides the following configuration examples:

Configuring an Ethernet Interface: Example

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

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


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

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

Configuring MAC-Accounting: Example

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

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

Configuring a Layer 2 VPN AC: Example

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

Configuring LLDP: Examples

This 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
```

This 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 module in this document.

For information about IPv6 see the *Implementing Access Lists and Prefix Lists* module in the *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

Related Topic	Document Title
Cisco IOS XR master command reference	Cisco IOS XR *Master Commands List*

Standards

Standards	Title
IEEE 802.1ag ITU-T Y.1731	*Ethernet OAM Connectivity Fault Management*

MIBs

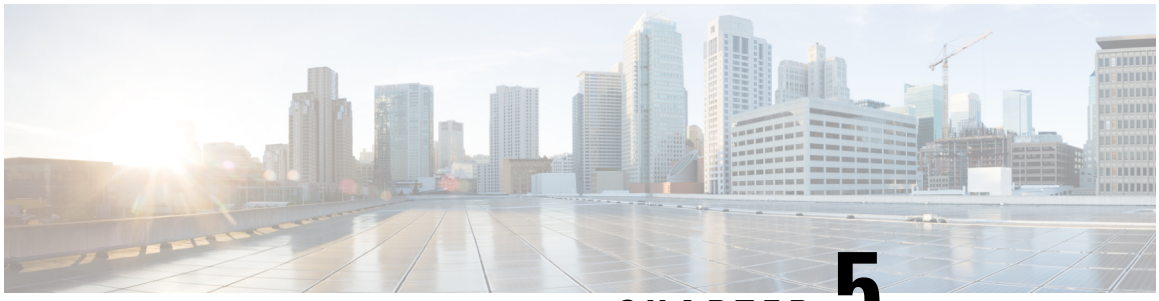
MIBs	MIBs Link
IEEE CFM MIB	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 5

Configuring Ethernet OAM

This module describes the configuration of Ethernet Operations, Administration, and Maintenance (OAM) on the Cisco NCS 6000 Series Router.

Feature History for Configuring Ethernet OAM

Release	Modification
Release 5.0.0	This feature was introduced.

- Prerequisites for Configuring Ethernet OAM , on page 47
- Information About Configuring Ethernet OAM , on page 47
- How to Configure Ethernet OAM, on page 49
- Configuration Examples for EOAM Interfaces, on page 58
- Where to Go Next, on page 60
- Additional References, on page 61

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.

Information About Configuring Ethernet OAM

To configure Ethernet OAM, you should understand these concepts:

Ethernet Link OAM

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

providers can monitor specific events, take actions on events, and if necessary, put specific interfaces into loopback mode for troubleshooting. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet link OAM can be configured in the following ways:

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

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

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

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

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

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

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

Neighbor Discovery

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

Link Monitoring

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

MIB Retrieval

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

Miswiring Detection (Cisco-Proprietary)

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

SNMP Traps

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

How to Configure Ethernet OAM

This section provides the following configuration procedures:

Configuring Ethernet Link OAM

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

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

Configuring an Ethernet OAM Profile

Perform these steps to configure an Ethernet OAM profile.

SUMMARY STEPS

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

23. **action high-threshold** {**error-disable-interface** | **log**}
24. **action session-down** {**disable** | **efd** | **error-disable-interface**}
25. **action session-up disable**
26. **action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**}
27. **action wiring-conflict** {**disable | efd | log**}
28. **uni-directional link-fault detection**
29. **commit**
30. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame window 6000`	The default value is 1000.
Step 7	**frame threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000`	(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.

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

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

	Command or Action	Purpose
		threshold of 100, then the threshold would be crossed if there are 1200 errored frames in that period (that is, 100 per million for 12 million).
Step 10	**frame-seconds window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds window 900000`	(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000. The default value is 60000. **Note** The only accepted values are multiples of the line cardinterface module-specific polling interval, that is, 1000 milliseconds for most line cardsinterface modules.
Step 11	**frame-seconds threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 high 900`	(Optional) Configures the thresholds (in seconds) that trigger a frame-seconds error event. The high threshold value can be configured only in conjunction with the low threshold value. The range is 1 to 900 The default value is 1.
Step 12	**exit** **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# exit`	Exits back to Ethernet OAM mode.
Step 13	**mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mib-retrieval`	Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.
Step 14	**connection timeout** <*timeout*> **Example:** `RP/0/RP0/CPU0:router(config-eoam)# connection timeout 30`	Configures the connection timeout period for an Ethernet OAM session. as a multiple of the hello interval. The range is 2 to 30. The default value is 5.
Step 15	**hello-interval** {**100ms**|**1s**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# hello-interval 100ms`	Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (**1s**).
Step 16	**mode** {**active**|**passive**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mode passive`	Configures the Ethernet OAM mode. The default is active.

	Command or Action	Purpose
Step 17	**require-remote mode** {**active**|**passive**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# require-remote mode active`	Requires that active mode or passive mode is configured on the remote end before the OAM session becomes active.
Step 18	**require-remote mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval`	Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.
Step 19	**action capabilities-conflict** {**disable** | **efd** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action capabilities-conflict efd`	Specifies the action that is taken on an interface when a capabilities-conflict event occurs. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 20	**action critical-event** {**disable** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action critical-event error-disable-interface`	Specifies the action that is taken on an interface when a critical-event notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 21	**action discovery-timeout** {**disable** | **efd** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action discovery-timeout efd`	Specifies the action that is taken on an interface when a connection timeout occurs. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 22	**action dying-gasp** {**disable** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface`	Specifies the action that is taken on an interface when a dying-gasp notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry.

	Command or Action	Purpose
		Note • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 23	**action high-threshold** {**error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action high-threshold error-disable-interface`	Specifies the action that is taken on an interface when a high threshold is exceeded. The default is to take no action when a high threshold is exceeded. **Note** • If you change the default, the **disable** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and take no action at the interface when the event occurs.
Step 24	**action session-down** {**disable** | **efd** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`	Specifies the action that is taken on an interface when an Ethernet OAM session goes down. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 25	**action session-up disable** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-up disable`	Specifies that no action is taken on an interface when an Ethernet OAM session is established. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 26	**action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**}	Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

	Command or Action	Purpose
Step 27	**action wiring-conflict** {**disable** | **efd** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`	Specifies the action that is taken on an interface when a wiring-conflict event occurs. The default is to put the interface into error-disable state. **Note** • If you change the default, the **error-disable-interface** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and put the interface into error-disable state when the event occurs.
Step 28	**uni-directional link-fault detection** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# uni-directional link-fault detection`	Enables detection of a local, unidirectional link fault and sends notification of that fault to an Ethernet OAM peer.
Step 29	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves the configuration changes to the running configuration file and remains within the configuration session.
Step 30	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Attaching an Ethernet OAM Profile to an Interface

Perform 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

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

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

Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

Verifying the Ethernet OAM Configuration

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

```
RP/0/RP0/CPU0:router# show ethernet oam configuration
Thu Aug  5 22:07:06.870 DST
GigabitEthernet0/4/0/0:
  Hello interval:                                          1s
  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
```

Configuration Examples for EOAM Interfaces

This section provides the following configuration examples:

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

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

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

```
configure terminal
 ethernet oam profile Profile_1
  mode passive
  action dying-gasp disable
  action critical-event disable
  action discovery-timeout disable
  action session-up disable
  action session-down disable
  action capabilities-conflict disable
  action wiring-conflict disable
  action remote-loopback disable
  action uni-directional link-fault error-disable-interface
  commit
configure terminal
 interface TenGigE 0/1/0/0
  ethernet oam
   profile Profile_1
    mode active
    action dying-gasp log
    action critical-event log
    action discovery-timeout log
    action session-up log
    action session-down log
    action capabilities-conflict log
    action wiring-conflict log
    action remote-loopback log
    action uni-directional link-fault log
    uni-directional link-fault detection
    commit
```

Clearing Ethernet OAM Statistics on an Interface: Example

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
```

Where to Go Next

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

For information about IPv6 see the *Implementing Access Lists and Prefix Lists* module in the *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

Related Topic	Document Title
Cisco IOS XR interface configuration commands	*Cisco IOS XR Interface and Hardware Component Command Reference*
Information about user groups and task IDs	*Cisco IOS XR Interface and Hardware Component Command Reference*

Standards

Standards	Title
IEEE 802.1ag	*Connectivity Fault Management*
ITU-T Y.1731	*OAM Functions and Mechansims for Ethernet Based Networks*
MEF 16	Metro Ethernet Forum, Technical Specification MEF 16, Ethernet Local Management Interface (E-LMI), January 2006

MIBs

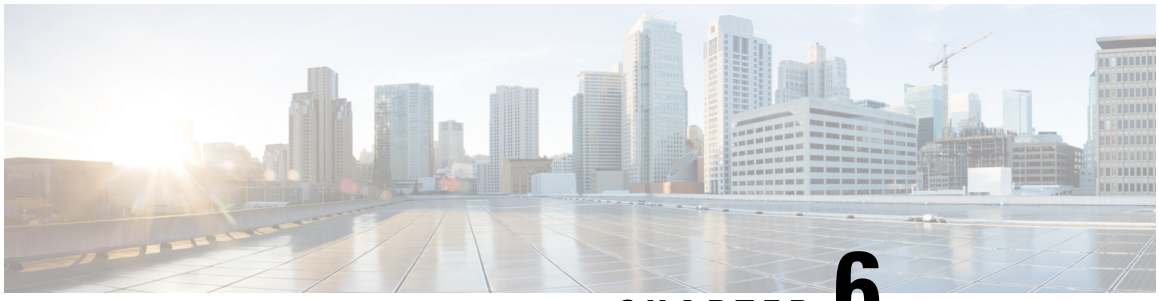
MIBs	MIBs Link
IEEE8021-CFM-MIB	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 6

Configuring Link Bundling

This module describes the configuration of link bundle interfaces on the Cisco NCS 6000 Series Router.

A link bundle is a group of one or more ports that are aggregated together and treated as a single link. 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 you 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.

Each bundle has a single MAC and shares a single Layer 3 configuration set, such as IP address, ACL, Quality of Service (QoS), and so on.

Note Link bundles do not have a one-to-one modular services card association. Member links can terminate on different cards.

Feature History for Configuring Link Bundling

Release	Modification
Release 5.2.5	Bundle Fast Convergence feature was added.
Release 5.0.0	This feature was introduced.

- Prerequisites for Configuring Link Bundling, on page 63
- Information About Configuring Link Bundling, on page 64
- How to Configure Link Bundling, on page 71
- Configuration Examples for Link Bundling, on page 91
- Additional References, on page 93

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

Before configuring link bundling, be sure that these 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:
  - 4-Port 10-Gigabit Ethernet PLIM
  - 8-Port 10-Gigabit Ethernet PLIM
  - 10-Port Gigabit Ethernet SPA
  - 42-Port Gigabit Ethernet PLIM
  - 1-Port 100-Gigabit Ethernet PLIM

Information About Configuring Link Bundling

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

Link Bundling Overview

The Link Bundling feature allows you to group multiple point-to-point links together into one logical link and provide higher bidirectional bandwidth, redundancy, and load balancing between two routers. A virtual interface is assigned to the bundled link. The component links can be dynamically added and deleted from the virtual interface.

The virtual interface is treated as a single interface on which one can configure an IP address and other software features used by the link bundle. Packets sent to the link bundle are forwarded to one of the links in the bundle.

The advantages of link bundles are as follows:

- Multiple links can span several line cards 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 if one of the links within a bundle fails. can without interrupting packet flow.

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 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 these features:

- ACL
- Basic IP
- Basic MPLS
- MPLS VPN
- Sampled Netflow
- BGP Policy Accounting
- HSRP/VRRP
- VLAN Bundling (Ethernet only)
- 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.
- For Ethernet link bundling, links within a single bundle should have the same speed.
- For POS link bundling, 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.
- Mixed bandwidth bundle member configuration is only supported when 1:1 redundancy is configured (this means that a 1 GigabitEthernet member can only be configured as the backup of the 10 GigabitEthernet interface).
- Mixed link bundle mode is supported only when active standby operation is configured (usually with the lower speed link in standby mode).
- A bundle can be administratively enabled or disabled.
- Each individual link within a bundle can be administratively enabled or disabled.
- 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 sub-interfaces cannot be bundle members. However, you can create VLANs as sub-interfaces 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.
-

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.

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.

LACP Short Period Time Intervals

As packets are exchanged across member links of a bundled interface, some member links may slow down or time-out and fail. LACP packets are exchanged periodically across these links to verify the stability and reliability of the links over which they pass. The configuration of short period time intervals, in which LACP packets are sent, enables faster detection and recovery from link failures.

Short period time intervals are configured as follows:

- In milliseconds
- In increments of 100 milliseconds
- In the range 100 to 1000 milliseconds
- The default is 1000 milliseconds (1 second)
- Up to 64 member links
- Up to 1280 packets per second (pps)

After 6 missed packets, the link is detached from the bundle.

When the short period time interval is not configured, LACP packets are transmitted over a member link every 30 seconds by default.

When the short period time interval is configured, LACP packets are transmitted over a member link once every 1000 milliseconds (1 second) by default. Optionally, both the transmit and receive intervals can be configured to less than 1000 milliseconds, independently or together, in increments of 100 milliseconds (100, 200, 300, and so on).

When you configure a custom LACP short period transmit interval at one end of a link, you must configure the same time period for the receive interval at the other end of the link.

Note You must always configure the transmit interval at both ends of the connection before you configure the receive interval at either end of the connection. Failure to configure the transmit interval at both ends first results in route flapping (a route going up and down continuously). When you remove a custom LACP short period, you must do it in reverse order. You must remove the receive intervals first and then the transmit intervals.

Load Balancing

Load balancing is a forwarding mechanism which distributes traffic over multiple links, based on Layer 3 routing information in the router. Per-flow load balancing is supported on all links in the bundle. This scheme achieves load sharing by allowing the router to distribute packets over one of the links in the bundle, that is determined through a hash calculation. The hash calculation is an algorithm for link selection based on certain parameters.

The standard hash calculation is a 3-tuple hashing, using the following parameters:

- IP source address
- IP destination address
- Router ID

7-tuple hashing can also be configured. based on Layer 3 and Layer 4 parameters:

- IP source address
- IP destination address
- Router ID
- Input interface
- IP protocol
- Layer 4 source port
- Layer 4 destination port

When per-flow load balancing and 3-tuple hashing is enabled, all packets for a certain source-destination pair will go through the same link, though there are multiple links available. Per-flow load balancing ensures that packets for a certain source-destination pair arrive in order.

Note For multicast traffic, ingress forwarding is based on the Fabric Multicast Group Identifier (FGID). Egress forwarding over the bundle is based on the bundle load balancing.

VLANs on an Ethernet Link Bundle

802.1Q VLAN subinterfaces can be configured on 802.3ad Ethernet link bundles. Keep the following information in mind when adding VLANs on an Ethernet link bundle:

- The maximum number of VLANs allowed per bundle is 128.
- The maximum number of bundled VLANs allowed per router is 4000.

Note The memory requirement for bundle VLANs is slightly higher than standard physical interfaces.

To create a VLAN subinterface on a bundle, include the VLAN subinterface instance with the **interface Bundle-Ether** command, as follows:

interface Bundle-Ether *interface-bundle-id.subinterface*

After you create a VLAN on an Ethernet link bundle, all VLAN subinterface configuration is supported on that link bundle.

VLAN sub-interfaces 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 XR 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. You can 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 NCS 6000 Series Router. If one member link in a bundle fails, traffic is redirected to the remaining operational member links.

Bundle Fast Convergence

The Bundle Fast Convergence (BFC) feature provides the ability to converge bundle members within sub seconds instead of multiple seconds. This feature provides faster bundle member convergence with deterministic traffic outage bounded within 50 milliseconds.

On bundle member shut, the packet drop is reduced to less than 50 milliseconds. On multiple members shut, the loss is less than (n*50ms), where 'n' is the number of members being shut.

BFC Functionality

The BFC feature decouples the tasks FIB is performing for bundle membership updates. This splits the FIB's bundle membership tasks into two separate threads:

- FAST Update (FRR thread): This option does In-Place-Modify (IPM) only, which guarantees deterministic outage time. FIB gets the bundle member down directly from Bundle Interface Manager (BIM) using Fast Protect Infra (similar to TE FRR update model).
- SLOW Update (Adjacency thread): Adjacency update from AIB, which is the current processing logic; does the following tasks:
  - IPM on the current hardware entries
  - Create a new set of hardware entries with new membership information

BFC tracks the bundle members that are down from FAST channel by setting a pending flag, then when SLOW update AIB is received it looks for FAST down member. If the BFC finds the FAST down member, it clears the pending flag, if it does not, it processes the SLOW update and considers the FAST down member as down and programs to the hardware.

Condition for BFC

The bundle must contain N members (they may be of different bandwidth, belong to different slices or Line Cards) and at the most N-1 members could be shut down in a single or multiple commits.

Note Shutting down bundle member link/s in single or serial commits is the only valid trigger for BFC. Bundle fast convergence time is calculated based on Frame delta and number of links affected. Convergence time < Number of links affected * 50ms --> 0 (50ms). Convergence time must be < 50ms per flow in the traffic.

Sample BFC Data

The following example shows convergence time data for sixteen individual flows that traffic was sent onto on dual router topology.

This data is taken for a bundle which has ten members including one 100Gig and nine 10Gig members where nine members (including 100Gig) are shut down with individual commits in order to trigger the BFC feature.

Frame Delta	Packet Loss Duration (ms)	Direction
337	6.667	Egress
61	1.187	Egress

254	4.942	Egress
0	0	Egress
255	4.961	Egress
255	4.961	Egress
47	0.914	Egress
0	0	Egress
1226	23.853	Ingress
866	16.849	Ingress
240	4.669	Ingress
1131	22.005	Ingress
0	0	Ingress
526	10.234	Ingress
1243	24.187	Ingress
264	5.136	Ingress

How to Configure Link Bundling

This section contains the following procedures:

Configuring Ethernet Link Bundles

This section describes how to configure an Ethernet link bundle.

Note 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.

Tip You can programmatically perform the configuration using `openconfig-lacp.yang`, `openconfig-if-aggregate.yang` OpenConfig data models or `Cisco-IOS-XR-um-lacp-cfg.yang` Unified data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco NCS 6000 Series Routers*.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface** {**TenGigE**} interface-path-id
10. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
11. **bundle port-priority** *priority*
12. **no shutdown**
13. **exit**
14. **interface** {**TenGigE**} number
15. Do one of the following:
    - **end**
    -
    - **commit**
16. **exit**
17. **exit**
18. Perform Step 1 through Step 15 on the remote end of the connection.
19. **show bundle Bundle-Ether** bundle-id
20. **show lacp bundle Bundle-Ether** *bundle-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3`	Creates a new Ethernet link bundle with the specified bundle-id. The range is 1 to 65535. This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface specific configuration commands are entered. Use the **exit** command to exit from the interface configuration submode back to the normal XR configuration mode.

	Command or Action	Purpose
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization. **Note** The priority of the active and standby links is based on the value of the **bundle port-priority** command.
Step 7	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration submode for the Ethernet link bundle.
Step 9	**interface** {**TenGigE**} interface-path-id **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0`	Enters interface configuration mode for the specified interface. Enter the **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the *rack*/*slot*/*module* format.
Step 10	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]	Adds the link to the specified bundle.

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the link to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** If you do not specify the **mode** keyword, the default mode is **on (**LACP is not run over the port).
Step 11	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.
Step 12	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The no shutdown command returns the link to an up or down state depending on the configuration and state of the link.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration submode for the Ethernet interface.
Step 14	**interface** {**TenGigE**} number **Example:** **bundle id** bundle-id [**mode** {**active** | **passive** | **on**}] **Example:** **no shutdown** **Example:** **exit** **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 1/0/2/1` **Example:**	(Optional) Repeat Step 8 through Step 11 to add more links to the bundle.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-if)# bundle id 3 **Example:** **Example:** RP/0/RP0/CPU0:router(config-if)# bundle port-priority 2 **Example:** RP/0/RP0/CPU0:router(config-if)# no shutdown **Example:** **Example:** RP/0/RP0/CPU0:router(config-if)# exit **Example:** **Example:** RP/0/RP0/CPU0:router(config)# interface TenGigE 1/0/2/3 **Example:** **Example:** RP/0/RP0/CPU0:router(config-if)# bundle id 3 **Example:** **Example:** RP/0/RP0/CPU0:router(config-if)# no shutdown **Example:** **Example:** RP/0/RP0/CPU0:router(config-if)# exit	
Step 15	Do one of the following: • **end** • • **commit** **Example:**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before

	Command or Action	Purpose

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

Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration mode.
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits XR configuration mode.
Step 18	Perform Step 1 through Step 15 on the remote end of the connection.	Brings up the other end of the link bundle.
Step 19	**show bundle Bundle-Ether** bundle-id **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3`	(Optional) Shows information about the specified Ethernet link bundle.
Step 20	**show lacp bundle Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show lacp bundle Bundle-Ether 3`	(Optional) Shows detailed information about LACP ports and their peers.

Configuring EFP Load Balancing on an Ethernet Link Bundle

This section describes how to configure Ethernet flow point (EFP) Load Balancing on an Ethernet link bundle.

By default, Ethernet flow point (EFP) load balancing is enabled. However, the user can choose to configure all egressing traffic on the fixed members of a bundle to flow through the same physical member link. This configuration is available only on an Ethernet Bundle subinterface with Layer 2 transport (**l2transport**) enabled.

Note If the active members of the bundle change, the traffic for the bundle may get mapped to a different physical link that has a hash value that matches the configured value.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id* **l2transport**
3. Do one of the following:
   - **bundle load-balance hash** *hash-value* [**auto**]
4. Do one of the following:
   - **end**
   -
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **l2transport** **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3 l2transport`	Creates a new Ethernet link bundle with the specified *bundle-id* and with Layer 2 transport enabled. The range is 1 to 65535.

Step 3	Do one of the following:
	• **bundle load-balance hash** *hash-value* [**auto**]
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto`
		Configures all egressing traffic on the fixed members of a bundle to flow through the same physical member link.
		• *hash-value*—Numeric value that specifies the physical member link through which all egressing traffic in this bundle will flow. The values are 1 through 8.
		• **auto**—The physical member link through which all egressing traffic on this bundle will flow is automatically chosen.

Step 4	Do one of the following: • **end** •	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-if)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR 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 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 describe in detail in the procedure that follows.

Note In order for a VLAN bundle to be active, you must perform the same configuration on both ends of the bundle connection.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface Bundle-Ether** *bundle-id.vlan-id*
10. **ipv4 address** *ipv4-address mask*
11. **no shutdown**
12. **exit**

13. Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.
14. Do one of the following:
    - **end**
    - or
    - **commit**
15. **exit**
16. **exit**
17. **configure**
18. **interface** {**TenGigE**} interface-path-id
19. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
20. **bundle port-priority** *priority*
21. **no shutdown**
22. —
23. Do one of the following:
    - **end**
    -
    - **commit**
24. Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.
25. **show bundle Bundle-Ether** bundle-id
26. **show vlan interface**
27. **show vlan trunks** [{**TenGigE** | **Bundle-Ether**} *interface-path-id*] [**brief** | **summary**] [**location** *node-id*]
28. **lacp fast-switchover**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3`	Creates and names a new Ethernet link bundle. This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the **exit** command to exit from the interface configuration submode back to the normal XR configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.

	Command or Action	Purpose
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization. **Note** The priority of the active and standby links is based on the value of the **bundle port-priority** command.
Step 7	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits the interface configuration submode.
Step 9	**interface Bundle-Ether** *bundle-id.vlan-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3.1`	Creates a new VLAN, and assigns the VLAN to the Ethernet bundle you created in Step 2. Replace the *bundle-id* argument with the *bundle-id* you created in Step 2. Replace the *vlan-id* with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved). **Note** When you include the *.vlan-id* argument with the **interface Bundle-Ether** *bundle-id* command, you enter subinterface configuration mode.

	Command or Action	Purpose
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.
Step 12	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	Exits subinterface configuration mode for the VLAN subinterface.

Step 13	Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.
	Example:
	Example:
	interface Bundle-Ether *bundle-id.vlan-id*
	Example:
	dot1q vlan *vlan-id*
	Example:
	ipv4 address *ipv4-address mask*
	Example:
	no shutdown
	Example:
	exit
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# interface Bundle-Ether 3.1`
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# ipv4 address 20.2.3.4/24`
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# no shutdown`
	Example:
		(Optional) Adds more sub-interfaces to the bundle.

	Command or Action	Purpose
	`exit`	

Step 14	Do one of the following:
	• **end**
	• or
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 15	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# end`	Exits interface configuration mode.
Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits XR configuration mode.
Step 17	**configure** **Example:** `RP/0/RP0/CPU0:router # configure`	Enters XR configuration mode.
Step 18	**interface** {**TenGigE**} interface-path-id **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0`	Enters interface configuration mode for the Ethernet interface you want to add to the Bundle. Enter the **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.

	Command or Action	Purpose
Step 19	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds an Ethernet interface to the bundle you configured in Step 2 through Step 13. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the interface to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** If you do not specify the **mode** keyword, the default mode is **on (**LACP is not run over the port).
Step 20	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.
Step 21	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The no shutdown command returns the link to an up or down state depending on the configuration and state of the link.
Step 22	—	Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.

Step 23	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router toXR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

	Command or Action	Purpose
Step 24	Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.	Brings up the other end of the link bundle.
Step 25	**show bundle Bundle-Ether** bundle-id **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3`	(Optional) Shows information about the specified Ethernet link bundle. The **show bundle Bundle-Ether** command displays information about the specified bundle. If your bundle has been configured properly and is carrying traffic, the State field in the **show bundle Bundle-Ether** command output shows the number "4," which means the specified VLAN bundle port is "distributing."
Step 26	**show vlan interface** **Example:** `RP/0/RP0/CPU0:router# show vlan interface`	Displays the current VLAN interface and status configuration.

Step 27	**show vlan trunks** [{**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 28	**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 the Default LACP Short Period Time Interval

This section describes how to configure the default short period time interval for sending and receiving LACP packets on a Gigabit Ethernet interface. This procedure also enables the LACP short period.

SUMMARY STEPS

1. **configure**
2. **interface GigabitEthernet** *interface-path*
3. **bundle id** *number* **mode active**
4. **lacp period short**
5. Do one of the following:
   - **end**
   -
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1`	Creates a Ten Gigabit Ethernet interface and enters interface configuration mode.
Step 3	**bundle id** *number* **mode active** **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active`	Specifies the bundle interface and puts the member interface in active mode.
Step 4	**lacp period short** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short`	Configures a short period time interval for the sending and receiving of LACP packets, using the default time period of 1000 milliseconds or 1 second.
Step 5	Do one of the following: • **end** • • **commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# end` **Example:** **Example:**	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 XR EXEC mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# commit`	• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Custom LACP Short Period Time Intervals

This section describes how to configure custom short period time intervals (less than 1000 milliseconds) for sending and receiving LACP packets on a Gigabit Ethernet interface.

Note You must always configure the *transmit* interval at both ends of the connection before you configure the *receive* interval at either end of the connection. Failure to configure the >*transmit* interval at both ends first results in route flapping (a route going up and down continuously). When you remove a custom LACP short period, you must do it in reverse order. You must remove the *receive* intervals first and then the *transmit* intervals.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links*
7. **exit**
8. **interface Bundle-Ether** *bundle-id.vlan-id*
9. **dot1q vlan** *vlan-id*
10. **ipv4 address** *ipv4-address mask*
11. **no shutdown**
12. **exit**
13. Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.
14. Do one of the following:
    - **end**
    -
    - **commit**
15. **exit**
16. **exit**

17. **show ethernet trunk bundle-ether** *instance*
18. **configure**
19. **interface** {**GigabitEthernet** | **TenGigE**} interface-path-id
20. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
21. **no shutdown**
22. Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.
23. Do one of the following:
    - **end**
    - or
    - **commit**
24. Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.
25. **show bundle Bundle-Ether** bundle-id [**reasons**]
26. **show ethernet trunk bundle-ether** *instance*

DETAILED STEPS

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

	Command or Action	Purpose
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.

	Command or Action	Purpose
Step 12	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	Exits subinterface configuration mode for the VLAN subinterface.
Step 13	Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.	(Optional) Adds more subinterfaces to the bundle.

Step 14	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router toEXEC mode.
		• Entering **no** exits the configuration session and returns the router toEXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 15	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	Exits interface configuration mode.
Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits XR configuration mode.
Step 17	**show ethernet trunk bundle-ether** *instance* **Example:** `RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5`	(Optional) Displays the interface configuration. The Ethernet bundle instance range is from 1 through 65535.
Step 18	**configure** **Example:**	Enters XR configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router # configure`	
Step 19	**interface** {**GigabitEthernet** | **TenGigE**} interface-path-id **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 1/0/0/0`	Enters the interface configuration mode for the Ethernet interface you want to add to the Bundle. Enter the **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the *rack/slot/module* format. **Note** A VLAN bundle is not active until you add an Ethernet interface on both ends of the link bundle.
Step 20	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds an Ethernet interface to the bundle you configured in Step 2 through Step 13. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the interface to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** If you do not specify the **mode** keyword, the default mode is **on (**LACP is not run over the port).
Step 21	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The no shutdown command returns the link to an up or down state depending on the configuration and state of the link.
Step 22	Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.	—

Step 23	Do one of the following:
	• **end**
	• or
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 24	Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.	Brings up the other end of the link bundle.
Step 25	**show bundle Bundle-Ether** bundle-id [**reasons**] **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3 reasons`	(Optional) Shows information about the specified Ethernet link bundle. The **show bundle Bundle-Ether** command displays information about the specified bundle. If your bundle has been configured properly and is carrying traffic, the State field in the **show bundle Bundle-Ether** command output will show the number "4," which means the specified VLAN bundle port is "distributing."
Step 26	**show ethernet trunk bundle-ether** *instance* **Example:** `RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5`	(Optional) Displays the interface configuration. The Ethernet bundle instance range is from 1 through 65535.

Configuration Examples for Link Bundling

This section contains the following examples:

Example: Configuring an Ethernet Link Bundle

This example shows how to join two ports to form an EtherChannel bundle running LACP:

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)   #interface Bundle-Ether 3
RP/0/RP0/CPU0:router(config-if)#ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 1

RP/0/RP0/CPU0:router(config -if )# bundle maximum-active links 1 hot -standby
RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)#interface TenGigE 0/3/0/0
RP/0/RP0/CPU0:router(config-if)#bundle id 3 mode active
RP/0/RP0/CPU0:router(config -if )# bundle port-priority 1
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config)# exit
RP/0/RP0/CPU0:router(config)#interface TenGigE 0/3/0/1
```

```
RP/0/RP0/CPU0:router(config -if )#bundle id 3 mode active
RP/0/RP0/CPU0:router(config-if)# bundle  port-priority 2
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# exit
```

Example: Configuring a VLAN Link Bundle

The following example shows how to create and bring up two VLANs on an Ethernet bundle:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1.1
RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RP0/CPU0:Router(config-subif)# ip addr 10.2.3.4/24
RP/0/RP0/CPU0:Router(config-subif)# no shutdown
RP/0/RP0/CPU0:Router(config-subif)# exit
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1.2
RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 20
RP/0/RP0/CPU0:Router(config-subif)# 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 EFP Load Balancing on an Ethernet Link Bundle

The following example shows how to configure all egressing traffic on the fixed members of a bundle to flow through the same physical member link automatically.

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto
RP/0/RP0/CPU0:router(config-subif)#
```

The following example shows how to configure all egressing traffic on the fixed members of a bundle to flow through a specified physical member link.

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1
RP/0/RP0/CPU0:router(config-subif)#
```

Examples: Configuring LACP Short Periods

The following example shows how to configure the LACP short period time interval to the default time of 1000 milliseconds (1 second):

```
config
```

```
interface TenGigE 0/1/0/1
bundle id 1 mode active
lacp period short
commit
```

The following example shows how to configure custom LACP short period transmit and receive intervals to *less than* the default of 1000 milliseconds (1 second):

Router A

```
config
interface TenGigE 0/1/0/1
bundle id 1 mode active
lacp period short
commit
```

Router B

```
config
interface TenGigE 0/1/0/1
bundle id 1 mode active
lacp period short
commit
```

Router A

```
config
interface TenGigE 0/1/0/1
lacp period short transmit 100
commit
```

Router B

```
config
interface TenGigE 0/1/0/1
lacp period short transmit 100
commit
```

Router A

```
config
interface TenGigE 0/1/0/1
lacp period short receive 100
commit
```

Router B

```
config
interface TenGigE 0/1/0/1
lacp period short receive 100
commit
```

Additional References

These sections provide references related to link bundle configuration.

Related Documents

Standards

Standards	Title
IEEE 802.3ad (incorporated as Annex 43 into 802.3-2002)	—

MIBs

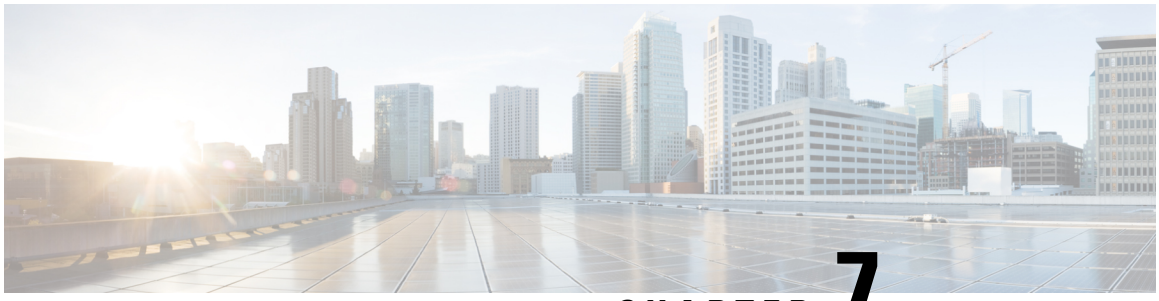
MIBs	MIBs Link
The IEEE-defined MIB for Link Aggregation (defined in 802.3 Annex 30C)	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 7

Configuring Traffic Mirroring

This module describes the configuration of the traffic mirroring feature. Traffic mirroring is sometimes called port mirroring, or switched port analyzer (SPAN).

- Overview of Traffic Mirroring, on page 95
- ERSPAN, on page 96
- ERPAN with UDF, on page 97
- Traffic Mirroring Terminology, on page 97
- Characteristics of the Source Port, on page 98
- Characteristics of the Monitor Session, on page 98
- Characteristics of the Destination Port, on page 98
- Configure Traffic Mirroring, on page 99

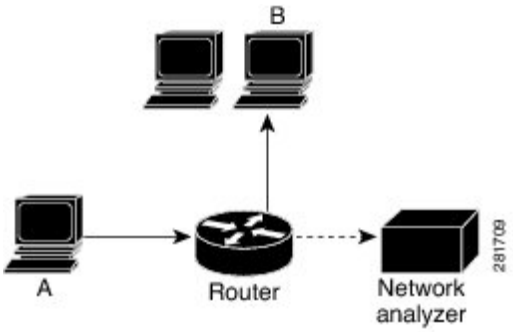
Overview of Traffic Mirroring

Traffic mirroring, which is sometimes called port mirroring, or Switched Port Analyzer (SPAN) is a Cisco proprietary feature that enables you to monitor network traffic passing in, or out of, a set of ports. You can then pass this traffic to a destination port on the same router.

Traffic mirroring copies traffic from one or more source ports 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 flow of traffic on the source interfaces or sub-interfaces, and allows the mirrored traffic to be sent to a destination interface or sub-interface.

For example, you need to attach a traffic analyzer to the router if you want to capture Ethernet traffic that is sent by host A to host B. All other ports see the traffic between hosts A and B.

Figure 1: Traffic Mirroring Operation



When local traffic mirroring is enabled, the traffic analyzer is attached directly to the port that is configured to receive a copy of every packet that host A sends. This port is called a traffic mirroring port. The other sections of this document describe how you can fine tune this feature.

ERSPAN

Encapsulated Remote Switched Port Analyzer (ERSPAN) mirrors traffic on one or more source ports and delivers the mirrored traffic to destination port on another switch or management server.

ERSPAN enables network operators to troubleshoot issues in the network in real-time using automated tools that auto-configures ERSPAN parameters on the network devices to send specific flows to management servers for in-depth analysis.

ERSPAN transports mirrored traffic over an IP network. The traffic is encapsulated at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination analyzer.

Supported Capabilities

The following capabilities are supported:

- Layer 3 interfaces, such as physical, and bundle interfaces or sub-interface, can be source interfaces.
- ERSPAN with GRE IPv4 has tunnel destinations.
- ERPSAN supports only RX direction
- One destination interface is allowed per monitor session.
- Only port mode or ACL permit packets are part of mirroring features.
- Full packet capture is supported.
- MPLS protocols are supported only with IPv4 unicast routing.
- To limit the amount of bandwidth used for SPAN, a static policer is applied before sending out the SPAN-replicated packet. There will be one policer for all the SPAN packets on RX source. Initially, the policing rate is set to 1Gbps per Network Processor Unit (NPU).

Restrictions

The following are the ERSPAN and SPAN ACL restrictions:

- The maximum number of user-defined fields (UDF) supported in configurations is 8.
- The maximum number of UDF configurations that can be add to access control entries (ACE) is 8.
- The maximum number of bytes involved in a UDF lookup is 16 bytes.
- Remove and re-apply monitor-sessions on all interfaces after modifying the access control list (ACL) and UDF
- Only port mode or ACL permit packets will be part of mirroring features.
- The UDF offset depth that can be configured is 64 bytes, beginning from the start of Layer 2 frame.
- GRE features do not support ERSPAN generic routing encapsulation (GRE) encapsulated packets.
- Tunnel statistics are updated in the ingress of ERSPAN packets. When these encapsulated packets are dropped in egress, the tunnel statistics is still updated.
- Only ERSPAN TYPE II header is supported. The value of the index and session-ID fields are always 0.
- Sequence bit is set in the GRE header and the value of sequence number is always 0 for ERSPAN packets

ERPAN with UDF

ERSPAN with UDF feature enables the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the ERSPAN destination. This feature helps you to analyze and isolate packet drops in the network.

Traffic Mirroring Terminology

- Ingress Traffic — Traffic that comes into the router.
- Egress Traffic — Traffic that goes out of the router.
- Source (SPAN) interface — An interface that is monitored using the SPAN feature.
- Monitor Session A designation for a collection of SPAN configurations consisting of many source interfaces and a set of destinations.
- Source port—A port that is monitored with the use of traffic mirroring. It is also called a monitored port.
- Destination port—A port that monitors source ports, usually where a network analyzer is connected. It is also called a monitoring port.
- Monitor session—A designation for a collection of traffic mirroring configurations consisting of a single destination and, potentially, many source interfaces.

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, 100-Gigabit Ethernet, or 10-Gigabit Ethernet.

Note Bridge group virtual interfaces (BVIs) are not supported.

- Each source port can be monitored in only one traffic mirroring session.
- It cannot be a destination port.
- Interfaces over which mirrored traffic may be routed must not be configured as a source port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor for local traffic mirroring. Remote traffic mirroring is supported in the ingress direction only. For bundles, the monitored direction applies to all physical ports in the group.

In the figure above, the network analyzer is attached to a port that is configured to receive a copy of every packet that host A sends. This port is called a traffic mirroring port.

Characteristics of the Monitor Session

A monitor session is a collection of traffic mirroring configurations consisting of a single destination and, potentially, many source interfaces. For any given monitor session, the traffic from the source interfaces (called *source ports*) is sent to the monitoring port or destination port. 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 port. The result is that the traffic that comes out of the destination port is a combination of the traffic from one or more source ports.

Monitor sessions have these characteristics:

- A single Cisco NCS 5500 Series Router can have a maximum of eight monitor sessions.
- A single monitor session can have only one destination port.
- A single destination port can belong to only one monitor session.
- 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 Port

Each session must have a destination port that receives a copy of the traffic from the source ports.

A destination port has these characteristics:

- A destination port must reside on the same router as the source port for local traffic mirroring. For remote mirroring the destination is always a GRE tunnel.
- A destination port for local mirroring can be any Ethernet physical port, EFP, and GRE tunnel interface, but not a bundle interface. It can be a Layer 2 or Layer 3 transport interface.
- A destination port can be a trunk (main) interface or a subinterface.
- At any one time, a destination port can participate in only one traffic mirroring session. A destination port in one traffic mirroring session cannot be a destination port for a second traffic mirroring session. In other words, no two monitor sessions can have the same destination port.
- A destination port cannot also be a source port.

Note
1. Source traffic mirroring ports (can be ingress or egress traffic ports).
2. Destination traffic mirroring port.

Configure Traffic Mirroring

```
 /* Configure remote  traffic mirroring. */

Router# configure
Router(config)# monitor-session session-name mpls-ipv4
Router(config)# destination interface tunnel-ip
Router(config)# exit
Router(config)# interface HundredGigE 0/1/0/1
Router(config-if)# monitor-session mon1 mpls-ipv4 direction rx-only
Router(config-if)# end

/* Attach the configurable source interface. */
Router# configure
Router(config)# interface HundredGigE 0/1/0/1
Router(config-if)# monitor-session mon1 mpls-ipv4 direction rx-only
Router(config-if-mon)# acl acl1
Router(config-if-mon)# end

/* Configure UDF-based ACL for traffic mirroring.*/
Router# configure
Router(config)# udf udf3 header outer 14 0 length
Router(config-if)# ipv4 access-list acl1
Router(config-ipv4-acl)#10 permit ipv4 any any udf udf1 0x1234 0xffff udf3 0x56 0xff
Router(config-ipv4-acl)# exit
Router(config)# interface HundredGigE 0/2/0/2
Router(config-if)# monitor-session mon1 mpls-ipv4 direction rx-only
Router(config-if-mon)# acl acl1
Router(config-if-mon)# commit
```

Running Configuration

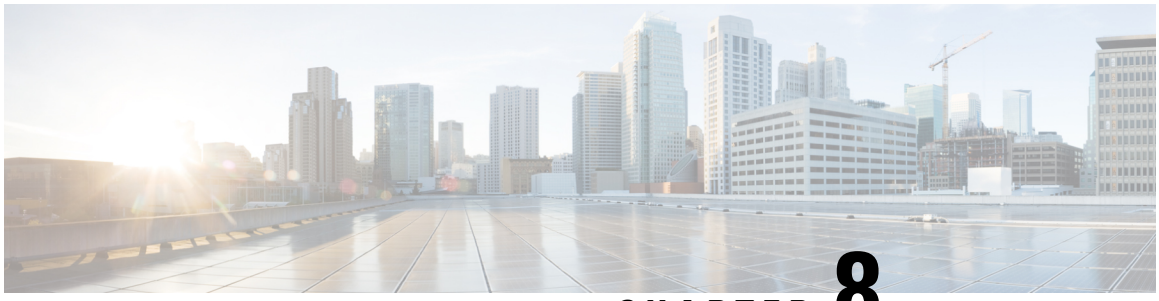
```
/* Configure remote  traffic mirroring. */

configure
monitor-session session-name mpls-ipv
 destination interface tunnel-ip
exit
interface HundredGigE 0/1/0/1
monitor-session mon1 mpls-ipv4 direction rx-only

!

/* Attach configurable source interface. */
 interface HundredGigE 0/1/0/1
  monitor-session mon1 mpls-ipv4 direction rx-only
  acl acl1
!

/* Configure UDF-based ACL for traffic mirroring. */
udf udf3 header outer 14 0 length
  ipv4 access-list acl1
  10 permit ipv4 any any udf udf1 0x1234 0xffff udf3 0x56 0xff
  exit
  interface HundredGigE 0/2/0/2
   monitor-session mon1 mpls-ipv4 direction rx-only
  acl acl1
!
```

Verification

```
/* The following output displays the statistics of traffic mirroring sessions. */
/* Note that all source interfaces and the replicated packet statistics for each interface.
  */

Router# show monitor-session counters

Sat May 20 06:09:11.505 UTC
Monitor-session test1 (MPLS-IPv4)
  TenGigE0/7/0/6/9.2
    Rx replicated: 56197 packets, 43440281 octets
    Tx replicated: 0 packets, 0 octets
    Non-replicated: 0 packets, 0 octets
  TenGigE0/7/0/6/9.3
    Rx replicated: 56134 packets, 43391582 octets
    Tx replicated: 0 packets, 0 octets
    Non-replicated: 0 packets, 0 octets
  TenGigE0/7/0/6/9.4
    Rx replicated: 56126 packets, 43385398 octets
    Tx replicated: 0 packets, 0 octets


/* The following output displays the configured traffic mirroring sessions. */
/* In this output, the list of source and destinations interfaces, their status, and other
 pertinent details are displayed. */

Router# show monitor-session status

Sat May 20 06:48:29.133 UTC
Monitor-session mon1 (MPLS-IPv4)
```

```
Destination interface tunnel-ip2
=========================================
Source Interface      Dir   Status
--------------------- ----  --------------
Te0/6/0/1/9           Rx    Operational
Te0/7/0/6/9.1         Rx    Operational
Te0/7/0/6/9.2         Rx    Operational
Te0/7/0/6/9.3         Rx    Operational
Te0/7/0/6/9.4         Rx    Operational
Te0/7/0/6/9.5         Rx    Operational
Te0/7/0/6/9.6         Rx    Operational
Te0/7/0/6/9.7         Rx    Operational
Te0/7/0/6/9.8         Rx    Operational
Te0/7/0/6/9.9         Rx    Operational

/* The following output diaplays the configured traffic mirroring sessions in detail for
the specofoed interface. */

Router#  show monitor-session mon1 status detail
Sat May 20 11:26:03.482 UTC
Monitor-session test3 (MPLS-IPv4)
  Destination interface tunnel-ip3
  Source Interfaces
  -----------------
  TenGigE0/7/0/6/9.200
    Direction:  Rx-only
    Port level: False
    ACL match:  Enabled (acl101200)
    Portion:    Full packet
    Interval:   Mirror all packets
    Status:     Operational
  TenGigE0/7/0/6/9.199
    Direction:  Rx-only
    Port level: False
    ACL match:  Enabled (acl101200)
    Portion:    Full packet
    Interval:   Mirror all packets
    Status:     Operational
  TenGigE0/7/0/6/9.198
    Direction:  Rx-only
    Port level: False
    ACL match:  Enabled (acl101200)



/* The following output displays the configured traffic mirroring sessions for the specifed
 interface. */

Router# show monitor-session source interface tenGigE 0/7/0/6/9 status internal
Sat May 20 06:13:52.934 UTC
Interface TenGigE0/7/0/6/9 (0x03800370)
SPAN MA:
  monitor-session test1 (MPLS-IPv4) (configured globally)
  destination interface tunnel-ip2 (0x08000084)
  replication direction: Rx-only
  port level: False
  ACL enabled (acl1)
  mirroring first 0 bytes
  interval: Mirror all packets
  state: up
  interface capsulation exists
  last PFI error: Success
SPAN EA, location 0/7/CPU0:
  monitor-session (MPLS-IPv4)
```

```
destination interface tunnel-ip2 (0x08000084)
replication direction: Rx-only
port level: False
ACL enabled (acl1)
mirroring first 0 bytes
interval: Mirror all packets
last platform error: Success.
```



CHAPTER 8

Configuring Virtual Loopback and Null Interfaces

This module describes the configuration of loopback and null interfaces on the Cisco NCS 6000 Series 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, Loopback 1, 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

Release	Modification
Release 5.0.0	This feature was introduced on the Cisco NCS 6000 Series Router.

- Prerequisites for Configuring Virtual Interfaces, on page 103
- Information About Configuring Virtual Interfaces, on page 103
- How to Configure Virtual Interfaces, on page 105
- Configuration Examples for Virtual Interfaces, on page 109

Prerequisites for Configuring Virtual Interfaces

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

Information About Configuring Virtual Interfaces

To configure virtual interfaces, you must understand the following concepts:

Virtual Loopback Interface Overview

A virtual loopback interface is a virtual interface with a single endpoint that is always up. Any packet transmitted over a virtual loopback interface is immediately received by the selfsame interface. Loopback interfaces emulate a physical interface.

In Cisco IOS XR software,virtual loopback interfaces perform the following functions:

- Loopback interfaces can act as a termination address for routing protocol sessions. This allows routing protocol sessions to stay up even if the outbound interface is down.
- You can ping the loopback interface to verify that the router IP stack is working properly.

In applications where other routers or access servers attempt to reach a virtual loopback interface, you must configure a routing protocol to distribute the subnet assigned to the loopback address.

Packets routed to the loopback interface are rerouted back to the router or access server and processed locally. IP packets routed out the loopback interface but not destined to the loopback interface are dropped. Under these two conditions, the loopback interface can behave like a null interface.

Null Interface Overview

A null interface functions similarly to the null devices available on most operating systems. This interface is always up and can never forward or receive traffic; encapsulation always fails. The null interface provides an alternative method of filtering traffic. You can avoid the overhead involved with using access lists by directing undesired network traffic to the null interface.

The Null 0 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 NCS 6000 Series 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

This task explains how to configure a basic loopback interface.

Note 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. Do one of the following:
   - **end**
   - **commit**
5. **show interfaces** *type interface-path-id*

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**interface loopback** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Loopback 3`	Enters interface configuration mode and names the new loopback interface.
Step 3	**ipv4 address** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38/32`	Assigns an IP address and subnet mask to the virtual loopback interface using the **ipv4 address** configuration command.

Step 4	Do one of the following:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	Example:
	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 XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 5	**show interfaces** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router# show interfaces Loopback 3`	(Optional) Displays the configuration of the loopback interface.

Configuring Null Interfaces

This task explains how to configure a basic null interface.

SUMMARY STEPS

1. **configure**

2. **interface null 0**
3. Do one of the following:
   - **end**
   -
   - **commit**
4. **show interfaces null 0**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface null 0** **Example:** `RP/0/RP0/CPU0:router#(config)# interface null 0`	Enters null0 interface configuration mode.

Step 3	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-null0)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-null0)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 4	**show interfaces null 0** **Example:** `RP/0/RP0/CPU0:router# show interfaces null0`	Verifies the configuration of the null interface.

Configuring Virtual IPv4 Interfaces

This task explains how to configure an IPv4 virtual interface.

SUMMARY STEPS

1. **configure**
2. **ipv4 address virtual address** *ipv4 address/mask*
3. Do one of the following:
   - **end**
   -
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**ipv4 address virtual address** *ipv4 address/mask* **Example:** `RP/0/RP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8`	Defines an IPv4 virtual address for the management Ethernet interface.

Step 3	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-null0)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-null0)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Virtual Interfaces

This section provides the following configuration examples:

Configuring a Loopback Interface: Example

The following example indicates how to configure a loopback interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Loopback 3
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38/32
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Loopback 3
Loopback3 is up, line protocol is up
  Hardware is Loopback interface(s)
  Internet address is 172.18.189.38/32
  MTU 1514 bytes, BW Unknown
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation Loopback,  loopback not set
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
```

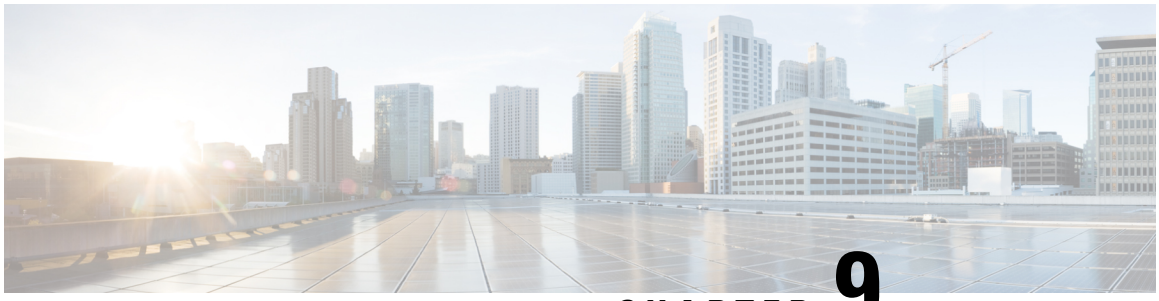
Configuring a Null Interface: Example

The following example indicates how to configure a null interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Null 0
RP/0/RP0/CPU0:router(config-null0)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Null 0
Null0 is up, line protocol is up
  Hardware is Null interface
  Internet address is Unknown
  MTU 1500 bytes, BW Unknown
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation Null,  loopback not set
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
```

Configuring a Virtual IPv4 Interface: Example

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8
RP/0/RP0/CPU0:router(config-null0)# commit
```



CHAPTER 9

Configuring 802.1Q VLAN Interfaces

This module describes the configuration and management of 802.1Q VLAN interfaces on the Cisco NCS 6000 Series 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

Release	Modification
Release 5.0.0	This feature was introduced.

- Prerequisites for Configuring 802.1Q VLAN Interfaces, on page 111
- Information About Configuring 802.1Q VLAN Interfaces, on page 112
- How to Configure 802.1Q VLAN Interfaces, on page 113
- , on page 118
- Configuration Examples for VLAN Interfaces, on page 119
- Additional References, on page 121

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 a 10-Gigabit Ethernet interface , a Fast Ethernet interface, or an Ethernet Bundle.

Information About Configuring 802.1Q VLAN Interfaces

To configure 802.1Q VLAN interfaces, you must understand the following concepts:

802.1Q VLAN Overview

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

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

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

Cisco IOS XR software supports VLAN subinterface configuration on 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.

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 Sub-interfaces 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 sub-interfaces can be added to a single Ethernet bundle.

The procedure for creating VLAN sub-interfaces on an Ethernet bundle is exactly the same as the procedure for creating VLAN sub-interfaces on a physical Ethernet interface. To create a VLAN subinterface on an Ethernet bundle, see the How to Configure 802.1Q VLAN Interfaces, on page 113 section later in this module.

How to Configure 802.1Q VLAN Interfaces

This section contains the following procedures:

Configuring 802.1Q VLAN Subinterfaces

This task explains how to configure 802.1Q VLAN sub-interfaces. To remove these sub-interfaces, see the Removing an 802.1Q VLAN Subinterface, on page 118 section of this module.

Configuring 802.1Q VLAN Subinterfaces

SUMMARY STEPS

1. **configure**
2. **interface** {**TenGigE** | **Bundle-Ether**} *interface-path-id.subinterface*
3. **encapsulation dot1q**
4. **ipv4 address** *ip-address mask*
5. **exit**
6. Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.
7. Do one of the following:
   - **end**
   -
   - **commit**
8. **show ethernet trunk bundle-ether** *instance*

DETAILED STEPS

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

Step 2	**interface** {**TenGigE** | **Bundle-Ether**} *interface-path-id.subinterface*
	Example:
	`RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10`
		Enters subinterface configuration mode and specifies the interface type, location, and subinterface number.
		• Replace the *interface-path-id* argument with one of the following instances:
		  • Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is *rack*/*slot*/*module*/*port*, and a slash between values is required as part of the notation.
		  • Ethernet bundle instance. Range is from 1 through 65535.
		• Replace the *subinterface* argument with the subinterface value. Range is from 0 through 4095.
		• Naming notation is *interface-path-id.subinterface*, and a period between arguments is required as part of the notation.

Step 3	**encapsulation dot1q** **Example:** `RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100`	Sets the Layer 2 encapsulation of an interface.

Step 4	**ipv4 address** *ip-address mask*
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24`
		Assigns an IP address and subnet mask to the subinterface.
		• Replace *ip-address* with the primary IPv4 address for an interface.
		• Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways:
		  • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.
		  • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.

	Command or Action	Purpose
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	(Optional) Exits the subinterface configuration mode. • The exit command is not explicitly required.
Step 6	Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.	—

Step 7	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	Example:
	Example:
	`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 XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR 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	**show ethernet trunk bundle-ether** *instance* **Example:** `RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5`	(Optional) Displays the interface configuration. The Ethernet bundle instance range is from 1 through 65535.

Configuring an Attachment Circuit on a VLAN

Use the following procedure to configure an attachment circuit on a VLAN.

SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE** | Bundle-Ether | **TenGigE**] *interface-path*] *id.subinterface* **l2transport**
3. **dot1q vlan** *vlan-id*
4. **l2protocol** {**cdp** | **pvst** | **stp** | **vtp**}{[**forward** | **tunnel**][**experimental** *bits*]|**drop**}
5. Do one of the following:

- **end**
-
- **commit**

6. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface*

DETAILED STEPS

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

Step 2	**interface** [**GigabitEthernet** | **TenGigE** | Bundle-Ether | **TenGigE**] *interface-path*] *id.subinterface* **l2transport**
	Example:
	`RP/0/RP0/CPU0:router(config)# interfaceTenGigE 0/1/0/0.1 l2transport`
		Enters subinterface configuration and specifies the interface type, location, and subinterface number.
		• Replace the *interface-path-id* argument with one of the following instances:
		  • Physical Ethernet interface instance,or Ethernet bundle instance. Naming notation is *rack*/*slot*/*module*/*port*, and a slash between values is required as part of the notation.
		  • Ethernet bundle instance. Range is from 1 through 65535.
		• Replace the *subinterface* argument with the subinterface value. Range is from 0 through 4095.
		• 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 that an AC.

Step 3	**dot1q vlan** *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: **dot1q vlan** vlan-id • To configure a Q-in-Q AC, use the following syntax: **dot1q vlan**

	Command or Action	Purpose
		vlan-id **vlan** vlan-id • To configure a Q-in-Any AC, use the following syntax: **dot1q vlan** vlan-id **vlan** **any**
Step 4	**l2protocol** {**cdp** | **pvst** | **stp** | **vtp**} {[**forward** | **tunnel**][**experimental** *bits*]|**drop**} **Example:** `RP/0/RP0/CPU0:router(config-if-l2)# l2protocol stp tunnel` **Example:**	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	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if-l2)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-if-l2)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 6	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface* **Example:** `RP/0/RP0/CPU0:router# show interfaces TenGigE 0/3/0/0.1`	(Optional) Displays statistics for interfaces on the router.

What to Do Next

- To configure a Point-to Point pseudo-wire cross connect on the AC, see the "*Implementing MPLS Layer 2 VPNs"* module of the *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 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, on page 113" section in this module.

SUMMARY STEPS

1. **configure**
2. **no interface** {**TenGigE** | **Bundle-Ether**] *interface-path-id.subinterface*
3. Repeat Step 2 to remove other VLAN subinterfaces.
4. Do one of the following:
   - **end**
   -
   - **commit**
5. **show ethernet trunk bundle-ether** *instance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**no interface** {**TenGigE** | **Bundle-Ether**] *interface-path-id.subinterface*	Removes the subinterface, which also automatically deletes all the configuration applied to the subinterface.

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10`	• Replace the *instance* argument with one of the following instances: • Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is *rack*/*slot*/*module*/*port*, and a slash between values is required as part of the notation. • Ethernet bundle instance. Range is from 1 through 65535. • Replace the *subinterface* argument with the subinterface value. Range is from 0 through 4095. Naming notation is *instance.subinterface*, and a period between arguments is required as part of the notation.
Step 3	Repeat Step 2 to remove other VLAN subinterfaces.	—

Step 4	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	Example:
	Example:
	`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 XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 5	**show ethernet trunk bundle-ether** *instance* **Example:** `RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5`	(Optional) Displays the interface configuration. The Ethernet bundle instance range is from 1 through 65535.

Configuration Examples for VLAN Interfaces

This section contains the following example:

VLAN Subinterfaces: Example

The following example shows how to create three VLAN subinterfaces at one time:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.1
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 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)# dot1q vlan 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)# dot1q vlan 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-subif)# exit
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 vlan interface
          -
Interface               Encapsulation    Outer 2nd   Service  MTU    LineP
                                         VLAN  VLAN                    State
Te0/6/0/0.1             802.1Q              1       L3        9604  up
Te0/6/0/4.1             802.1Q              1       L3        9604  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)#  dot1q vlan 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)# dot1q vlan 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 0/0/0/0.1
RP/0/RP0/CPU0:router(config-subif)# l2transport
RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20
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

The following sections provide references related to VLAN interface configuration.

Related Documents

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature	—

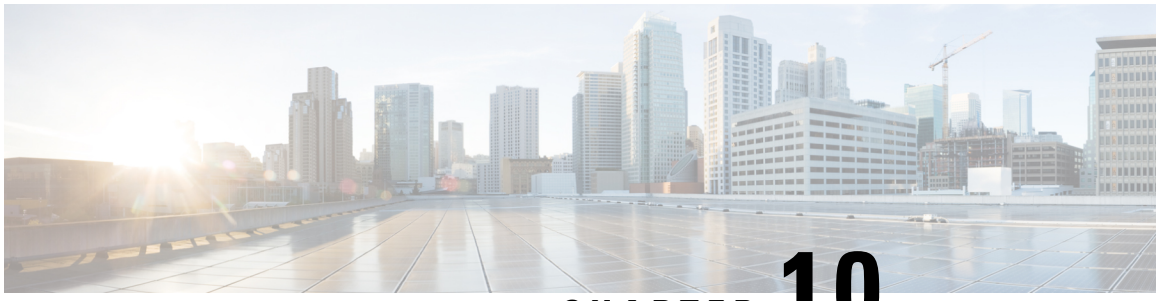
MIBs

MIBs	MIBs Link
There are no applicable MIBs for this module.	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL:http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 10

Configuring GRE Tunnels

Generic Routing Encapsulation (GRE) is a tunneling protocol that provides a simple generic approach to transport packets of one protocol over another protocol by means of encapsulation. This module provides information about how to configure a GRE tunnel.

- Configuring Static GRE, on page 123
- IP-in-IP De-capsulation, on page 124
- Single Pass GRE Encapsulation Allowing Line Rate Encapsulation, on page 128

Configuring Static GRE

Tunneling provides a mechanism to transport packets of one protocol within another protocol. Generic Routing Encapsulation (GRE) is a tunneling protocol that provides a simple generic approach to transport packets of one protocol over another protocol by means of encapsulation. GRE encapsulates a payload, that is, an inner packet that needs to be delivered to a destination network inside an outer IP packet. The GRE tunnel behave as virtual point-to-point link that have two endpoints identified by the tunnel source and tunnel destination address. The tunnel endpoints send payloads through GRE tunnels by routing encapsulated packets through intervening IP networks. Other IP routers along the way do not parse the payload (the inner packet); they only parse the outer IP packet as they forward it towards the GRE tunnel endpoint. Upon reaching the tunnel endpoint, GRE encapsulation is removed and the payload is forwarded to the packet's ultimate destination.

Encapsulation by the outer packet takes place at the tunnel source whereas decapsulation of the outer packet takes place at the tunnel destination. Encapsulation and decapsulation data is collected periodically or on demand. Encapsulation statistics provide us the number of packets encapsulated at the tunnel source. Decapsulation statistics provide us the number of packets that are decapsulated at the tunnel destination. This data is stored as statistics in logical tables that are based on statistics type in the route processor. The different statistics types include L2 Interface TX Stats, L3 Interface TX Stats, TRAP stats, and so on. Encapsulation statistics can help you to infer the source of the traffic, and decapsulation statistics provide you the destination of the traffic. Decapsulation statistics also help you to detect the type of traffic as well.

Restrictions for Configuring GRE Tunnels

The following restrictions apply while configuring GRE tunnels:

- NCS6000 Series Routers support up to 500 GRE tunnels.
- Only up to 16 unique source IP addresses are supported for the tunnel source.

Configuration Example

Configuring a GRE tunnel involves creating a tunnel interface and defining the tunnel source and destination. This example shows how to configure a GRE tunnel between Router1 and Router2. You need to configure tunnel interfaces on both the routers. Tunnel source IP address on Router1 will be configured as the tunnel destination IP address on Router2. Tunnel destination IP address on Router1 will be configured as the tunnel source IP address on Router2. In this example, OSPF is used as the routing protocol between the two routers. You can also configure BGP or IS-IS as the routing protocol.

```
RP/0/RP0/CPU0:Router1# configure
RP/0/RP0/CPU0:Router1(config)#  interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-if)# tunnel mode gre ipv4
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/CPU0:Router1(config-if)# ipv6 address 10:1:1::1/64
RP/0/RP0/CPU0:Router1(config-if)# tunnel source 192.168.1.1
RP/0/RP0/CPU0:Router1(config-if)# tunnel destination 192.168.2.1
RP/0/RP0/CPU0:Router1(config-if)# exit
RP/0/RP0/CPU0:Router1(config)# interface Loopback 0
RP/0/RP0/CPU0:Router1(config-if)# ipv4 address 1.1.1.1
RP/0/RP0/CPU0:Router1(config-if)# exit
RP/0/RP0/CPU0:Router1(config)# router ospf 1
RP/0/RP0/CPU0:Router1(config-ospf)# router-id 192.168.4.1
RP/0/RP0/CPU0:Router1(config-ospf)# area 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-ospf-ar)#  interface Loopback 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# commit

RP/0/RP0/CPU0:Router2# configure
RP/0/RP0/CPU0:Router2(config)#  interface tunnel-ip 30
RP/0/RP0/CPU0:Router2(config-if)# tunnel mode gre ipv4
RP/0/RP0/CPU0:Router2(config-if)# ipv4 address 10.1.1.2 255.255.255.0
RP/0/RP0/CPU0:Router2(config-if)# ipv6 address 10:1:1::2/64
RP/0/RP0/CPU0:Router2(config-if)# tunnel source 192.168.2.1
RP/0/RP0/CPU0:Router2(config-if)# tunnel destination 192.168.1.1
RP/0/RP0/CPU0:Router2(config-if)# exit
RP/0/RP0/CPU0:Router2(config)# interface Loopback 0
RP/0/RP0/CPU0:Router2(config-if)# ipv4 address 2.2.2.2
RP/0/RP0/CPU0:Router2(config)# router ospf 1
RP/0/RP0/CPU0:Router2(config-ospf)# router-id 192.168.3.1
RP/0/RP0/CPU0:Router2(config-ospf)# area 0
RP/0/RP0/CPU0:Router2(config-ospf-ar)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router2(config-ospf-ar)# interface Loopback 0
RP/0/RP0/CPU0:Router2(config-if)# commit
```

IP-in-IP De-capsulation

Encapsulation of datagrams in a network is done for multiple reasons, such as when a source server wants to influence the route that a packet takes to reach the destination host. The source server is also known as the encapsulation server.

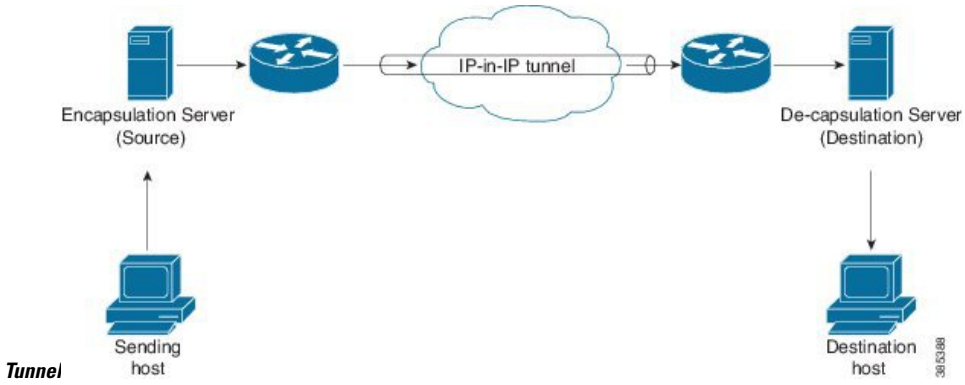
IP-in-IP encapsulation involves the insertion of an outer IP header over the existing IP header. The source and destination address in the outer IP header point to the end points of the IP-in-IP tunnel. The stack of IP headers are used to direct the packet over a predetermined path to the destination, provided the network adminstrator knows the loopback addresses of the routers transporting the packet. This tunneling mechanism can be used for determining availability and latency for most network architectures. It is to be noted that the entire path from source to the destination does not have to be included in the headers, but a segment of the network can be chosen for directing the packets.

Note Cisco NCS 6000 Series Router supports 512 IP-in-IP decapsulation tunnels.

Note IP-in-IP packets, when received on an interface, are recirculated on the same interface after the ucode removes the outer IP-in-IP header. So, the same packet is received twice on the same interface. The first time, the packet's outer IP-in-IP header is intact; and the second time the packet's outer IP-in-IP header is not available but the inner payload is available. Because of this, the show interface accounting command displays double statistics.

The following illustration describes the basic IP-in-IP encapsulation and decapsulation model.

Figure 2: Basic Encapsulation and De-capsulation with an IP-in-IP

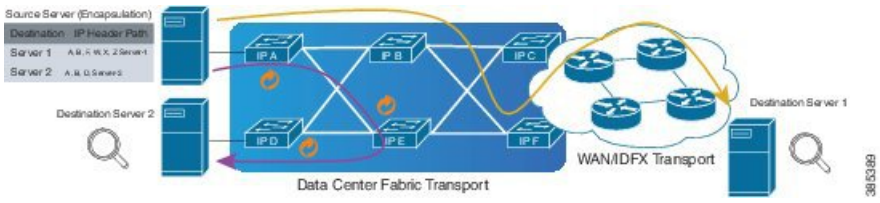


Tunnel

Use Case: Configure IP-in-IP de-capsulation

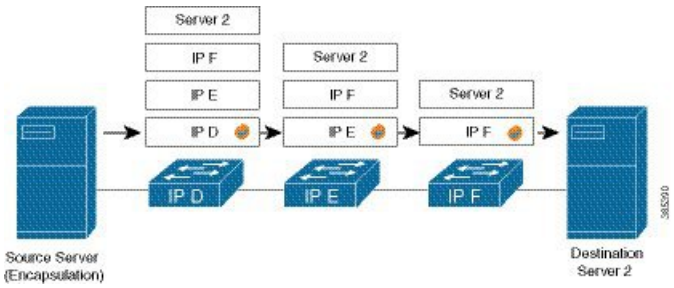
The following topology describes a use case where IP-in-IP encapsulation and de-capsulation is used for different segments of the network from source to destination. The IP-in-IP tunnel consists of multiple routers used to de-capsulate and direct the packet through the data center fabric network.

Figure 3: IP-in-IP De-capsulation through a Data Center Network



The following illustration shows how the stacked IPv4 headers are de-capsulated as they traverse through the de-capsulating routers.

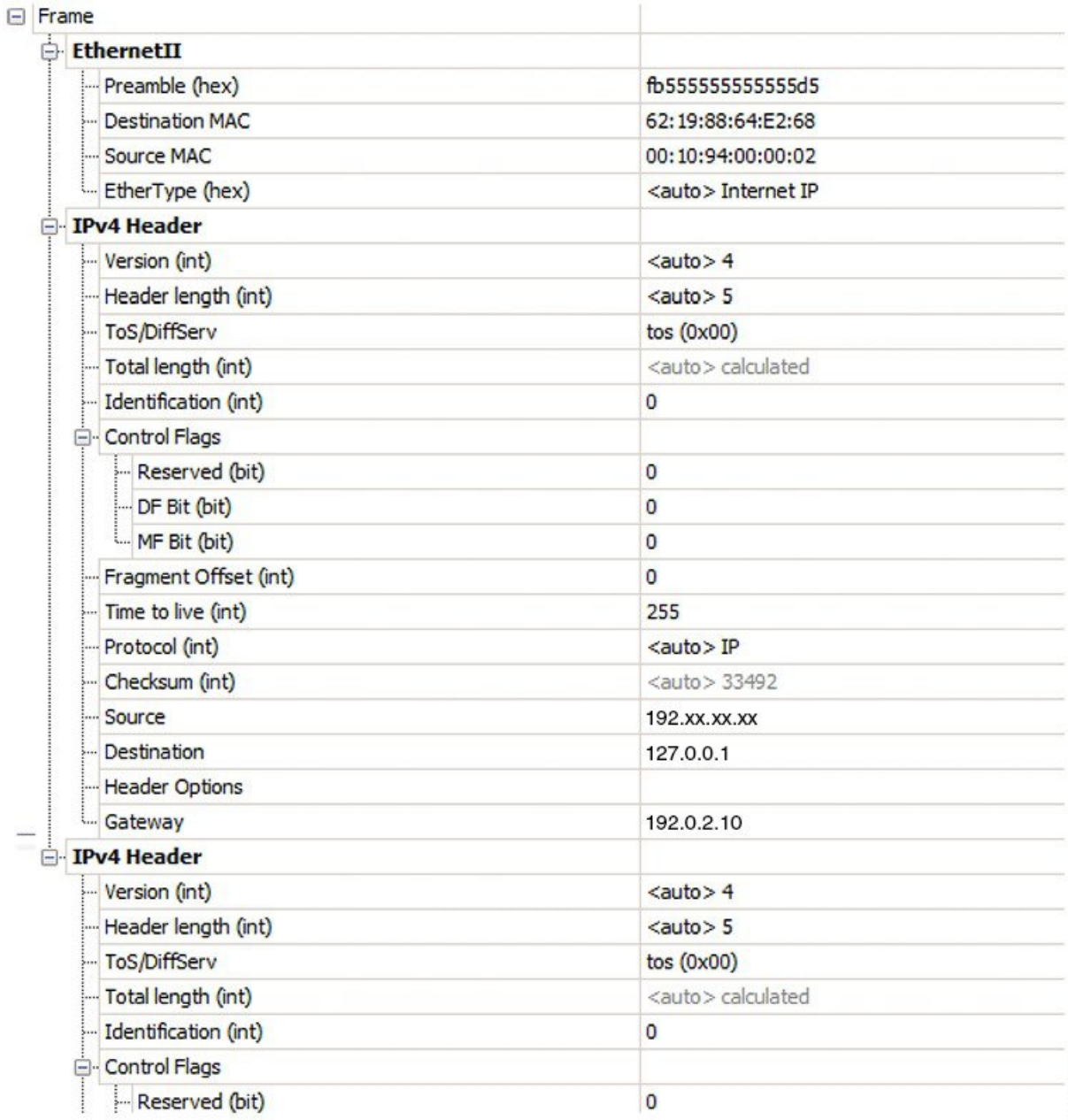
Figure 4: IP Header De-capsulation



Stacked IP Header in an Encapsulated Packet

The encapsulated packet will have an outer IPv4 header stacked over the original IPv4 header, as shown in the following illustration.

Encapsulated Packet



Configuration

You can use the following sample configuration on the routers to decapsulate the packet as it traverses the IP-in-IP tunnel:

```
RP/0/RP0/CPU0:router(config)# interface tunnel-ip 10
RP/0/RP0/CPU0:router(config-if)# tunnel mode ipv4 decap
RP/0/RP0/CPU0:router(config-if)# tunnel source loopback 0
RP/0/RP0/CPU0:router(config-if)# tunnel destination 10.10.1.2/32
```

- **tunnel-ip**: configures an IP-in-IP tunnel interface.

- **ipv4 unnumbered loopback address**: enables ipv4 packet processing without an explicit address, except for loopback address.
- **tunnel mode ipv4 decap**: enables IP-in-IP de-capsulation.
- **tunnel source**: indicates the source address for the IP-in-IP decap tunnel w.r.t the router interface.
- **tunnel destination**: indicates the destination address for the IP-in-IP decap tunnel w.r.t the router interface.

Running Configuration

```
RP/0/RP0/CPU0:router# show running-config interface tunnel-ip 10
...
interface tunnel-ip 10
tunnel mode ipv4 decap
tunnel source Loopback 0
tunnel destination 10.10.1.2/32
```

This completes the configuration of IP-in-IP de-capsulation.

Single Pass GRE Encapsulation Allowing Line Rate Encapsulation

Single Pass GRE Encapsulation Allowing Line Rate Encapsulation feature, also known as Prefix-based GRE Tunnel Destination for Load Balancing feature, enables line rate GRE encapsulation traffic and enables flow entropy. Data-plane forwarding performance supports full line rate, which is adjusted to consider added encapsulation. GRE tunnel goes down if the destination is not available in RIB. GRE tunnel destination address is an anycast address. All of the GRE encapsulation must be assigned based upon either an ACL or a policy-map, or both. Destinations may be individual addresses or /28 prefixes.

Configuration

Perform the following steps to configure the GRE Single-Pass Entropy feature:

- GRE Single-pass
- GRE Entropy(ECMP/UCMP)

```
/* GRE Single-Pass */

Router# configure
Router(config)# interface tunnel-ip30016
Router(config-if)# ipv4 address 216.1.1.1 255.255.255.0
Router(config-if)# ipv6 address 216:1:1::1/64
Router(config-if)# ipv6 enable
Router(config-if)# tunnel mode gre ipv4 encap
Router(config-if)# tunnel source Loopback22
Router(config-if)# tunnel destination 170.170.170.22
Router(config-if)# commit
Router(config-if)# exit

/* GRE Entropy(ECMP/UCMP)*/

ECMP (ISIS)
```

```
Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# commit
!

/* UCMP(ISIS) */

Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# ucmp
Router(config-isis-af)# metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# redistribute connected
Router(config-isis-af)# commit
Router(config-isis-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether3
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 20
Router(config-af)# commit
Router(config-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether111
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 15
Router(config-af)# commit
Router(config-af)# exit
!

/* ECMP(OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# address-family ipv4 unicast
```

```
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit


/* UCMP(OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# ucmp
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit


/* ECMP(BGP) */
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# network 170.170.170.3/32
Router(config-bgp-af)# network 170.170.170.10/32
Router(config-bgp-af)# network 170.170.170.11/32
Router(config-bgp-af)# network 170.170.172.3/32
Router(config-bgp-af)# network 180.180.180.9/32
Router(config-bgp-af)# network 180.180.180.20/32
Router(config-bgp-af)# network 180.180.180.21/32
Router(config-bgp-af)# network 180.180.180.24/32
```

```
Router(config-bgp-af)# network 180.180.180.25/32
Router(config-bgp-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 4.1.1.2
Router(config-bgp-nbr)# remote-as 300
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# commit
!

/* UCMP(BGP) */

Router# configure
Router(config)# router bgp 800
Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# maximum-paths ebgp 5
Router(config-bgp-af)# network 180.180.180.9/32
Router(config-bgp-af)# network 180.180.180.20/32
Router(config-bgp-af)# network 180.180.180.21/32
Router(config-bgp-af)# network 180.180.180.24/32
Router(config-bgp-af)# network 180.180.180.25/32
Router(config-bgp-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 7.1.5.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSIT0_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# 4.1.111.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSIT0_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!

/* Configure roupte policy */

Router# configure
Router(config)# route-policy TRANSIT0_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:1250000)
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
!
```

```
Router# configure
Router(config)# route-policy TRANSIT1_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:37500000
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
```

Running Configuration

```
/* GRE Single-Pass configuration */

interface tunnel-ip30016
ipv4 address 216.1.1.1 255.255.255.0
ipv6 address 216:1:1::1/64
ipv6 enable
tunnel mode gre ipv4 encap
tunnel source Loopback22
tunnel destination 170.170.170.22
!

/* GRE Entropy(ECMP/UCMP) */

ECMP (ISIS)

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
!

/* UCMP(ISIS) */

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
ucmp
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
redistribute connected
!
interface Bundle-Ether3
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 20
!
```

```
interface Bundle-Ether111
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 15
!

!

/* ECMP(OSPF) */

router ospf 3
nsr
maximum paths 5
address-family ipv4 unicast
area 0
interface Bundle-Ether3
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
!
interface Loopback23
!
interface hundredGigE0/9/0/0
!
!
!
/* UCMP (OSPF) */

router ospf 3
nsr
maximum paths 5
ucmp
address-family ipv4 unicast
area 0
interface Bundle-Ether3
cost 2
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
cost 2
!
interface Loopback23
!
interface hundredGigE0/9/0/0
!
!
!

/* ECMP(BGP)*/

router bgp 800
bgp bestpath as-path multipath-relax
address-family ipv4 unicast
maximum-paths ebgp 5
network 170.170.170.3/32
network 170.170.170.10/32
network 170.170.170.11/32
```

```
network 170.170.172.3/32
network 180.180.180.9/32
network 180.180.180.20/32
network 180.180.180.21/32
network 180.180.180.24/32
network 180.180.180.25/32
!
neighbor 4.1.1.2
remote-as 300
address-family ipv4 unicast
route-policy PASS-ALL in
route-policy PASS-ALL out
next-hop-self
!
!

/* UCMP(BGP) */

router bgp 800
bgp bestpath as-path multipath-relax
address-family ipv4 unicast
maximum-paths ebgp 5
network 180.180.180.9/32
network 180.180.180.20/32
network 180.180.180.21/32
network 180.180.180.24/32
network 180.180.180.25/32
!

neighbor 7.1.5.2
remote-as 4000
address-family ipv4 unicast
route-policy TRANSIT0_IN in
route-policy PASS-ALL out
next-hop-self
!
!
neighbor 4.1.111.2
remote-as 4000
address-family ipv4 unicast
route-policy TRANSIT1_IN in
route-policy PASS-ALL out
next-hop-self
!
!

/* Configure roupte policy */

route-policy TRANSIT0_IN
if destination in (170.170.170.24/32) then
set extcommunity bandwidth (2906:1250000)
else
pass
endif
end-policy
!
route-policy TRANSIT1_IN
if destination in (170.170.170.24/32) then
set extcommunity bandwidth (2906:37500000)
else
pass
endif
end-policy
!
```

Verification

Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show interfaces tunnel-ip 100

Sun Jul 10 15:49:04.812 VN_TIME

tunnel-ip100 is up, line protocol is up

  Interface state transitions: 2

  Hardware is Tunnel

  Internet address is Unknown

  MTU 1500 bytes, BW 100 Kbit (Max: 100 Kbit)

     reliability 255/255, txload 0/255, rxload 0/255

  Encapsulation TUNNEL_GRE,  loopback not set,

  Tunnel TOS 0

  Tunnel mode GRE IPV4,

  Keepalive is enabled, interval 10 seconds, maximum retry 3

  Tunnel source 172.16.16.1 (GigabitEthernet0_0_0_0), destination 172.16.16.2

  Tunnel TTL 100

  Last input 2d03h, output 2d04h

  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

     689 packets input, 26212 bytes, 0 total input drops

     0 drops for unrecognized upper-level protocol

     Received 0 broadcast packets, 0 multicast packets

     3 packets output, 192 bytes, 0 total output drops

     Output 0 broadcast packets, 0 multicast packets
```

Verify if the tunnel mode GRE encapsulation and decapsulation are enabled.

```
Router# sh interfaces tunnel-ip 5 accounting
Wed May 16 01:50:57.258 UTC
tunnel-ip5
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  IPV4_UNICAST              489            55746            0                0
  IPV6_UNICAST              489            55746            0                0
  MPLS                      587            69266            0                0
```

Verify if the recycle of the packets are not done under Recycle VoQ: 48:

```
 Router# show tunnel ip ea summary location 0/7/CPU0

Number of tunnel updates to retry: 0
```

```
Number of tunnel updates retried: 0
Number of tunnel retries failed: 0
Platform:
Recycle VoQ: 48
          ReceivedBytes   ReceivedPackets    ReceivedKbps
          DroppedBytes    DroppedPackets     DroppedKbps

NPU 0:0       0               0                   0
              0               0                   0
1             0               0                   0
              0               0                   0
2             0               0                   0
              0               0                   0
3             0               0                   0
              0               0                   0
...
NPU 1:0       0               0                   0
              0               0                   0
1             0               0                   0
              0               0                   0
2             0               0                   0
              0               0                   0
3             0               0                   0
              0               0                   0

NPU 2:0       0               0                   0
              0               0                   0
1             0               0                   0
              0               0                   0
2             0               0                   0
              0               0                   0
3             0               0                   0
              0               0                   0
```

Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show interfaces tunnel-ip * brief

Thu Sep 7 00:04:39.125 PDT
Intf Intf LineP Encap MTU BW
Name     State    State   Type       (byte) (Kbps)
--------------------------------------------------------------------------------
ti30001  down     down    TUNNEL_IP   1500   100
ti30002  up       up      TUNNEL_IP   1500   100
```

Verify the tunnel endpoint route in RIB.

```
Router# show route 10.1.1.1

Routing entry for 10.0.0.0/8
Known via "static", distance 1, metric 0 (connected)
Installed Oct 2 15:50:56.755 for 00:39:24
Routing Descriptor Blocks
directly connected, via tunnel-ip109
Route metric is 0, Wt is 1
No advertising protos.
```

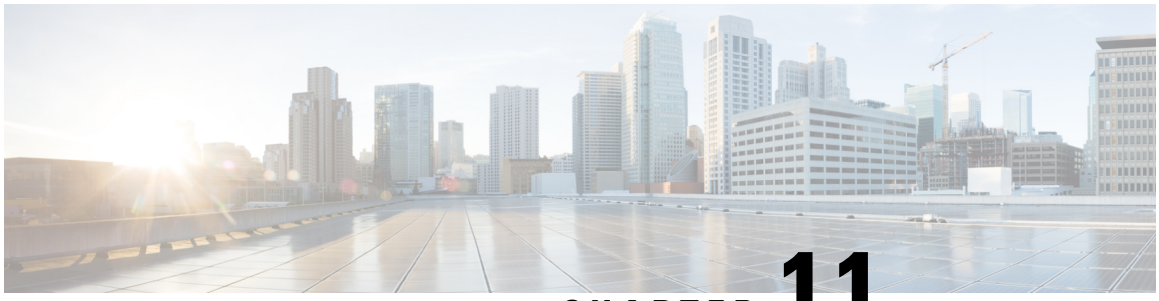
Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show tunnel ip ea database tunnel-ip 109 location 0/7/CPU0

----- node0_0_CPU0 -----
tunnel ifhandle 0x80022cc
tunnel source 161.115.1.2
tunnel destination 162.1.1.1/32
tunnel transport vrf table id 0xe0000000
```

```
tunnel mode gre ipv4, encap
tunnel bandwidth 100 kbps
tunnel platform id 0x0
tunnel flags 0x40003400
IntfStateUp
BcStateUp
Ipv4Caps
Encap
tunnel mtu 1500
tunnel tos 0
tunnel ttl 255
tunnel adjacency flags 0x1
tunnel o/p interface handle 0x0
tunnel key 0x0, entropy length 0 (mask 0xffffffff)
tunnel QT next 0x0
tunnel platform data (nil)
Platform:
Handle: (nil)
Decap ID: 0
Decap RIF: 0
Decap Recycle Encap ID: 0x00000000
Encap RIF: 0
Encap Recycle Encap ID: 0x00000000
Encap IPv4 Encap ID: 0x4001381b
Encap IPv6 Encap ID: 0x00000000
Encap MPLS Encap ID: 0x00000000
DecFEC DecRcyLIF DecStatsId EncRcyLIF
```

Verify if the QoS table is updated properly.

```
Router# show controllers npu stats voq base 48 instance all location
0/RP0/CPU0
Asic Instance = 0
VOQ Base = 48
       ReceivedPkts    ReceivedBytes    DroppedPkts    DroppedBytes
-------------------------------------------------------------------
COS0 =   0              0                0              0
COS1 =   0              0                0              0
COS2 =   0              0                0              0
COS3 =   0              0                0              0

Asic Instance = 1
VOQ Base = 48
       ReceivedPkts    ReceivedBytes    DroppedPkts    DroppedBytes
-------------------------------------------------------------------
COS0 =   0              0                0              0
COS1 =   0              0                0              0
COS2 =   0              0                0              0
COS3 =   0              0                0              0

Asic Instance = 2
VOQ Base = 48
       ReceivedPkts    ReceivedBytes    DroppedPkts    DroppedBytes
-------------------------------------------------------------------
COS0 =   0              0                0              0
COS1 =   0              0                0              0
COS2 =   0              0                0              0
COS3 =   0              0                0              0
```



CHAPTER 11

Configuring Tunnel Interfaces

This module describes the configuration of Tunnel-IPSec interfaces on the Cisco NCS 6000 Series 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

Release	Modification
Release 5.0.0	This feature was introduced.

- Prerequisites for Configuring Tunnel Interfaces, on page 140
- Information About Configuring Tunnel Interfaces, on page 140
- How to Configure Tunnel Interfaces, on page 141
- Configuration Examples for Tunnel Interfaces, on page 143
- Where to Go Next, on page 144

Prerequisites for Configuring Tunnel Interfaces

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

Information About Configuring Tunnel Interfaces

To implement tunnel interfaces, you must understand the following concepts:

Tunnel Interfaces Overview

Tunneling provides a way to encapsulate arbitrary packets inside of a transport protocol. This feature is implemented as a virtual interface to provide a simple interface for configuration. The tunnel interfaces are not tied to specific "passenger" or "transport" protocols, but, rather, they represent an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. Because supported tunnels are point-to-point links, you must configure a separate tunnel for each link.

There are three necessary steps in configuring a tunnel interface:

1. Specify the tunnel interface—**interface tunnel-ipsec** *identifier*
2. Configure the tunnel source—**tunnel source** {*ip-address* | *interface-id* }
3. Configure the tunnel destination—**tunnel destination** {*ip-address* | *tunnel-id* }

Virtual Interface Naming Convention

Virtual interface names never use the physical interface naming notation *rack/slot/module/port* for identifying an interface's rack, slot, module, and port, because they are not tied to any physical interface or subinterface.

Virtual interfaces use a globally unique numerical identifier (per virtual interface type).

Examples of naming notation for virtual interfaces:

```
Interface                      IP-Address      Status  Protocol
Loopback0                      10.9.0.0         Up           Up
Loopback10                     10.7.0.0         Up           Up
Tunnel-TE5000                  172.18.189.38    Down         Down
Null10                         10.8.0.0          Up           Up
```

Tunnel-IPSec Overview

IPSec (IP security) is a framework of open standards for ensuring secure private communications over the Internet. It can be used to support Virtual Private Network (VPN), firewalls, and other applications that must transfer data across a public or insecure network. The router IPSec protocol suite provides a set of standards that are used to provide privacy, integrity, and authentication service at the IP layer. The IPSec protocol suite also includes cryptographic techniques to support the key management requirements of the network-layer security.

When IPSec is used, there is no need to use Secure Shell (SSH) or Secure Socket Layer (SSL). Their use causes the same data to be encrypted or decrypted twice, which creates unnecessary overhead. The IPSec daemon is running on both the RPs and the DRPs. IPSec is an optional feature on the router. IPSec is a good choice for a user who has multiple applications that require secure transport. On the client side, customers can use “Cisco VPN 3000 Client” or any other third-party IPSec client software to build IPSec VPN.

Note IPSec tunnel exists in the control plane, so you do not have to bring up or bring down the tunnel. Entry into the IPSec tunnel is only for locally sourced traffic from the RP or DRP, and is dictated by the access control lists (ACL) configured as a part of the profile that is applied to the Tunnel-IPSec.

Tunnel-IPSec Naming Convention

A profile is entered from interface configuration submode for interface tunnel-ipsec. For example:

```
interface tunnel-ipsec 30
     profile <profile name>
```

Crypto Profile Sets

Crypto profile sets must be configured and applied to tunnel interfaces (or to the crypto IPSec transport). For IPSec to succeed between two IPSec peers, the crypto profile entries of both peers must contain compatible configuration statements.

Two peers that try to establish a security association must each have at least one crypto profile entry that is compatible with one of the other peer's crypto profile entries. For two crypto profile entries to be compatible, they must at least meet the following criteria:

- They must contain compatible crypto access lists. In the case where the responding peer is using dynamic crypto profiles, the entries in the local crypto access list must be “permitted” by the peer's crypto access list.
- They must each identify the other peer (unless the responding peer is using dynamic crypto profiles).
- They must have at least one transform set in common.

Note Crypto profiles cannot be shared; that is, the same profile cannot be attached to multiple interfaces.

How to Configure Tunnel Interfaces

This section contains the following procedures:

Configuring Tunnel-IPSec Interfaces

This task explains how to configure Tunnel-IPSec interfaces.

Before you begin

To use the profile command, you must be in a user group associated with a task group that includes the proper task IDs for crypto commands. To use the **tunnel destination** command, you must be in a user group associated with a task group that includes the proper task IDs for interface commands.

For detailed information about user groups and task IDs, see the *Configuring AAA Services* module of *System Security Configuration Guide for the Cisco NCS 6000 Series Routers* 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* module in *System Security Configuration Guide for Cisco NCS 6000 Series Routers* .

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. Do one of the following:
   - **end**
   -
   - **commit**
7. **show ip route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR configuration mode.
Step 2	**interface tunnel-ipsec** *identifier* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-ipsec 30`	Identifies the IPSec interface to which the crypto profile will be attached and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# profile user1`	Assigns the crypto profile name to be applied to the tunnel for IPSec processing. • The same crypto profile cannot be shared in different IPSec modes.
Step 4	**tunnel source** {*ip-address* | *interface-id* } **Example:** `RP/0/RP0/CPU0:router(config-if)# tunnel source Ethernet0/1/1/2`	Specifies the tunnel source IP address or interface ID. • This command is required for both static and dynamic profiles.
Step 5	**tunnel destination** {*ip-address* | *tunnel-id* } **Example:** `RP/0/RP0/CPU0:router(config-if)# tunnel destination 192.168.164.19`	(Optional) Specifies the tunnel destination IP address. • This command is not required if the profile is dynamic.

Step 6	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	Example:
	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 XR EXEC mode.
		• Entering **no** exits the configuration session and returns the router to XR 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 ip route** **Example:** `RP/0/RP0/CPU0:router# show ip route`	Displays forwarding information for the tunnel. • The command show ip route displays what was advertised and shows the routes for static and autoroute.

Configuration Examples for Tunnel Interfaces

This section contains the following example:

Tunnel-IPSec: Example

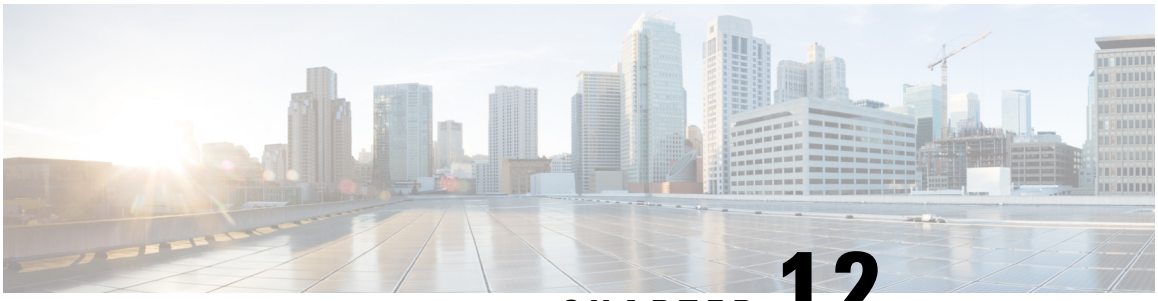
This example shows the process of creating and applying a profile to an IPSec tunnel. The necessary preliminary steps are also shown. You must first define a transform set and then create a profile before configuring the IPSec tunnel.

```
RP/0/RP0/CPU0:router# configureRP/0/RP0/CPU0:router(config)# crypto ipsec transform-set
tset1
RP/0/RP0/CPU0:router(configtransform-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# configure
RP/0/RP0/CPU0:router(config)# crypto ipsec profile user1
RP/0/RP0/CPU0:router(config-user1)# match sampleac1 transform-set tset1
RP/0/RP0/CPU0:router(config-user1)# set pfs group5
RP/0/RP0/CPU0:router(config-user1)# set type dynamic
RP/0/RP0/CPU0:router(config-user1)# exit


RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-ipsec 30
RP/0/RP0/CPU0:router(config-if)# profile user1
RP/0/RP0/CPU0:router(config-if)# tunnel source MgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:router(config-if)# tunnel destination 192.168.164.19
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
```

Where to Go Next

You now must apply a crypto profile to each transport. Applying the crypto profile set to a transport instructs the router to evaluate all the interface's traffic against the crypto profile set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

For information on applying a crypto profile to each transport, see the *Implementing IPSec Network Security on* module of *System Security Configuration Guide* .



CHAPTER 12

Configuring Dense Wavelength Division Multiplexing Controllers

This module describes the configuration of dense wavelength division multiplexing (DWDM) controllers.

DWDM is an optical technology that is used to increase bandwidth over existing fiber-optic backbones. DWDM can be configured on supported 10-Gigabit Ethernet (GE) or Packet-over-SONET/SDH physical layer interface modules (PLIMs). After you configure the DWDM controller, you can configure an associated 10-Gigabit Ethernet interface.

Feature History for Configuring DWDM Controller Interfaces

Release	Modification
Release 5.2.3	Support for OTN Termination was included.

- Prerequisites for Configuring DWDM Controller Interfaces, on page 145
- Information About the DWDM Controllers, on page 146
- Information about IPoDWDM, on page 146
- How to Configure DWDM Controllers, on page 147
- Forward Error Correction (FEC), on page 151
- Configuring IPoDWDM , on page 153
- Configuration Examples, on page 156
- Additional References, on page 157

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.

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.

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 (OCh) OH is added to form the OCh. The OCh provides the OTN management functionality and contains four subparts: the OPU, and frame alignment signal (FAS).

Figure 5: OTN Optical Channel Structure

These are the conditions for OTU Port configuration:

- OTU4 can be configured at slice level only.
- Slice reset occurs immediately after commit.
- Interface is removed from the slice.
- Slice is powered back up in OTU4 mode.
- Two 100 GigE interfaces are created.

Information about IPoDWDM

Cisco IOS XR software includes the IP over Dense Wavelength Division Multiplexing (IPoDWDM) feature.

IPoDWDM currently provides the following software features:

- Shared Risk Link Group (SRLG)

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:

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 task:

Note All interface configuration tasks for Gigabit Ethernet interfaces still must be performed in interface configuration mode.

Configuring the Optical Parameters

This task describes how to configure the wavelength parameters for the DWDM controller to set the operational wavelength of a tunable SFP+ module. The DWDM controllers are configured in the physical layer control element of the Cisco IOS XR software configuration space.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **admin-state out-of-service**
4. **commit**
5. **wavelength** *channel-number*
6. **commit**
7. **admin-state in-service**
8. Do one of the following:
   - **end**
   -
   - **commit**
9. **show controllers dwdm** *interface-path-id* **optics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enter the XR Config mode.
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller dwdm 0/1/0/0`	Specifies the DWDM controller name in the notation *rack/slot/module/port* and enters DWDM configuration mode.

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router(config-dwdm)#`	
Step 3	**admin-state out-of-service** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# admin-state out-of-service`	Specifies the DWDM interface administrative state. You must put the controller in **out-of-service** state before you can use the DWDM configuration commands.
Step 4	**commit** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# commit`	Saves configuration changes. This performs the shutdown from the previous step. After the controller has been shut down, you can proceed with the wavelength configuration.
Step 5	**wavelength** *channel-number* **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# wavelength 1`	Configures the channel number corresponding to the first wavelength. Values can range from 1 to 96. Use the **show controller dwdm** command with the **wavelength-map** keyword to determine which channels and wavelengths are supported on a specific controller.
Step 6	**commit** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# commit`	Saves configuration changes.
Step 7	**admin-state in-service** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# admin-state in-service`	Places the DWDM port in In-Service (IS) state, to support all normal operation.

Step 8	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-dwdm)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-dwdm)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 9	**show controllers dwdm** *interface-path-id* **optics** **Example:** `RP/0/RP0/CPU0:router# show controller dwdm 0/1/0/0 optics`	Displays the output power level, input power level, and wavelength information.

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.

Before you begin

The **g709 disable**, **loopback**, and **g709 fec** commands can be used only when the controller is in the shutdown state. Use the **admin-state** command.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. Do one of the following:
   - **admin-state maintenance**
   -
   - admin-state out-of-service
4. **commit**
5. **g709 disable**
6. **g709 fec** {**disable** | **standard**}
7. **g709 report** *alarm* **disable**
8. Do one of the following:
   - **end**
   -
   - **commit**
9. **admin-state in-service**
10. **show controllers dwdm** *interface-path-id* **g709**

DETAILED STEPS

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller dwdm 0/1/0/0`	Specifies the DWDM controller name in the notation *rack/slot/module/port* and enters DWDM configuration mode.
Step 3	Do one of the following: • **admin-state maintenance** • • admin-state out-of-service **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# admin-state out-of-service`	Disables the DWDM controller. You must disable the controller before you can use the DWDM configuration commands.
Step 4	**commit** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# commit`	Saves configuration changes. This performs the shutdown from the previous step. When the controller has been shut down, you can proceed with the configuration.
Step 5	**g709 disable** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# g709 disable`	(Optional) Disables the G.709 wrapper. The wrapper is enabled by default. **Note** The **g709 disable** command is available on the Cisco 4-Port 10-Gigabit Ethernet DWDM PLIM only.
Step 6	**g709 fec** {**disable** | **standard**} **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# g709 fec disable`	(Optional) Configures the forward error correction mode (FEC) for the DWDM controller. By default, enhanced FEC is enabled.
Step 7	**g709 report** *alarm* **disable** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# g709 odu bdi disable`	(Optional) Disables the logging of selected optical channel alarms to the console for a DWDM controller. By default, all alarms are logged to the console.
Step 8	Do one of the following: • **end** • • **commit** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# 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]:`

	Command or Action	Purpose
	Example: **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# 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.
Step 9	**admin-state in-service** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# admin-state in-service`	Places the DWDM port in In Service (IS) state, to support all normal operation.
Step 10	**show controllers dwdm** *interface-path-id* **g709** **Example:** `RP/0/RP0/CPU0:router# show controller dwdm 0/1/0/0 optics`	Displays the G.709 Optical Transport Network (OTN) protocol alarms and counters for Bit Errors, along with the FEC statistics and threshold-based alerts.

What to Do Next

All interface configuration tasks for the Gigabit Ethernet interfaces still must be performed in interface configuration mode. Refer to the corresponding modules in this book for more information.

Forward Error Correction (FEC)

FEC feature allows you to enable or disable RS-FEC (Reed-Solomon Codec for Ethernet IEEE 802.3 Clause 91) for 100G optics. The 100G optics can operate in both FEC and non-FEC mode. In FEC mode, these optics can cover an extended range. In non-FEC mode, 100G optics can interoperate with other optics.

FEC is a mechanism to recover lost packets on a link by sending extra “parity” packets for every group (N) of packets. As long as the receiver receives a subset of packets in the group (N-1) and the parity packet, up to a single lost packet in the group can be recovered.

FEC is supported on these optics:

- QSFP-100G-ER4L-S
- CPAK-100G-ER4F

Configuring FEC on an Interface

By default the interfaces operate in non-FEC mode. To enable FEC on an interface, use the **fec standard** command in the interface configuration mode.

Note The standard FEC refers to RS-FEC. The RS-FEC stands for Reed Solomon Forward Error Correction.

The RS-FEC is only supported on these 100G optic interfaces:

- QSFP-100G-ER4L-S
- CPAK-100G-ER4F

```
RP/0/RP0/CPU0:ios#(config)# interface HundredGigE 0/6/0/8
RP/0/RP0/CPU0:ios#(config-if)# fec standard
RP/0/RP0/CPU0:ios#(config)# commit
```

Verification

Use the **show controllers** command to verify the FEC configuration:

```
RP/0/RP0/CPU0:ios# show controllers hundredGigE 0/6/0/8

Thu Jul 25 05:58:38.162 UTC
 Operational data for interface HundredGigE0/6/0/8:

 State:
 Administrative state: enabled
 Operational state: Down (Reason: Link loss or low light, no loopback)
 LED state: Yellow On

 Phy:
 Media type: fiber over 4 lane optics (25km reach) (lite)
 Optics:
 Vendor: CISCO
 Part number: CISxxxx/101
 Serial number: xxxxxx
 Digital Optical Monitoring:
 Transceiver Temp: 45.996 C

 Alarms key: (H) Alarm high, (h) Warning high
 (L) Alarm low, (l) Warning low
 Wavelength Tx Power Rx Power Laser Bias
 Lane (nm) (dBm) (mW) (dBm) (mW) (mA)
 -- ----- ------ ------ ------ ------ ------
 1 0 0.7 1.1628 0.5 1.1238 218.560
 2 0 0.6 1.1368 0.5 1.1169 218.560
 3 0 0.6 1.1389 0.7 1.1790 218.560
 4 0 0.6 1.1497 0.8 1.1948 218.560

 DOM alarms:
 Not available
 Statistics:
 FEC:
 Corrected Codeword Count: 0
 Uncorrected Codeword Count: 0

 MAC address information:
 Operational address: e41f.0fae.7865
```

```
Burnt-in address: e41f.0fae.7865
No unicast addresses in filter
Operating in multicast promiscuous mode

Autonegotiation disabled.

Operational values:
Speed: 100Gbps
Duplex: Full Duplex
Flowcontrol: None
Loopback: None (or external)
MTU: 1514
MRU: 1514
Inter-packet gap: standard (12)
BER monitoring:
Signal Degrade: 1e-255 (no-alarm)
Signal Fail: 1e-0 (report-alarm)
Forward error correction: Standard (Reed-Solomon)
```

The show command confirms that RS-FEC is enabled on the HundredGigE0/6/0/8 interface.

Configuring IPoDWDM

This section provides the following configuration procedures:

Configuring the Optical Layer DWDM Ports

Use the following procedure to configure the Optical Layer DWDM ports.

SUMMARY STEPS

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **network port id** *id-number*
4. **network connection id** *id-number*
5. Do one of the following:
   - **end**
   -
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# config`	Enters global configuration mode.
Step 2	**controller dwdm** *interface-path-id* **Example:**	Specifies the DWDM controller and enters DWDM controller mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# controller dwdm 0/1/0/1`	
Step 3	**network port id** *id-number* **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# network port id 1/0/1/1`	Assigns an identifier number to a port for the Multi Service Transport Protocol (MSTP).
Step 4	**network connection id** *id-number* **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# network connection id 1/1/1/1`	Configures a connection identifier for the Multi Service Transport Protocol (MSTP).

Step 5	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-dwdm)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-dwdm)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Administrative State of DWDM Optical Ports

Use 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. Do one of the following:

- **interface pos** *interface-path-id*
-
- **interface tengige** *interface-path-id*

6. **maintenance disable**
7. Do one of the following:
   - **end**
   -
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# config`	Enters global configuration mode.
Step 2	**controller dwdm** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# controller dwdm 0/1/0/1`	Specifies the DWDM controller and enters DWDM controller mode.
Step 3	**admin-state** {**in-service** | **maintenance** | **out-of-service**} **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# admin-state maintenance`	Specifies the transport administration state.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-dwdm)# exit`	Exits to the previous mode.
Step 5	Do one of the following: • **interface pos** *interface-path-id* • • **interface tengige** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface pos 1/0/1/1` **Example:** **Example:**	Specifies the interface and enters interface configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# interface tengige 1/0/1/1`	
Step 6	**maintenance disable** **Example:** `RP/0/RP0/CPU0:router(config-if)# maintenance disable`	Provisions the maintenance embargo flag, which prevents maintenance activities from being performed on an interface.

Step 7	Do one of the following:
	• **end**
	•
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-dwdm)# end`
	Example:
	Example:
	`RP/0/RP0/CPU0:router(config-dwdm)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples

This section includes the following examples:

Turning On the Laser: Example

Note This is a required configuration. 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# configure

RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1

RP/0/RP0/CPU0:Router(config-dwdm)# admin-state in-service
RP/0/RP0/CPU0:Router(config-dwdm)# commit
```

Turning Off the Laser: Example

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
```

IPoDWDM Configuration: Examples

This section includes the following examples:

Optical Layer DWDM Port Configuration: Examples

The following example shows how to configure 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 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:

```
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
```

Additional References

These sections provide references related to DWDM controller configuration.

Related Documents

Related Topic	Document Title
Cisco IOS XR interface configuration commands	*Cisco IOS XR Interface and Hardware Component Command Reference*

Related Topic	Document Title
Initial system bootup and configuration information for a router using Cisco IOS XR software	*Cisco IOS XR Getting Started Guide*
Cisco IOS XR AAA services configuration information	*Cisco IOS XR System Security Configuration Guide* and*Cisco IOS XR System Security Command Reference*

Standards

Standards	Title
ITU-T G.709/Y.1331	Interfaces for the optical transport network (OTN)

MIBs

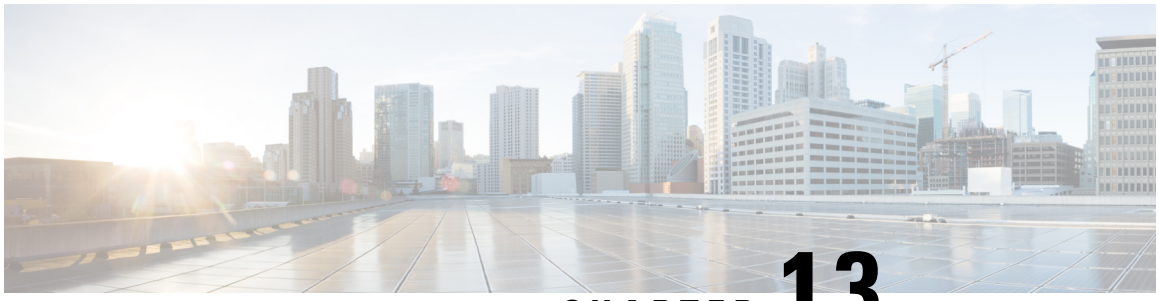
MIBs	MIBs Link
—	To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
OTN-MIB	IPoDWDM MIB

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
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.	http://www.cisco.com/support



CHAPTER 13

Configuring IP-in-IP Decapsulation

This module describes how to configure IP-in-IP Decapsulation.

- IP-in-IP Decapsulation, on page 159

IP-in-IP Decapsulation

Encapsulation of datagrams in a network is done for multiple reasons, such as when a source server wants to influence the route that a packet takes to reach the destination host. The source server is also known as the encapsulation server.

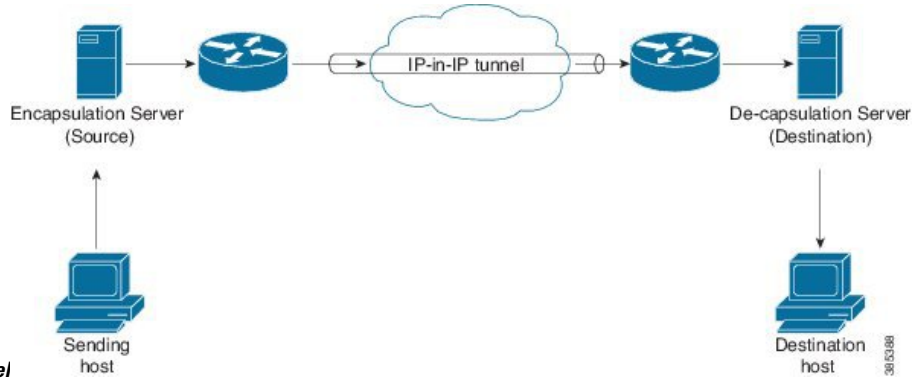
IP-in-IP encapsulation involves the insertion of an outer IP header over the existing IP header. The source and destination address in the outer IP header point to the end points of the IP-in-IP tunnel. The stack of IP headers are used to direct the packet over a predetermined path to the destination, provided the network adminstrator knows the loopback addresses of the routers transporting the packet. This tunneling mechanism can be used for determining availability and latency for most network architectures. It is to be noted that the entire path from source to the destination does not have to be included in the headers, but a segment of the network can be chosen for directing the packets.

Note Cisco NCS 6000 Series Router supports 512 IP-in-IP decapsulation tunnels.

Note IP-in-IP packets, when received on an interface, are recirculated on the same interface after the ucode removes the outer IP-in-IP header. So, the same packet is received twice on the same interface. The first time, the packet's outer IP-in-IP header is intact; and the second time the packet's outer IP-in-IP header is not available but the inner payload is available. Because of this, the show interface accounting command displays double statistics.

The following illustration describes the basic IP-in-IP encapsulation and decapsulation model.

Figure 6: Basic Encapsulation and De-capsulation with an IP-in-IP

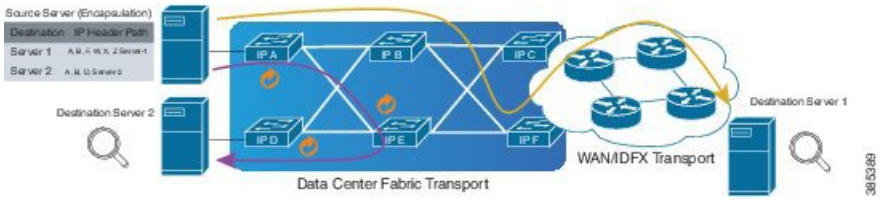


Tunnel

Use Case: Configure IP-in-IP de-capsulation

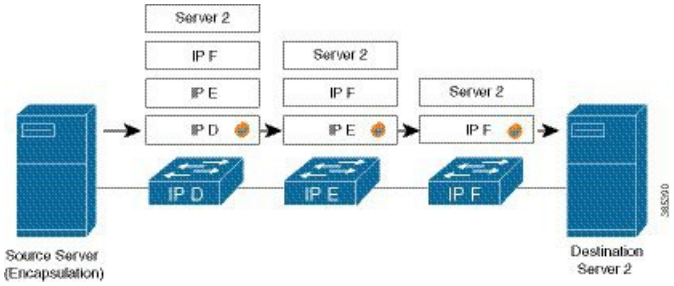
The following topology describes a use case where IP-in-IP encapsulation and de-capsulation is used for different segments of the network from source to destination. The IP-in-IP tunnel consists of multiple routers used to de-capsulate and direct the packet through the data center fabric network.

Figure 7: IP-in-IP De-capsulation through a Data Center Network



The following illustration shows how the stacked IPv4 headers are de-capsulated as they traverse through the de-capsulating routers.

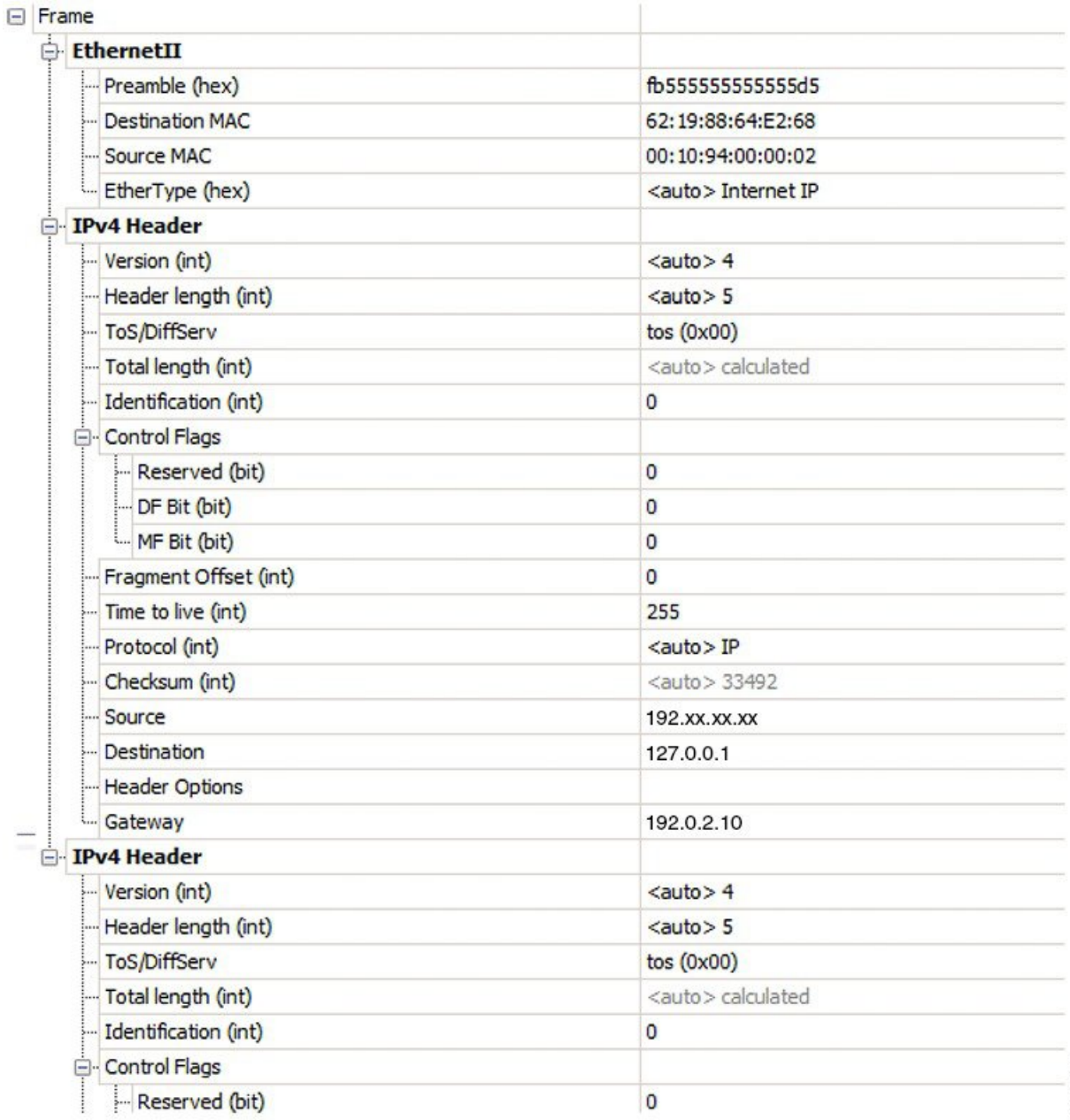
Figure 8: IP Header De-capsulation



Stacked IP Header in an Encapsulated Packet

The encapsulated packet will have an outer IPv4 header stacked over the original IPv4 header, as shown in the following illustration.

Encapsulated Packet



Configuration

You can use the following sample configuration on the routers to decapsulate the packet as it traverses the IP-in-IP tunnel:

```
RP/0/RP0/CPU0:router(config)# interface tunnel-ip 10
RP/0/RP0/CPU0:router(config-if)# tunnel mode ipv4 decap
RP/0/RP0/CPU0:router(config-if)# tunnel source loopback 0
RP/0/RP0/CPU0:router(config-if)# tunnel destination 10.10.1.2/32
```

- **tunnel-ip**: configures an IP-in-IP tunnel interface.

- **ipv4 unnumbered loopback address**: enables ipv4 packet processing without an explicit address, except for loopback address.
- **tunnel mode ipv4 decap**: enables IP-in-IP de-capsulation.
- **tunnel source**: indicates the source address for the IP-in-IP decap tunnel w.r.t the router interface.
- **tunnel destination**: indicates the destination address for the IP-in-IP decap tunnel w.r.t the router interface.

Running Configuration

```
RP/0/RP0/CPU0:router# show running-config interface tunnel-ip 10
...
interface tunnel-ip 10
tunnel mode ipv4 decap
tunnel source Loopback 0
tunnel destination 10.10.1.2/32
```

This completes the configuration of IP-in-IP de-capsulation.