Interface and Hardware Component Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 7.11.x

First Published: 2023-12-01

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

CONTENTS

Preface

This guide describes the interface and hardware component configuration details for Cisco IOS XR software. This chapter contains details on the changes made to this document.

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

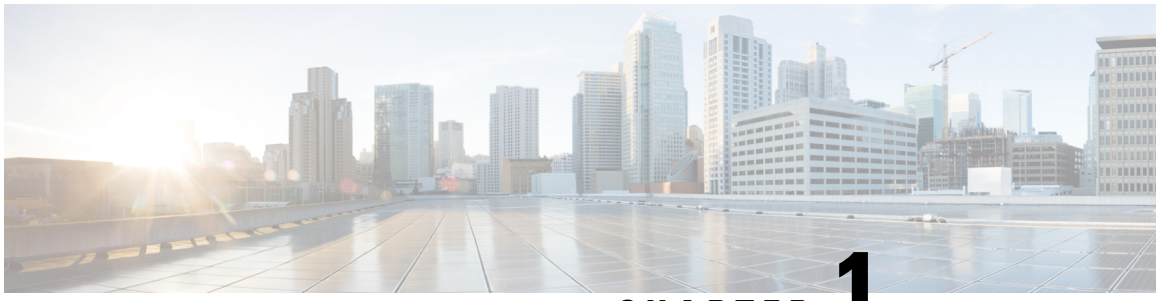
Date	Summary
December 2023	Initial release of this document

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

New and Changed Feature Information

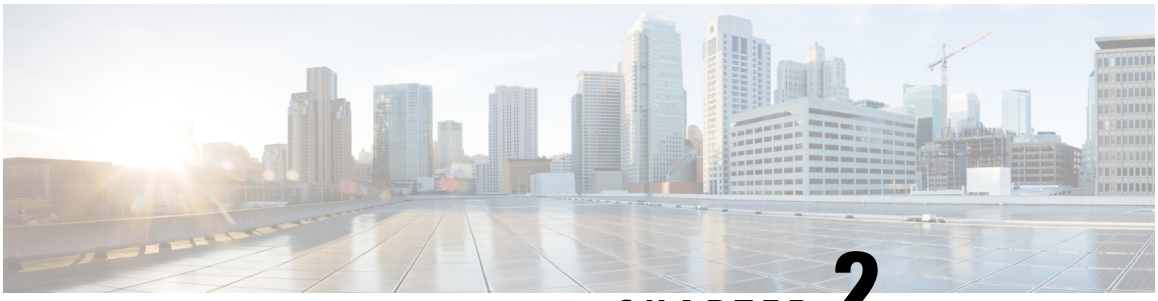
This table summarizes the new and changed feature information for the *Interfaces Configuration Guide for Cisco 8000 Series Routers* for Cisco 8000 Series Routers, and tells you where they are documented.

Interface and Hardware Component Features Added or Modified in IOS XR Release 7.11.x

This table summarizes the new and changed feature information for the *Interfaces Configuration Guide for Cisco 8000 Series Routers* for Cisco 8000 Series Routers, and tells you where they are documented.

Table 2: New and Changed Features

Feature	Description	Introduced in Release	Where Documented
Outer IP Header-Driven Hash Computation for Incoming GUE Packets	This feature was introduced.	Release 7.11.1	Configuring Generic UDP Encapsulation, on page 239
Loopback on Optical Transceivers	This feature was introduced.	Release 7.11.1	Loopback on Optical Transceivers, on page 307
Disable Auto-Squelching	This feature was introduced.	Release 7.11.1	Disable Auto-Squelching
On-demand transfer of Redundant Power Modules to Power Reservation Pool	This feature is introduced.	Release 7.11.1	On-demand transfer of Redundant Power Modules to Power Reservation Pool , on page 332
Ability to Set Maximum Power Limit the for Router	This feature is introduced.	Release 7.11.1	Ability to Set Maximum Power Limit for the Router , on page 337



CHAPTER 2

YANG Data Models for Interfaces and Hardware Component Features

This chapter provides information about the YANG data models for Interface and Hardware Component features.

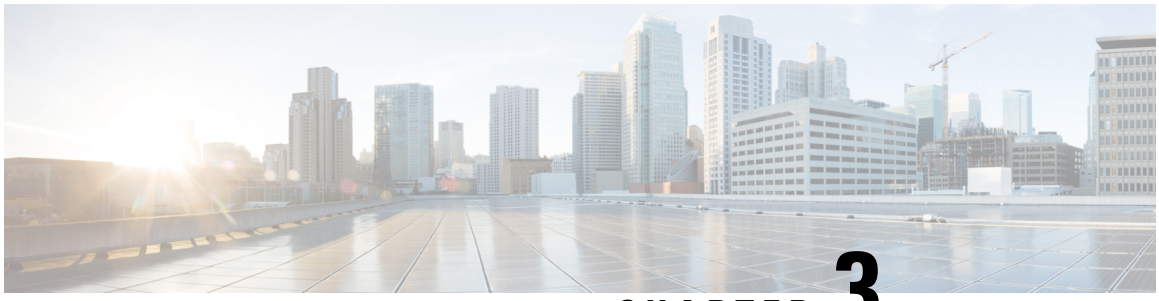
Using YANG Data Models

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the Github repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPaths. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository.

You can also view the data model definitions using the YANG Data Models Navigator tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.



CHAPTER 3

Preconfiguring Physical Interfaces

This module describes the preconfiguration of physical interfaces.

The system supports preconfiguration for the following interfaces:

- 10-Gigabit Ethernet
- 40-Gigabit Ethernet
- 100-Gigabit Ethernet
- 400-Gigabit Ethernet
- Management Ethernet

Preconfiguration allows you to configure line cards before you insert them into the router. When you insert the cards, they are instantly configured. The system creates the preconfiguration information in a different system database tree, rather than with the regularly configured interfaces. That database tree is known as the *preconfiguration directory* on the Route Processor.

There might be some preconfiguration data that you cannot verify unless the line card is present. This is because the verifiers themselves run only on the line card. You can verify such preconfiguration data when you insert the line card and initiate the verifiers. The system rejects a configuration if errors are found when you copy the configuration from the preconfiguration area to the active area.

Note Gigabit Ethernet interface is not supported. You can only preconfigure physical interfaces.

Prerequisites for Preconfiguring Physical Interfaces

Before preconfiguring physical interfaces, ensure that you meet the following condition(s):

- Preconfiguration drivers and files are installed. Although it might 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. You cannot verify or apply preconfigured interfaces until you insert the actual interface into the router with the matching location. The location can be the rack, slot, or module. When you insert the anticipated line card and create the interface, the system verifies the precreated configuration information. If the verification is successful, the system immediately applies the running configuration of the router.

Note When you plug the anticipated line card in, ensure that you verify any preconfiguration by using the appropriate **show** commands.

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

Note We recommend filling out preconfiguration information in your site planning guide. This allows you to compare the anticipated configuration with the actual preconfigured interfaces when you install the card 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 the 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, you can instantly configure the new modular services card that actively runs during the line card bootup.

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

Use of the Interface Preconfigure Command

To preconfigure the interfaces that are not yet present in the system, use the **interface preconfigure** command in global configuration mode.

The **interface preconfigure** command places the router in interface configuration mode. You must be able to add any possible interface commands. The verifiers registered for the preconfigured interfaces verify the

configuration. The preconfiguration is complete when you enter the **end** command, or any matching exit or global configuration mode command.

Note It is possible that you are not able to verify some configurations until you insert the line card is inserted.

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

You must provide names during preconfiguration that matches with the name of the interface that is created. If the interface names do not match, the system does not apply preconfiguration 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 Hu0/3/0/0) is not permitted.

Active and Standby RPs and Virtual Interface Configuration

The standby RP is available and is in a state in which it can take the load from the an active RP, if required. Following are the conditions when a standby RP becomes an active RP:

- 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, the system may insert a second RP. The second RP then 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 failover, the virtual interfaces become available on the standby (now active) RP. Their state and configuration is unchanged, and there is no loss of forwarding (in the case of tunnels) over the interfaces during the failover. The routers use nonstop forwarding (NSF) over tunnels through the failover of the host RP.

Note You do not need to configure anything to guarantee that the standby interface configurations are maintained.

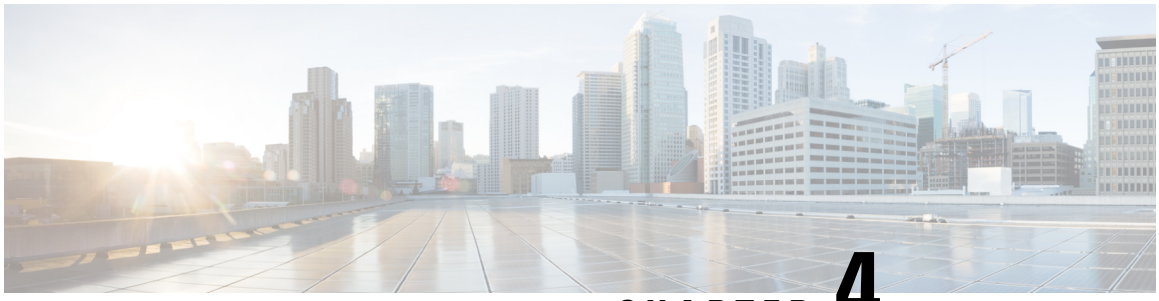
How to Preconfigure Physical Interfaces

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

```
/* Enter global configuration mode. */
 RP/0/RP0/CPU0:router:router:hostname# configure
```

```
/* 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. */

RP/0/RP0/CPU0:router:router(config)# interface preconfigure HundredGigE 0/3/0/2

/* Assign an IP address and mask to the interface. Use one of the following commands:
- ipv4 address ip-address subnet-mask
- ipv4 address ip-address/prefix */
RP/0/RP0/CPU0:router(config-if-pre)# ipv4 address 192.168.1.2/31
RP/0/RP0/CPU0:router(config-if-pre)# end
or
RP/0/RP0/CPU0:router(config-if-pre)# commit
RP/0/RP0/CPU0:router# show running-config
```

- When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit best-effort** command to save the configuration changes to the running configuration file and remain within the configuration session. The **commit best-effort** command merges the target configuration with the running configuration and commits only valid changes (best effort). Some configuration changes might fail due to semantic errors.



CHAPTER 4

Advanced Configuration and Modification of the Management Ethernet Interface

This module describes the configuration of Management Ethernet interfaces.

Before you use Telnet to access the router through the LAN IP address, you must set up a Management Ethernet interface and enable the Telnet servers.

Note By default, the Management Ethernet interfaces are present on the system. However, you must configure these interfaces to:

- Access the router.
- Use protocols and applications, such as Simple Network Management Protocol (SNMP), HTTP, eXtensible Markup Language (XML), TFTP, Telnet, and Command-Line Interface (CLI.)

Prerequisites for Configuring Management Ethernet Interfaces

Before you perform the Management Ethernet interface configuration procedures that are described in this chapter, ensure that you meet the following tasks and conditions:

- 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, ensure that both the active and standby Management Ethernet interfaces are 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(s):

Default Interface Settings

This table describes the default Management Ethernet interface settings that you can change with manual configuration. The system does not display the default settings in the **show running-config** command output.

Table 3: Management Ethernet Interface Default Settings

Parameter	Default Value	Configuration File Entry
Speed in Mbps	Default speed is **1G** with autonegotiated.	Speed in non-configurable.
Duplex mode	Default duplex mode is **full-duplex** with autonegotiated.	Duplex mode is non-configurable.
MAC address	MAC address is read from the hardware burned-in address (BIA).	MAC address is non-configurable.

How to Perform Advanced Management Ethernet Interface Configuration

This section contains the following procedures:

Configure a Management Ethernet Interface

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

Note The maximum MTU value for the management interface MgmtEth0/RP0/CPU0/0 is 9678 bytes.

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

/* Enter interface configuration mode and specify the Ethernet interface name and notation
 rack/slot/module/port. */
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0

RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.76.18.150/16 (or)
ipv4 address 1.76.18.150 255.255.0.0
```

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. You can specify the network mask in either of the two ways:
- The network mask can be a four-part dotted decimal address. For example, 255.255.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.
- The system indicates the network mask as a slash (/) and number. For example, /16 indicates that the first 16 bits of the mask are ones, and the corresponding bits of the address are the network address.

```
RP/0/RP0/CPU0:router(config-if)# mtu 1488
```

Note (Optional) The maximum transmission unit (MTU) value for the management interface is 9678 bytes.

- The default is 1514 bytes.
- The range for the Management Ethernet interface Interface **mtu** values is from 64 through 9678 bytes.

```
/* Remove the shutdown configuration, which removes the forced administrative down on the
interface, enabling it to move to an up or down state.  */
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
or
RP/0/RP0/CPU0:router(config-if)# commit
```

Saves configuration changes.

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

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

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

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

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 1.76.18.150/16
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# commit
```

```
RP/0/RP0/CPU0:router:Mar 26 01:09:28.685 :ifmgr[190]:%LINK-3-UPDOWN :Interface
MgmtEth0/RP0/CPU0/0, changed state to Up
RP/0/RP0/CPU0:router(config-if)# end

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

MgmtEth0/RP0/CPU0/0 is up, line protocol is up
  Interface state transitions: 3
  Hardware is Management Ethernet, address is 1005.cad8.4354 (bia 1005.cad8.4354)
  Internet address is 1.76.18.150/16
  MTU 1488 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, 1000BASE-T, link type is autonegotiation
  loopback not set,
  Last link flapped 00:00:59
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:00:02
  Last clearing of "show interface" counters never
  5 minute input rate 4000 bits/sec, 3 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     21826 packets input, 4987886 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 12450 broadcast packets, 8800 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1192 packets output, 217483 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
     3 carrier transitions

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

interface MgmtEth0/RP0/CPU0/0
 mtu 1488
 ipv4 address 1.76.18.150/16
 ipv6 address 2002::14c:125a/64
 ipv6 enable
!
```

The following example displays VRF configuration and verification of the Management Ethernet interface on the RP with the source address:

```
RP/0/RP0/CPU0:router# show run interface MgmtEth 0/RP0/CPU0/0
interface MgmtEth0/RP0/CPU0/0
 vrf httpupload
 ipv4 address 10.8.67.20 255.255.0.0
 ipv6 address 2001:10:8:67::20/48
!

RP/0/RP0/CPU0:router# show run http
Wed Jan 30 14:58:53.458 UTC
http client vrf httpupload
http client source-interface ipv4 MgmtEth0/RP0/CPU0/0

RP/0/RP0/CPU0:router# show run vrf
Wed Jan 30 14:59:00.014 UTC
vrf httpupload
!
```

Verify Management Ethernet Interface Configuration

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

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

Configuration Examples for Management Ethernet Interfaces

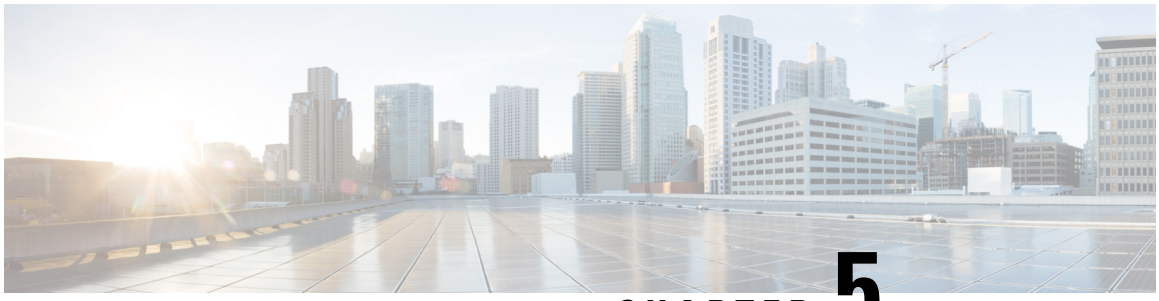
This section provides the following configuration examples:

Configuring a Management Ethernet Interface: Example

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

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

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

MMgmtEth0//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
     0 input errors, 0 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//CPU0:router# show running-config interface MgmtEth 0/

interface MgmtEth0/RP0/CPU0/0
 description Connected to Lab LAN
```

```
 ipv4 address 172.29.52.70 255.255.255.0
!
```



CHAPTER 5

Configuring Ethernet Interfaces

This module describes the configuration of Ethernet interfaces.

The distributed 10-Gigabit, , 40-Gigabit, 100-Gigabit Ethernet, 400-Gigabit 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, Layer 2 switches and Layer 3 switches.

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Introduction of IP MTU on Q200-based Systems	Release 7.5.2	You can configure IP MTU for IPv4 and IPv6 on a Layer 3 interface. Depending on your specific network requirements, this ability to specify IP MTU settings helps optimize router data transmission. Use the `show ipv4/ipv6 interfaces` command to view the IP MTU configurations.

Tip You can programmatically configure and manage the Ethernet interfaces using `openconfig-if-ethernet.yang` and `openconfig-interfaces.yang` OpenConfig data models. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

Prerequisites for Configuring Ethernet Interfaces

Before configuring Ethernet interfaces, ensure that you meet the following conditions:

- Access to Cisco 8200 series routers or Cisco 8800 series routers with at least one of the supported line cards installed.
- Know the interface IP address.
- Ensure to specify the generalized interface name with the standard notation of *rack/slot/module/port*.

Note An ACL-dependent feature refers to a capability in network systems that relies on Access Control Lists (ACLs) for its operation. These features include both global such as Lawful Intercept (LI), BGP Flow Specification (BGPFS) and interface-level configurations, such as Quality of Service with ACL (QoS-ACL), Security ACL, SPAN ACL, Qos Policy Propagation via BGP (QPPB), Policy Based Routing (PBR), Peering QoS, and L2 ACL for packets with L3 payload.

An interface, whether physical or virtual, supports the configuration of up to four ACL dependent features, such as ACLs, QoS with ACL, BGP Flow Specification, SPAN ACL, and Lawful Intercept. To add a new feature, such as Policy-Based Routing, you must first remove one of the existing features and then configure the new feature.

Information About Configuring Ethernet

This section provides the following information:

Cisco 8000 Modular Line Cards

The current release of the Cisco 8800 Series Routers support the following line cards:

- 36-port QSFP56-DD 400 GbE Line Card - This line card provides 14.4 Tbps via 36 QSFP56-DD ports. It also supports 100G, 2x100G, and 400G modules. If 36 of 2x100G modules are used, the line card can have 72 HundredGigE interfaces.
- 48-port QSFP28 100 GbE Line Card - This line card provides 4.8 Tbps with MACsec support on all ports. It also supports QSFP+ optics for 40G compatibility.

The 8800 Series line cards utilize multiple #ChipName forwarding ASICs to achieve high performance and bandwidth with line rate forwarding.

Default Configuration Values for 100-Gigabit Ethernet

This table describes the default interface configuration parameters that are present when an interface is enabled on a 36-port Line Card or a 48-port Line Card.

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

Table 5: 100-Gigabit Ethernet Line Card Default Configuration Values

Parameter	Configuration File Entry	Default Value
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)

Layer 2 VPN on Ethernet Interfaces

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

The L2VPN feature enables service providers (SPs) to provide Layer 2 services to geographically disparate customer sites. Typically, an SP uses an access network to connect the customer to the core network. On the router, this access network is typically Ethernet.

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.

On the router, an AC is an interface that is attached to an L2VPN component, such as a bridge domain.

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

Switching takes place through local switching where traffic arriving on one AC is immediately sent out of another AC without passing through a pseudowire.

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 SP network.

Use the **show interfaces** command to display AC 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 various speeds over various physical media. The IEEE 802.3 standard covers 10 Mbps Ethernet. Extensions to the IEEE 802.3 standard specify implementations for 40-Gigabit Ethernet and 100-Gigabit Ethernet.

IEEE 802.3ae 10-Gbps Ethernet

Under the International Standards Organization's Open Systems Interconnection (OSI) model, Ethernet is fundamentally a Layer 2 protocol. 10-Gigabit Ethernet uses the IEEE 802.3 Ethernet MAC protocol, the IEEE 802.3 Ethernet frame format, and the minimum and maximum IEEE 802.3 frame size. 10-Gbps Ethernet conforms to the IEEE 802.3ae protocol standards.

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

IEEE 802.3ba 100 Gbps Ethernet

IEEE 802.3ba is supported on the Cisco 1-Port 100-Gigabit Ethernet PLIM beginning in Cisco IOS XR 7.0.11.

MAC Address

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

Ethernet MTU

The Ethernet maximum transmission unit (MTU) is the size of the largest frame, minus the 4-byte frame check sequence (FCS), that the system transmits 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 unfragmented packet that the system can send. 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, the system fragments that packet as necessary. This process ensures that the originating device does not send an IP packet that is too large.

The system automatically enables the jumbo frame support 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).

IP MTU

In IP protocol, Maximum Transmission Unit (MTU) refers to the maximum size of an IP packet that the system transmits without fragmentation over a given medium. The size of an IP packet includes IP headers but excludes headers from the data link layer, also known as the Ethernet headers. The default IP MTU on all router interfaces is 1500 bytes, when IP is enabled by using the IP address configuration commands. However, you can configure the IP MTU to different value as well.

Starting Cisco IOS XR Release 7.5.2, the system supports IP MTU (IPv4 and IPv6) on Q200-based systems on the following Cisco 8000 Series router and line card:

- 8201-32FH
- 88-LC0-36FH-M

How is Ethernet MTU different from IP MTU?

Ethernet MTU defines the maximum packet size that an interface supports, while IP MTU defines the MTU size of an IP packet.

How is IP MTU calculated?

The following scenarios provide information on how the system calculates IP MTU size, when:

- Ethernet MTU is not configured, the system sets:
  - Default value as 1514 bytes on a physical or bundle main interface
  - 1518 bytes for single tag VLAN interface
  - 1522 bytes for double tag VLAN (QinQ) interface

Note In this case, IP MTU value will be 1500 bytes for IPv4 and IPv6 MTUs.

- Ethernet MTU is configured as X bytes, the system sets:
  - X bytes on a physical or bundle main interface

- X+4 bytes for single tag VLAN interface
- X+8 bytes for double tag VLAN (QinQ) interface

Note In this case, IP MTU is X-14 bytes for IPv4 and IPv6 MTUs.

The following are some of the important guidelines for IP MTU size:

- When no Ethernet MTU and IP MTU is configured, the default value is 1500B.
- When Ethernet MTU and no IP MTU is configured, IP MTU value in the hardware is Ethernet MTU-14B.
- IP MTU value can't be more than Ethernet MTU value. For example, if the Ethernet MTU value is 3000 bytes and you configure IP MTU value as 5000 bytes. The system sets the IP MTU value as 2986 (3000-14)

What is Maximum Receive Unit (MRU) and how is it different from MTU?

MRU is the largest packet size that an interface can receive. This is an ingress parameter. Usually, MRU equals MTU. However, you can't configure an MRU value. The Ethernet MTU, also known as a Layer2 (L2) value that you configure on a physical interface is also applied as the MRU of that physical interface.

The following table lists Ethernet MTU, IPv4, and IPv6 MTU support across various platforms and their limitations as applicable:

Table 6: IP MTU Support Across Platforms

ASIC	Ethernet MTU Check	IPv4 and IPv6 MTU
Q100-based	Ethernet MTU check on an egress interface is not supported.	Supported and the system derives IP MTU value from Ethernet MTU.
Q200-based	Ethernet MTU check on an egress interface is not supported.	Supported **Note** IPv4 and IPv6 can have their own separate MTU values.

IP MTU Checks

Cisco IOS XR supports MRU checks, ethernet MTU checks, and IP MTU checks. These checks decide if an IP payload packets needs fragmentation or not. Ethernet MTU and IP MTU check is applied on all egress traffic for each packet that flows in the system. In the forwarding plane, also known as dataplane, the IP packet length is compared against IP MTU value applied on the L3 forwarding interface. Full packet length is compared against ethernet MTU value applied on the physical port.

The following table describes IP MTU check to each forwarding flow on different systems.

Table 7: IP MTU Check

ASIC	IP MTU Check
Q100-based	These interfaces implement IP MTU checks and provide fragmentation for all IP payload packets. A single value for both IPv4 and IPv6 is supported.
Q200-based	These interfaces implement IP MTU checks and provide fragmentation for all IP payload packets.
Mixed System with Q100 and Q200-based line cards	Interfaces with Q100-based system provides its behavior and interfaces with Q200-based systems provides its behavior as described in this table.

IP MTU Configuration Guidelines

An Ethernet interface has a default of 1514 bytes. IP MTU is always derived from the default Ethernet MTU. If Ethernet MTU is configured, IP MTU is derived from the previously configured Ethernet MTU.

The following are the configuration guidelines for IP MTU across various platforms:

Guidelines for Q200-based Systems

- If you don't define any Ethernet MTU configuration, the system assigns the default value of Ethernet packet size of 1514 bytes.
- For IPv4 and IPv6 MTUs, 1500 bytes is used, which is derived by subtracting 14 bytes of Ethernet header size from its default value.
- On any interface, if an Ethernet MTU is configured, the new configuration takes higher precedence than the default interface configuration. However, in such a scenario, the new configuration doesn't get applied on the subinterfaces. The rest of the interfaces continue to work with the default Ethernet MTU configurations.
- For the double tag (QinQ) packets, MRU and Ethernet MTU value are derived by adding 8 bytes to the configured value.
- IPv4 and IPv6 MTU values are derived by subtracting 14 bytes of the Ethernet header size out of the configured value.
- If an IPv4 or IPv6 MTU is configured, it is applied to IP MTU value. MRU or MTU values remain unaffected by this configuration.
- Configuration restrictions apply to validate that the IP MTU value is smaller than Ethernet MTU value that is configured on an interface.

The following sample table explains how the system calculates the Ethernet MTU, IPv4, and IPv6 MTU values when configured on various interfaces that are Q200-based systems.

Table 8: Interfaces and Configurations

Interface Type	Default Configurations *when, no Ethernet or IP MTU configuration is applied*	Ethernet MTU Configurations *for example, MTU = 1614*	IPv4/IPv6 Configurations *for example, IPv4 | IPv6 MTU = 1550, Ethernet MTU = 1614*
Any Main Interface	NA	Configuration that is applied on any main interface and its subinterfaces is effective. Configuration that is applied on any subinterface is not effective.	NA
Physical Port	MRU = 1514 + 8	MRU = 1614 + 8	MRU = 1614 + 8
L3 Main Interface (Physical or bundle)	Ethernet MTU = 1514 + 8 IPv4/IPv6 MTU = 1500	Ethernet MTU = 1614 + 8 IPv4/IPv6 MTU = 1614 - 14	Ethernet MTU = 1614 + 8 IPv4/IPv6 MTU = 1550
L3 Sub-interface (Physical or bundle)	Ethernet MTU = 1514 + 8 IPv4/IPv6 MTU = 1500	**Note** Configuration not applicable on subinterfaces. Takes main interface configuration. Ethernet MTU = Not applicable IPv4/IPv6 MTU = 1614 - 14	Ethernet MTU = 1614 + 8 IPv4/IPv6 MTU = 1550
SVI/BVI	IPv4/IPv6 MTU = 1500	Ethernet MTU = Not applicable IPv4/IPv6 MTU = 1600	IPv4/IPv6 MTU = 1550
Ethernet Main Interface	Ethernet MTU = 1514 + 8	Ethernet MTU = 1614 + 8 IPv4/IPv6 MTU = Not applicable	Not applicable Ethernet MTU = 1614 + 8
Ethernet Sub-interface	Ethernet MTU = 1514 + 8	Ethernet MTU = 1614 + 8 IPv4/IPv6 MTU = Not applicable	Not applicable Ethernet MTU = 1614 + 8

Guidelines for Q100-based Systems

- If you don't define any MTU configuration in the system, the system assigns default value of 1514 bytes to the Ethernet packets.
- An MRU of 1522 bytes is set to allow for any double tag packets on an interface.
- Ethernet MTU check on egress interface is not supported.
- For IPv4 and IPv6 MTUs, 1500 bytes is used, which is derived by subtracting 14 bytes of Ethernet header size from its default value.
- If you configure an IP MTU (IPv4/IPv6 MTU) on any interface, the configuration does not take effect.

The following table lists the MTU configurations and MTU calculations on various interfaces for Q100-based systems:

Table 9: Interfaces and Configurations

Interface Type	Default Configurations	Interface MTU Configurations ***where, MTU = 1614***
Any Main Interface	NA	Configuration applied on any main interface and its subinterfaces is effective. Configuration applied on any sub-interface is not effective.
Physical Port	MRU = 1514 + 8	MRU = 1614 + 8
L3 Main Interface (Physical or bundle)	Ethernet MTU check on egress interface is not supported. IPv4/IPv6 MTU = 1522 applied on full packet size	Ethernet MTU check on egress interface is not supported. IPv4/IPv6 MTU = 1622 applied on full packet size

Interface Type	Default Configurations	Interface MTU Configurations *where, MTU = 1614*
L3 Subinterface (Physical or bundle)	Ethernet MTU check on egress interface is not supported. IPv4/IPv6 MTU = 1522 applied on full packet size	**Note** Configuration not applicable on sub-interfaces. Takes main interface configuration. Ethernet MTU = Not applicable IPv4/IPv6 MTU = 1622 applied on full packet size
SVI and BVI	Ethernet MTU is not applicable IPv4/IPv6 MTU = 1522 applied on full packet size	
Ethernet Main Interface	Ethernet MTU check on egress interface is not supported. IPv4/IPv6 MTU is not applicable	
Ethernet Subinterface	Ethernet MTU check on egress interface is not supported. IPv4/IPv6 MTU is not applicable	
GRE tunnel	GRE interface uses the configured L3 MTU. If an egress packet is above the configured L3 MTU value, the packet is discarded.	

IP MTU Limitations and Feature Support

The following are the limitations and feature support for IP MTU:

- MPLS MTU is not supported.
- GRE/IPinIP MTU is not supported.

Note GRE interface uses the configured L3 MTU. If an egress packet is above the configured L3 MTU value, the packet is discarded.

- Multicast MDT MTU is not supported.
- Ethernet MTU configuration on sub-interfaces is not supported.
- Ethernet MTU configuration on BVI interfaces is not supported.

Important Although CLIs for the unsupported features might be available on your router, they are not functional.

IP MTU Scale

The following are the scale support and limitations for IP MTU configurations:

- Ethernet MTU configuration is allowed on every physical or bundle interface without any scale limits.
- IPv4/IPv6 MTU configuration is allowed on every L3 forwarding interface. However, the system applies it as an IP MTU profile with a combination of <IPv4 MTU and IPv6 MTU> set. Each profile takes a unique set of these two MTUs, and supports eight such unique MTUs. The IP MTU profiles are stored identically across all NPUs of a given line card.
- When system runs out of resources (OOR), the system generates syslog messages while the continuing to use the previously configured IP MTU values.

Configure IP MTU

To configure IPv4, or IPv6 MTU, you must be in the configuration mode on L3 interface.

To configure IP MTU, use the following configuration steps:

```
Router#config t

Router#interface HundredGigE0/0/0/33

Router:abc(config)#

Router:abc(config-if)#ipv4 mtu 2000

Router:abc(config-subif)#commit

Router:abc(config-if)#end
```

Running Configuration

```
Router:abc#
Router:abc#sh ipv4 int HundredGigE0/0/0/33
Thu Apr 28 16:54:10.820 UTC
HundredGigE0/0/0/33 is Shutdown, ipv4 protocol is Down
  Vrf is default (vrfid 0x60000000)
  Internet address is 10.10.10.91/29
  MTU is 9642 (2000 is available to IP)
```

Verification

```
Router:abc#sh int HundredGigE0/0/0/33
Thu Apr 28 16:57:23.390 UTC
HundredGigE0/0/0/33 is administratively down, line protocol is administratively down
  Interface state transitions: 0
  Hardware is HundredGigE, address is e41f.7bde.123c (bia e41f.7bde.123c)
  Internet address is 194.19.242.94/29
  MTU 9642 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 100000Mb/s, 100GBASE-AOC, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  30 second input rate 0 bits/sec, 0 packets/sec
```

```
  30 second 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


Router:abc#
```

Configure IPv4 MTU

To configure IPv4 MTU, use the following configuration steps:

```
Router#config t

Router#interface HundredGigE0/0/0/33

Router:abc(config)#

Router:abc(config-if)#ipv4 mtu 2000

Router:abc(config-subif)#commit

Router:abc(config-if)#end
```

Running Configuration

```
Router:abc#sh run int HundredGigE0/0/0/33
Thu Apr 28 16:22:06.796 UTC
interface HundredGigE0/0/0/33
mtu 9642
ipv4 mtu 2000
ipv4 address 10.10.10.1 255.255.255.248
ipv6 address 10::1:10:9/112
load-interval 30
!
```

Verification

```
Router:abc#sh ipv4 int HundredGigE0/0/0/33
Thu Apr 28 16:54:10.820 UTC
HundredGigE0/0/0/33 is Shutdown, ipv4 protocol is Down
  Vrf is default (vrfid 0x60000000)
  Internet address is 10.10.10.1/29
  MTU is 9642 (2000 is available to IP)
```

Configure IPv6 MTU

To configure IPv6 MTU, use the following configuration steps:

```
Router#config t

Router#interface HundredGigE0/0/0/33

Router:abc(config)#

Router:abc(config-if)#ipv6 mtu 3000
```

```
Router:abc(config-subif)#commit

Router:abc(config-if)#end
```

Running Configuration

```
Router:abc#sh run int HundredGigE0/0/0/33
Thu Apr 28 16:23:09.141 UTC
interface HundredGigE0/0/0/33
mtu 9642
ipv4 mtu 2000
ipv4 address 10.10.10.1 255.255.255.248
ipv6 mtu 3000
ipv6 address 10::10:10:9/112
load-interval 30
!

Router:abc#
```

Verification

```
Router:abc#sh ipv6 int HundredGigE0/0/0/33
Thu Apr 28 16:54:41.222 UTC
HundredGigE0/0/0/33 is Shutdown, ipv6 protocol is Down, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::e61f:7bff:fede:123c [TENTATIVE]
  Global unicast address(es):
    194::19:242:94, subnet is 10::10:10:0/112 [TENTATIVE]
  Joined group address(es): ff02::2 ff02::1
  MTU is 9642 (3000 is available to IPv6)
```

Flow Control on Ethernet Interfaces

The flow control that the system uses 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 that is used on standard management interfaces. You can activate or deactivate flow control for ingress traffic only. The system automatically implements flow control for egress traffic.

802.1Q VLAN

A VLAN is a group of devices on one or more LANs that the system configures so that they can communicate as if they are attached to the same wire, when in fact they are located on several different LAN segments. Because VLANs are based on logical instead of physical connections, it is 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 to 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.

Interfaces and subinterfaces on the router

In Cisco IOS XR, interfaces are, by default, main interfaces. A main interface is also known as a trunk interface, which you must not confuse with the word trunk in the context of VLAN trunking.

There are two types of trunk interfaces:

- Physical
- Bundle

On the router, the system automatically creates the physical interfaces when the router recognizes a card and its physical interfaces. However, the system does not automatically create bundle interfaces but you must create them at the time of cofiguration.

The following configuration samples are examples of the trunk interfaces that you can create:

- interface HundredGigE 0/5/0/0
- interface bundle-ether 1

A subinterface is a logical interface that the system create under a trunk interface.

To create a subinterface, you must first identify a trunk interface under which to place it. In case of bundle interfaces, if a trunk interface does not exist, you must create a bundle interface before creating any subinterfaces under it.

You can then assign a subinterface number to the subinterface that you want to create. The subinterface number must be a positive integer from zero to some high value. For a given trunk interface, each subinterface under it must have a unique value.

Subinterface numbers do not need to be contiguous or in numeric order. For example, the following subinterfaces numbers are valid under one trunk interface:

1001, 0, 97, 96, 100000

Subinterfaces can never have the same subinterface number under one trunk.

In the following example, the card in slot 5 has trunk interface, HundredGigE 0/5/0/0. A subinterface, HundredGigE 0/5/0/0.0, is created under it.

```
RP/0/RSP0/CPU0:router#  conf
Mon Sep 21 11:12:11.722 EDT
RP/0/RSP0/CPU0:router(config)#  interface HundredGigE0/5/0/0.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit

RP/0/RSP0/CPU0:Sep 21 11:12:34.819 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000152' to view the
changes.

RP/0/RSP0/CPU0:router(config-subif)# end

RP/0/RSP0/CPU0:Sep 21 11:12:35.633 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by root
RP/0/RSP0/CPU0:router#
```

The **show run** command displays the trunk interface first, then the subinterfaces in ascending numerical order.

```
RP/0/RSP0/CPU0:router# show run | begin HundredGigE 0/5/0/0
Mon Sep 21 11:15:42.654 EDT
Building configuration...
interface HundredGigE 0/5/0/0
 shutdown
!
interface HundredGigE 0/5/0/0.0
```

```
 encapsulation dot1q 100
!
interface HundredGigE 0/5/0/1
 shutdown
!
```

When a subinterface is first created, the router recognizes it as an interface that, with few exceptions, is interchangeable with a trunk interface. After the new subinterface is configured further, the **show interface** command can display it along with its unique counters:

The following example shows the display output for the trunk interface, HundredGigE 0/5/0/0, followed by the display output for the subinterface HundredGigE 0/5/0/0.0.

```
RP/0/RSP0/CPU0:router# show interface HundredGigE 0/5/0/0
Mon Sep 21 11:12:51.068 EDT
HundredGigE0/5/0/0 is administratively down, line protocol is administratively down.
  Interface state transitions: 0
  Hardware is HundredGigE, address is 0024.f71b.0ca8 (bia 0024.f71b.0ca8)
  Internet address is Unknown
  MTU 1514 bytes, BW 1000000 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN,
  Full-duplex, 1000Mb/s, SXFD, link type is force-up
  output flow control is off, input flow control is off
  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions


RP/0/RSP0/CPU0:router# show interface HundredGigE0/5/0/0.0
Mon Sep 21 11:12:55.657 EDT
HundredGigE0/5/0/0.0 is administratively down, line protocol is administratively down.
  Interface state transitions: 0
  Hardware is VLAN sub-interface(s), address is 0024.f71b.0ca8
  Internet address is Unknown
  MTU 1518 bytes, BW 1000000 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN, VLAN Id 100,  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  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
```

This example shows two interfaces being created at the same time: first, the bundle trunk interface, then a subinterface attached to the trunk:

```
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 10:57:31.736 EDT
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1
RP/0/RSP0/CPU0:router(config-if)# no shut
RP/0/RSP0/CPU0:router(config-if)# interface bundle-Ether1.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:Sep 21 10:58:15.305 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : C
onfiguration committed by user 'root'. Use 'show configuration commit changes 10
00000149' to view the changes.
RP/0/RSP0/CPU0:router# show run | begin Bundle-Ether1
Mon Sep 21 10:59:31.317 EDT
Building configuration..
interface Bundle-Ether1
!
interface Bundle-Ether1.0
 encapsulation dot1q 100
!
```

You delete a subinterface using the **no interface** command.

```
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# show run | begin HundredGigE 0/5/0/0
Mon Sep 21 11:42:27.100 EDT
Building configuration...
interface HundredGigE 0/5/0/0
 negotiation auto
!
interface HundredGigE 0/5/0/0.0
 encapsulation dot1q 100
!
interface HundredGigE 0/5/0/1
 shutdown
!
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 11:42:32.374 EDT
RP/0/RSP0/CPU0:router(config)# no interface HundredGigE 0/5/0/0.0
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:Sep 21 11:42:47.237 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000159' to view the
changes.
RP/0/RSP0/CPU0:router(config)# end
RP/0/RSP0/CPU0:Sep 21 11:42:50.278 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by root
RP/0/RSP0/CPU0:router# show run | begin HundredGigE 0/5/0/0
Mon Sep 21 11:42:57.262 EDT
Building configuration...
interface HundredGigE 0/5/0/0
 negotiation auto
!
interface HundredGigE 0/5/0/1
 shutdown
!
```

Layer 2, Layer 3, and EFPs

On the router, a trunk interface can be either a Layer 2 or Layer 3 interface. A Layer 2 interface is configured using the **interface** command with the **l2transport** keyword. When the **l2transport** keyword is not used, the

interface is a Layer 3 interface. Subinterfaces are configured as Layer 2 or Layer 3 subinterface in the same way.

A Layer 3 trunk interface or subinterface is a routed interface and can be assigned an IP address. Traffic sent on that interface is routed.

A Layer 2 trunk interface or subinterface is a switched interface and cannot be assigned an IP address. A Layer 2 interface must be connected to an L2VPN component. Once it is connected, it is called an access connection.

Subinterfaces can only be created under a Layer 3 trunk interface. Subinterfaces cannot be created under a Layer 2 trunk interface.

A Layer 3 trunk interface can have any combination of Layer 2 and Layer 3 interfaces.

The following example shows an attempt to configure a subinterface under an Layer 2 trunk and the commit errors that occur. It also shows an attempt to change the Layer 2 trunk interface to an Layer 3 interface and the errors that occur because the interface already had an IP address assigned to it.

```
RP/0/RP0/CPU0:router# config
Mon Sep 21 12:05:33.142 EDT
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/5/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.0.0.1/24
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:Sep 21 12:05:57.824 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000160' to view the
changes.
RP/0/RP0/CPU0:router(config-if)# end
RP/0/RP0/CPU0:Sep 21 12:06:01.890 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
console by root
RP/0/RP0/CPU0:router# show run | begin HundredGigE0/5/0/0
Mon Sep 21 12:06:19.535 EDT
Building configuration...
interface HundredGigE0/5/0/0
 ipv4 address 10.0.0.1 255.255.255.0
 negotiation auto
!
interface HundredGigE0/5/0/1
 shutdown
!
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router# conf
Mon Sep 21 12:08:07.426 EDT
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/5/0/0 l2transport
RP/0/RP0/CPU0:router(config-if-l2)# commit

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed' from this session
 to view the errors
RP/0/RP0/CPU0:router(config-if-l2)# no ipv4 address
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:Sep 21 12:08:33.686 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000161' to view the
changes.
RP/0/RP0/CPU0:router(config-if)# end
RP/0/RP0/CPU0:Sep 21 12:08:38.726 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
console by root
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router# show run interface HundredGigE0/5/0/0
Mon Sep 21 12:09:02.471 EDT
interface HundredGigE0/5/0/0
```

```
 negotiation auto
 l2transport
 !
!
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router# conf
Mon Sep 21 12:09:08.658 EDT
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/5/0/0.0
                                                        ^
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/5/0/0.0
RP/0/RP0/CPU0:router(config-subif)# commit

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed' from this session
 to view the errors
RP/0/RP0/CPU0:router(config-subif)#
RP/0/RP0/CPU0:router(config-subif)# interface HundredGigE0/5/0/0
RP/0/RP0/CPU0:router(config-if)# no l2transport
RP/0/RP0/CPU0:router(config-if)# interface HundredGigE0/5/0/0.0
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 99
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 11.0.0.1/24
RP/0/RP0/CPU0:router(config-subif)# interface HundredGigE0/5/0/0.1 l2transport
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 700
RP/0/RP0/CPU0:router(config-subif)# commit
RP/0/RP0/CPU0:Sep 21 12:11:45.896 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000162' to view the
changes.
RP/0/RP0/CPU0:router(config-subif)# end
RP/0/RP0/CPU0:Sep 21 12:11:50.133 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
console by root
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router# show run | b HundredGigE0/5/0/0
Mon Sep 21 12:12:00.248 EDT
Building configuration...
interface HundredGigE0/5/0/0
 negotiation auto
!
interface HundredGigE0/5/0/0.0
 ipv4 address 11.0.0.1 255.255.255.0
 encapsulation dot1q 99
!
interface HundredGigE0/5/0/0.1 l2transport
 encapsulation dot1q 700
!
interface HundredGigE0/5/0/1
 shutdown
!
```

All subinterfaces must have unique encapsulation statements, so that the router can send incoming packets and frames to the correct subinterface. If a subinterface does not have an encapsulation statement, the router will not send any traffic to it.

In Cisco IOS XR, an Ethernet Flow Point (EFP) is implemented as a Layer 2 subinterface, and consequently, a Layer 2 subinterface is often called an EFP.

A Layer 2 trunk interface can be used as an access connection. However, a Layer 2 trunk interface is not an EFP because an EFP, by definition, is a substream of an overall stream of traffic.

Cisco IOS XR also has other restrictions on what can be configured as a Layer 2 or Layer 3 interface. Certain configuration blocks only accept Layer 3 and not Layer 2. For example, OSPF only accepts Layer 3 trunks and subinterface. Refer to the appropriate Cisco IOS XR configuration guide for other restrictions.

Enhanced Performance Monitoring for Layer 2 Subinterfaces (EFPs)

Beginning in Cisco IOS XR Release 7.2.12, the router adds support for basic counters for performance monitoring on Layer 2 subinterfaces. This section provides a summary of the new support for Layer 2 interface counters.

The **interface basic-counters** keyword has been added to support a new entity for performance statistics collection and display on Layer 2 interfaces in the following commands:

- **performance-mgmt statistics interface basic-counters**
- **performance-mgmt threshold interface basic-counters**
- **performance-mgmt apply statistics interface basic-counters**
- **performance-mgmt apply threshold interface basic-counters**
- **performance-mgmt apply monitor interface basic-counters**
- show performance-mgmt monitor interface basic-counters
- show performance-mgmt statistics interface basic-counters

The **performance-mgmt threshold interface basic-counters** command supports the following attribute values for Layer 2 statistics, which also appear in the **show performance-mgmt statistics interface basic-counters** and **show performance-mgmt monitor interface basic-counters** command:

Attribute	Description
InOctets	Bytes received (64-bit)
InPackets	Packets received (64-bit)
InputQueueDrops	Input queue drops (64-bit)
InputTotalDrops	Inbound correct packets discarded (64-bit)
InputTotalErrors	Inbound incorrect packets discarded (64-bit)
OutOctets	Bytes sent (64-bit)
OutPackets	Packets sent (64-bit)
OutputQueueDrops	Output queue drops (64-bit)
OutputTotalDrops	Outband correct packets discarded (64-bit)
OutputTotalErrors	Outband incorrect packets discarded (64-bit)

Other Performance Management Enhancements

The following additional performance management enhancements are included in Cisco IOS XR Release 7.0.11:

- You can retain performance management history statistics across a process restart or route processor (RP) failover using the new **history-persistent** keyword option for the **performance-mgmt statistics interface** command.

- You can save performance management statistics to a local file using the **performance-mgmt resources dump local** command.
- You can filter performance management instances by defining a regular expression group (**performance-mgmt regular-expression** command), which includes multiple regular expression indices that specify strings to match. You apply a defined regular expression group to one or more statistics or threshold templates in the **performance-mgmt statistics interface** or **performance-mgmt thresholds** interface commands.

Frequency Synchronization and SyncE

Cisco IOS XR Software supports SyncE-capable Ethernet on the router. Frequency Synchronization enables you to distribute the precision clock signals around the network. The system injects a highly accurate timing signal into the router in the network. The timing signals use an external timing technology, such as Cesium atomic clocks, or GPS, and then pass the signals to the physical interfaces of the router. Peer routers can then recover this precision frequency from the line, and also transfer it around the network. This feature is traditionally applicable to SONET or SDH networks, but is now available on Ethernet for Cisco 8000 Series Router with Synchronous Ethernet capability. For more information, see *Cisco 8000 Series Router System Management Configuration Guide*.

LLDP

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
LLDP Snooping	Release 7.3.3	With this release, you can further leverage the Link Layer Discovery Protocol (LLDP) information for directly attached devices or equipment in an L2 (Layer 2) network via LLDP snoop. In order to utilize the LLDP snoop functionality, the neighbouring devices must exchange the LLDP packets with the L2 network. With the help of the LLDP snoop functionality, you can identify the cabling and modeling failures and isolate faults. To enable LLDP snoop, enable the LLDP command on an interface while the outgoing (TX) traffic is disabled.

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

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

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

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

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

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

Note MAcsec encrypts LLDP packets by default. You can enable exceptions in the MACsec policy using the **allow lldp-in-clear** command to retain the LLDP packets unencrypted with MACsec. For more information, see *MACsec Policy Exceptions for Link Layer Discovery Protocol Packets* section in the *Configuring MACsec* chapter of the *System Security Configuration Guide for Cisco 8000 Series Routers*.

Note For LLDP to work on any bundle member, enable LLDP on the bundle main interface either globally or on the interface itself. You can then choose to disable LLDP transmission on bundle main interface by using the `lldp transmit disable` command.

You can also control LLDP transmit or receive on each bundle member interface as desired.

LLDP Frame Format

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

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

LLDP TLV Format

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

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

Specifying User-Defined LLDP TLV Values

It is possible to override the system default values for some of the mandatory LLDP Type-Length-Values (TLVs) that are advertised by routers to their directly connected neighboring devices. While advertising their identity and capabilities, routers can assign user-defined meaningful names instead of autogenerated values. Using the following CLIs you can specify these user-defined values:

- `Router(config)#lldp system-name` *`system-name`*
- `Router(config)#lldp system-description` *`system-description`*
- `Router(config)#lldp chassis-id-type` *`chassis-type`*
- `Router(config)#lldp chassis-id` *`local-chassis-id`*

Note The **chassis-id** value is configurable only when the **chassis-id-type** is set as **Local**. If there is a mismatch, you encounter a configuration failed error message.

The configured values, such as the system name, system description, chassis-id, chassis-type become part of the TLV in the LLDP packets that are sent to its neighbors. Values are transmitted only to LLDP enabled interfaces to which the router is connected.

You can assign any of the following values for the `chassis-id-type`. The chassis-id-types are objects that are part of the management information base (MIB). Depending on the selected chassis-id-type, values are assigned to these objects, and they are advertised by the router to its neighboring devices.

chassis-id-type	Description
chassis-component	Chassis identifier based on the value of entPhysicalAlias object that is defined in IETF RFC 2737.
interface-alias	Chassis identifier based on the value of ifAlias object as defined in IETF RFC 2863.
interface-name	Chassis identifier based on the name of the interface.
local	Chassis identifier based on a locally defined value.

chassis-id-type	Description
mac-address	Chassis identifier based on the value of a unicast source address.
network-address	Chassis identifier based on a network address that is associated with a particular chassis.
port-component	Chassis identifier based on the value of entPhysicalAlias object defined in IETF RFC 2737 for a port or backplane component.

Tip You can programmatically modify default values of LLDP TLVs by using the `openconfig-lldp` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

Configuration Example

This example shows the configuration for the LLDP TLVs that will be advertised by routers to their directly connected neighboring devices.

```
Router(config)#lldp system-name cisco-xr
Router(config)#lldp system-description cisco-xr-edge-device
Router(config)#lldp chassis-id-type local
Router(config)#lldp chassis-id ce-device9
```

Running Configuration

```
Router#show lldp
Tue Sep 13 16:03:44.550 +0530
Global LLDP information:
Status: ACTIVE
LLDP Chassis ID: ce-device9
LLDP Chassis ID Subtype: Locally Assigned Chassis Subtype
LLDP System Name: cisco-xr
LLDP advertisements are sent every 30 seconds
LLDP hold time advertised is 120 seconds
LLDP interface reinitialisation delay is 2 seconds
```

LLDP Operation

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

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

LLDP supports the following additional operational characteristics:

- LLDP can operate independently in transmit or receive modes. On global lldp enablement, the default mode is to operate in both transmit and receive modes.

- LLDP operates as a slow protocol with transmission speeds not greater than one frame per five seconds.
- LLDP packets are sent when the following occurs:
  - The packet update frequency specified by the **lldp timer** command is reached. The default is 30 seconds.
  - When a change in the values of the managed objects occurs from the local system's LLDP MIB.
  - When LLDP is activated on an interface (3 frames are sent upon activation similar to CDP).
- When an LLDP frame is received, the LLDP remote services and PTOPO MIBs are updated with the information in the TLVs.
- LLDP supports the following actions on these TLV characteristics:
  - Interprets a neighbor TTL value of 0 as a request to automatically purge the information of the transmitting device. These shutdown LLDPDUs are typically sent prior to a port becoming inoperable.
  - An LLDP frame with a malformed mandatory TLV is dropped.
  - A TLV with an invalid value is ignored.
  - A copy of an unknown organizationally-specific TLV is maintained if the TTL is non-zero, for later access through network management.

Supported LLDP Functions

The router supports the following LLDP functions:

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

  If the transmitting interface does not have a configured address, then the system populates the TLV with an address from another interface. The advertised LLDP IP address is implemented according to the following priority order of IP addresses for interfaces on the router:
  - Locally configured address on the transmitting interface
  - MgmtEth0/RSP0RP0/CPU0/0
  - MgmtEth0/RSP0RP0/CPU0/1
  - MgmtEth0/RSP1RP1/CPU0/0
  - MgmtEth0/RSP1RP1/CPU0/1
  - Loopback interfaces

    There are some differences between IPv4 and IPv6 address management in LLDP:
- For IPv4, as long as the IPv4 address is configured on an interface, it can be used as an LLDP management address.
- For IPv6, after the IPv6 address is configured on an interface, the interface status must be Up and pass the DAD (Duplicate Address Detection) process before it is can be used as an LLDP management address.
- LLDP is supported for the nearest physically attached, non-tunneled neighbors.

- LLDP is supported for Ethernet interfaces, L3 subinterfaces, bundle interfaces, and L3 bundle subinterfaces.
- LLDP snoop is supported on L2 interfaces, when the incoming (RX) traffic is enabled and outgoing (TX) traffic is disabled.

Unsupported LLDP Functions

The following LLDP functions are not supported on the router:

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

Setting the carrier delay on physical interfaces

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Setting the carrier delay on physical interfaces	Release 7.5.4	You can configure the Ethernet interfaces to delay the processing of hardware link-down and link-up notifications. With this functionality, the interface state remains stable for the configured delay duration, even if the hardware link state fluctuates. This prevents interface flapping and improves network reliability. Use the following CLI command in interface configuration mode to configure the delay time: **carrier-delay**

Hardware links take time to stabilize after a state change and may experience link flaps. Link flap is a condition where a physical interface frequently fluctuates between an up and a down state.

During link flaps, the network reestablishes and updates routing paths after a disruption, which leads to resource exhaustion on routers. To overcome the problem, we recommend waiting until the link state is stable before taking action.

The carrier delay introduces a delay in processing interface link-state notifications in the router to provide enough time for the interface link to stabilize.

When there is a change in the link state, the carrier-delay timer starts. If the link state goes up, the **carrier-delay up** timer starts. Similarly, when the link state goes down, the **carrier-delay down** timer starts. During this delay period, the Ethernet interface state remains unchanged even if the link is physically restored. Setting a delay timer ensures the link state is established before the interface becomes operational again and avoids unnecessary interface state changes and associated traffic rerouting.

Guidelines and Restrictions for Setting the Carrier Delay on Physical Interfaces

The following usage guidelines and restrictions are applicable for setting the carrier delay on physical interfaces:

- You can configure carrier-delay for only link-up, only link-down, or both link-up and link-down notifications.
- If the **carrier-delay down** *milliseconds* command is configured on a physical link that fails and cannot be recovered, link down detection time increases, and it may take longer for the routing protocols to reroute the traffic around the failed link.
- Loss of Signal (LOS) is not supported on carrier delay.

Configure the Carrier-delay Timer

Configuration Example

In this example, link-up and link-down notifications are configured to be delayed by 1000 ms and 150 ms using **carrier-delay** command.

```
Router#configure
Router(config)#interface HundredGigE 0/0/0/0
Router(config-if)#carrier-delay up 1000 down 150
Router(config-if)#commit
```

Running Configuration

```
 interface HundredGigE0/0/0/0
 carrier-delay up 1000 down 150
!
```

Verification

Run the **show interfaces** command to see the current state of the carrier-delay configuration for an interface.

```
Router#show interfaces HundredGigE 0/0/0/0 | include Carrier
Fri Mar 31 07:25:05.273 UTC
Carrier delay (up) is 1000 msec, Carrier delay (down) is 150 msec
```

How to Configure Ethernet

This section provides the following configuration procedures:

Configuring LLDP

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

This section includes the following configuration topics for LLDP:

LLDP Default Configuration

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

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

Enabling LLDP Per Interface

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

```
RP/0/RSP0/CPU0:ios(config)# int HundredGigE 0/2/0/0
RP/0/RSP0/CPU0:ios(config-if)# no sh
RP/0/RSP0/CPU0:ios(config-if)#commit
RP/0/RSP0/CPU0:ios(config-if)#lldp ?
RP/0/RSP0/CPU0:ios(config-if)#lldp enable
RP/0/RSP0/CPU0:ios(config-if)#commit
```

Running configuration

```
RP/0/RSP0/CPU0:ios#sh running-config
Wed Jun 27 12:40:21.274 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Wed Jun 27 00:59:29 2018 by UNKNOWN
!
interface HundredGigE0/1/0/0
 shutdown
!
interface HundredGigE0/1/0/1
 shutdown
!
interface HundredGigE0/1/0/2
 shutdown
!
```

```
interface HundredGigE0/2/0/0
 Shutdown
!
interface HundredGigE0/2/0/1
 shutdown
!
interface HundredGigE0/2/0/2
 shutdown
!
end
```

Verification

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


HundredGigE0/2/0/0:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
RP/0/RSP0/CPU0:ios#

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

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

Total entries displayed: 1

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

Enabling LLDP Globally

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

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

Note For LLDP to work on any bundle member, enable LLDP on the bundle main interface either globally or on the interface itself. You can then choose to disable LLDP transmission on bundle main interface by using the `lldp transmit disable` command.

You can also control LLDP transmit or receive on each bundle member interface as desired.

The following table describes the global attributes that you can configure:

Attribute	Default	Range	Description
Holdtime	120	0-65535	Specifies the holdtime (in sec) that are sent in packets
Reinit	2	2-5	Delay (in sec) for LLDP initialization on any interface
Timer	30	5-65534	Specifies the rate at which LLDP packets are sent (in sec)

To enable LLDP globally, complete the following steps:

1. `RP/0//CPU0:router # configure`
2. `RP/0//CPU0:router(config) #lldp`
3. `end or commit`

Running configuration

```
RP/0/RP0/CPU0:turin-5#show run lldp
Fri Dec 15 20:36:49.132 UTC
lldp
!

RP/0/RP0/CPU0:turin-5#show lldp neighbors
Fri Dec 15 20:29:53.763 UTC
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf          Hold-time  Capability     Port ID
SW-NOSTG-I11-PUB.cis Mg0/RP0/CPU0/0      120        N/A            Fa0/28

Total entries displayed: 1

RP/0/RP0/CPU0:turin-5#show lldp neighbors mgmtEth 0/RP0/CPU0/0
Fri Dec 15 20:30:54.736 UTC
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf          Hold-time  Capability     Port ID
SW-NOSTG-I11-PUB.cis Mg0/RP0/CPU0/0      120        N/A            Fa0/28

Total entries displayed: 1
```

Configuring Global LLDP Operational Characteristics

When you enable LLDP globally on the router using the **lldp** command, these defaults are used for the protocol.

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

Procedure

Step 1 **Example:**

```
/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **lldp holdtime** *seconds*

Example:

```
RP/0//CPU0:router(config)#lldp holdtime 60
```

(Optional) Specifies the length of time that information from an LLDP packet should be held by the receiving device before aging and removing it.

Step 3 **lldp reinit** *seconds*

Example:

```
RP/0//CPU0:router(config)# lldp reinit 4
```

(Optional) Specifies the length of time to delay initialization of LLDP on an interface.

Step 4 **lldp timer** *seconds*

Example:

```
RP/0//CPU0:router(config)#lldp reinit 60
```

(Optional) Specifies the LLDP packet rate.

Step 5 **end** or **commit**

Example:

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

or

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

Saves configuration changes.

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

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

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

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

Disabling Transmission of Optional LLDP TLVs

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

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

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **lldp tlv-select** *tlv-name* **disable**

Example:

```
RP/0/RSP0/CPU0:router(config)# lldp tlv-select system-capabilities disable
```

(Optional) Specifies that transmission of the selected TLV in LLDP packets is disabled. The *tlv-name* can be one of the following LLDP TLV types:

- **management-address**
- **port-description**
- **system-capabilities**
- **system-description**
- **system-name**

Step 3 **end** or **commit**

Example:

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

or

```
RP/0/RSP0RP0/CPU0:router(config)# commit
```

Saves configuration changes.

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

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

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

Disabling LLDP Receive and Transmit Operation for an Interface

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

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

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface HundredGigE 0/2/0/0**

Example:

```
RP/0/RSP0RP0/CPU0:router(config)#interface HundredGigE 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:

- HundredGigE
- TenGigE

Step 3 **lldp**

Example:

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

(Optional) Enters LLDP configuration mode for the specified interface.

Step 4 **receive disable**

Example:

```
RP/0/RSP0/CPU0:router(config-lldp)#receive disable
```

(Optional) Disables LLDP receive operations on the interface.

Step 5 **transmit disable**

Example:

```
RP/0/RSP0/CPU0:router(config-lldp)#transmit disable
```

(Optional) Disables LLDP transmit operations on the interface.

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

Example:

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

or

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

Saves configuration changes.

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

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

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

Verifying the LLDP Configuration

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

Verifying the LLDP Global Configuration

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

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

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

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

Verifying the LLDP Interface Configuration

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

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


HundredGigE0/1/0/7:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
```

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

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

To collect or clear LLDP interface statistics, you can use the following commands:

Command	Description
show lldp traffic interface *interface_name*	Displays LLDP traffic statistics for the specified interface.

Command	Description
clear lldp counters interface *interface_name*	Clears LLDP traffic statistics for the specified interface. Global statistics remains intact. Similarly, clearing global statistics does not impact the interface statistics.

Examples for LLDP Interface Statistics

This example shows interface statistics for **gigabitEthernet0/0/0/0**:

```
Router#show lldp traffic interface gigabitEthernet0/0/0/0
```

This example clears the interface statistics for **gigabitEthernet0/0/0/0**.

```
Router#show lldp traffic interface gigabitEthernet0/0/0/0
```

Running Configuration

```
Router#show lldp traffic interface gigabitEthernet 0/2/0/8
Wed Aug 24 17:38:11.829 IST

LLDP Interface statistics:
        Total frames out: 28786
        Total frames in: 38417
        Total frames received in error: 0
        Total frames out error: 0
        Total frames discarded: 0
        Total TLVs discarded: 0
        Total TLVs unrecognized: 0
```

Configuring LLDP Snoop

If you have LLDP enabled on all Ethernet interfaces, the system enables Link Layer Discovery Protocol (LLDP) snoop on all L2 interfaces by default. You can use LLDP snooping to troubleshoot problems at the client ports.

Note LLDP snoop is enabled only when LLDP RX is enabled and LLDP TX (transmit) is disabled either on interface or global LLDP configuration.

To enable LLDP snoop on an L2 interface, perform the following steps:

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

RP/0/RSP0/CPU0:ios(config)# interface FourHundredGigE 0/0/0/5

RP/0/RSP0/CPU0:ios(config-if)#lldp

RP/0/RSP0/CPU0:ios(config-if)#enable

RP/0/RSP0/CPU0:ios(config-if)#transmit disable

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

Running Configuration

```
RP/0/RP0/CPU0:router#show run
Fri Jan 21 17:45:17.529 UTC
Building configuration...
!! IOS XR Configuration 7.7.1.06I
!! Last configuration change at Fri Jan 21 17:20:27 2022 by cisco
!
hostname router1
logging console disable
username xxxx
 group root-lr
 group cisco-support
 secret 10
$6$JELNK0oJaZZN7K0.$8YmyRWkq3D92i.lJc5QSdDkq4kUjU.g9U7sYIIAV1QVnSBemng5q.5EyYv6xSL9niDxRmKaFEATs9BkitDqpr.
!
line console
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
line default
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
vty-pool default 0 99 line-template default
call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method email disable
  destination transport-method http
 !
!
interface MgmtEth0/RP0/CPU0/0
 shutdown
!
interface FourHundredGigE0/0/0/0
 lldp
  enable
  transmit disable
 !
 l2transport
 !
!
interface FourHundredGigE0/0/0/1
 shutdown
!
interface FourHundredGigE0/0/0/2
 shutdown
!
interface FourHundredGigE0/0/0/3
 shutdown
!
interface FourHundredGigE0/0/0/4
 shutdown
!
interface FourHundredGigE0/0/0/5
 lldp
  enable
  transmit disable
 !
```

```
 l2transport
 !
!
interface FourHundredGigE0/0/0/6
 shutdown
!
interface FourHundredGigE0/0/0/7
 shutdown
!
interface FourHundredGigE0/0/0/8
 shutdown
!
interface FourHundredGigE0/0/0/9
 shutdown
!
interface FourHundredGigE0/0/0/10
 shutdown
!
interface FourHundredGigE0/0/0/11
 shutdown
!
interface FourHundredGigE0/0/0/12
 shutdown
!
interface FourHundredGigE0/0/0/13
 shutdown
!
interface FourHundredGigE0/0/0/14
 shutdown
!
interface FourHundredGigE0/0/0/15
 shutdown
!
interface FourHundredGigE0/0/0/16
 shutdown
!
interface FourHundredGigE0/0/0/17
 shutdown
!
interface FourHundredGigE0/0/0/18
 shutdown
!
interface FourHundredGigE0/0/0/19
 shutdown
!
interface FourHundredGigE0/0/0/20
 shutdown
!
interface FourHundredGigE0/0/0/21
 shutdown
!
interface FourHundredGigE0/0/0/22
 shutdown
!
interface FourHundredGigE0/0/0/23
 shutdown
!
interface HundredGigE0/0/0/24
 shutdown
!
interface HundredGigE0/0/0/25
 shutdown
!
interface HundredGigE0/0/0/26
```

```
 shutdown
!
interface HundredGigE0/0/0/27
 shutdown
!
interface HundredGigE0/0/0/28
 shutdown
!
interface HundredGigE0/0/0/29
 shutdown
!
interface HundredGigE0/0/0/30
 shutdown
!
interface HundredGigE0/0/0/31
 shutdown
!
interface HundredGigE0/0/0/32
 shutdown
!
interface HundredGigE0/0/0/33
 shutdown
!
interface HundredGigE0/0/0/34
 shutdown
!
interface HundredGigE0/0/0/35
 shutdown
!
l2vpn
 bridge group bg1
  bridge-domain bd1
   interface FourHundredGigE0/0/0/0
   !
   interface FourHundredGigE0/0/0/5
   !
  !
 !
!
end

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

Verification

```
router0 <———> router1 <———-> router2
          0/0/0/0           0/0/0/0/5

RP/0/RP0/CPU0:router0#config
Fri Jan 21 17:16:41.713 UTC
RP/0/RP0/CPU0:router0(config)#lldp
RP/0/RP0/CPU0:router0(config-lldp)#exit
RP/0/RP0/CPU0:router0(config)#int hu 0/0/0/0
RP/0/RP0/CPU0:router0(config-if)#no shut
RP/0/RP0/CPU0:router0(config-if)#end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes

RP/0/RP0/CPU0:router1#config
Fri Jan 21 17:17:41.459 UTC
RP/0/RP0/CPU0:router1(config)#int FourHundredGigE 0/0/0/0
RP/0/RP0/CPU0:router1(config-if)#no shut
RP/0/RP0/CPU0:router1(config-if)#l2transport
RP/0/RP0/CPU0:router1(config-if-l2)#exit
RP/0/RP0/CPU0:router1(config-if)#lldp
```

```
RP/0/RP0/CPU0:router1(config-lldp)#enable
RP/0/RP0/CPU0:router1(config-lldp)#transmit disable
RP/0/RP0/CPU0:router1(config-lldp)#exit
RP/0/RP0/CPU0:router1(config-if)#exit
RP/0/RP0/CPU0:router1(config)#int FourHundredGigE 0/0/0/5
RP/0/RP0/CPU0:router1(config-if)#no shut
RP/0/RP0/CPU0:router1(config-if)#l2transport
RP/0/RP0/CPU0:router1(config-if-l2)#exit
RP/0/RP0/CPU0:router1(config-if)#lldp
RP/0/RP0/CPU0:router1(config-lldp)#enable
RP/0/RP0/CPU0:router1(config-lldp)#transmit disable
RP/0/RP0/CPU0:router1(config-lldp)#exit
RP/0/RP0/CPU0:router1(config-if)#exit
RP/0/RP0/CPU0:router1(config)#l2vpn bridge group bg1
RP/0/RP0/CPU0:router1(config-l2vpn-bg)#bridge-domain bd1
RP/0/RP0/CPU0:router1(config-l2vpn-bg-bd)#interface FourHundredGigE 0/0/0/0
RP/0/RP0/CPU0:router1(config-l2vpn-bg-bd-ac)#exit
RP/0/RP0/CPU0:router1(config-l2vpn-bg-bd)#interface FourHundredGigE 0/0/0/5
RP/0/RP0/CPU0:router1(config-l2vpn-bg-bd-ac)#end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes


RP/0/RP0/CPU0:router0#config
Fri Jan 21 17:16:41.713 UTC
RP/0/RP0/CPU0:router0(config)#lldp
RP/0/RP0/CPU0:router0(config-lldp)#exit
RP/0/RP0/CPU0:router0(config)#int hu 0/0/0/0
RP/0/RP0/CPU0:router0(config-if)#no shut
RP/0/RP0/CPU0:router0(config-if)#end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes


RP/0/RP0/CPU0:router0#sh lldp neighbors
Fri Jan 21 17:21:15.857 UTC
Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf                  Hold-time  Capability      Port ID
router2            HundredGigE0/0/0/0          120        R
FourHundredGigE0/0/0/5

Total entries displayed: 1

RP/0/RP0/CPU0:router0#

RP/0/RP0/CPU0:router0#sh lldp neighbors
Fri Jan 21 17:21:15.857 UTC
Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf                  Hold-time  Capability      Port ID
router2            HundredGigE0/0/0/0          120        R
FourHundredGigE0/0/0/5

Total entries displayed: 1

RP/0/RP0/CPU0:router0#

RP/0/RP0/CPU0:router2#sh lldp neighbors
Fri Jan 21 17:21:20.998 UTC
Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
```

```
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf                     Hold-time  Capability      Port ID
router0            FourHundredGigE0/0/0/5          120        R
HundredGigE0/0/0/0

Total entries displayed: 1

RP/0/RP0/CPU0:router2#


RP/0/RP0/CPU0:router1#show controllers npu stats traps-all instance all location all | inc
 LLDP
Fri Jan 21 17:24:07.964 UTC
LLDP                                          0    22   RPLC_CPU    206   1538  6    4000
      3975       IFG     1520    0                    0
LLDP_SNOOP                                    0    28   RPLC_CPU    206   1538  6    4000
      3862       NPU     N/A     16                   0
RP/0/RP0/CPU0:router1#
```

Configuration Examples for Ethernet

This section provides the following configuration examples:

Configuring an Ethernet Interface: Example

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

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

RP/0//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
  Encapsulation ARPA,
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 LLDP: Examples

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

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

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

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

Where to Go Next

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

For information about modifying Ethernet management interfaces for the shelf controller (SC), route processor (RP), and distributed RP, see the Advanced Configuration and Modification of the Management Ethernet Interface later in this document.

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

Configuring a Layer 2 VPN AC: Example

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

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RSP0/CPU0:router(config-if)# l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol  tunnel
RP/0/RSP0/CPU0:router(config-if-l2)# commit
```

Configuring Physical Ethernet Interfaces

Use this procedure to create a basic Ethernet interface configuration.

Procedure

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 line card.

Step 2 **show interfaces** [**TenGigE FortyGigE HundredGigE FourHundredGigE**] *interface-path-id*

Example:

```
RP/0/RP0/CPU0:router# show interface HundredGigE 0/1/0/1
```

(Optional) Displays the configured interface and checks the status of each interface port.

Step 3 **configure**

Example:

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

Enters global configuration mode.

Step 4 **show interfaces** [**TenGigE FortyGigE HundredGigE FourHundredGigE**] *interface-path-id*

Example:

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

Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. Possible interface types for this procedure are:

- 10GigE
- 40GigE
- 100GigE

  Note
  - The example indicates a 100-Gigabit Ethernet interface in the line card in slot 1.

- 400GigE

The examples of *interface-path-id* ranges are:

- **TenGigE** — 0/0/0/0 - 0/0/0/31
- **FortyGigE** — 0/0/1/0 - 0/0/1/1
- **HundredGigE** — 0/0/1/0 - 0/0/1/1

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.

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 100-Gigabit Ethernet mtu values is 64 bytes to 65535 bytes.

Step 8 **no shutdown**

Example:

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

Removes the shutdown configuration, which forces an interface administratively down.

Step 9 **end** or **commit**

Example:

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

or

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

Saves configuration changes.

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

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

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

Step 10 **show interfaces** [**TenGigE FortyGigE HundredGigE FourHundredGigE**] *interface-path-id*

Example:

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

(Optional) Displays statistics for interfaces on the router.

Example

This example shows how to configure an interface for a 100-Gigabit Ethernet line card:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224

RP/0/RP0/CPU0:router(config-if)# mtu 1448

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 HundredGigE 0/5/0/24
HundredGigE0/5/0/24 is up, line protocol is up
  Interface state transitions: 1
  Hardware is HundredGigE, address is 6219.8864.e330 (bia 6219.8864.e330)
  Internet address is 3.24.1.1/24
  MTU 9216 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
     reliability 255/255, txload 3/255, rxload 3/255
  Encapsulation ARPA,
```

```
  Full-duplex, 100000Mb/s, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  Last link flapped 10:05:07
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:08:56, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 1258567000 bits/sec, 1484160 packets/sec
  5 minute output rate 1258584000 bits/sec, 1484160 packets/sec
     228290765840 packets input, 27293508436038 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 15 broadcast packets, 45 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     212467849449 packets output, 25733664696650 bytes, 0 total output drops
     Output 23 broadcast packets, 15732 multicast packets
     39 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

```
RP/0/RP0/CPU0:router# show running-config interface HundredGigE 0/5/0/24

interface HundredGigE 0/5/0/24
 mtu 9216
 service-policy input linerate
 service-policy output elinerate
 ipv4 address 3.24.1.1 255.255.255.0
 ipv6 address 3:24:1::1/64
 flow ipv4 monitor perfv4 sampler fsm ingress
!
```

Viewing Interface Counters Report

The Interface Counters report summarizes the statistics for all interfaces configured on the router.

The report displays the interfaces configured, the input and output rate, the total number of packets transmitted and received, the time interval, the current status of each interface, and the packet counts for input and output broadcast, multicast, and errored packets.

The **show interfaces** command, displays statistics per interface with many lines of data. The traffic rate displays the average number of packets received per second over the load interval. The load interval is configurable on the physical and bundle main interface. The report displays the load on the interface for a longer duration of time and does not show a spike in the traffic rate. This rate is the exponentially weighted average with a time constant of the load interval.

Note For the average to be within two percent of the instantaneous rate of a uniform stream of traffic, four times the load interval must pass.

For more information about the use of **show interfaces** command, see *Interface and Hardware Component Command Reference for Cisco 8000 Series Routers*.

Instant Display of Traffic Rates for all the Physical Interfaces

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
Instant display of traffic rates on all physical interfaces	Release 7.5.4	You can now display a snapshot of the traffic throughput and traffic rate on all physical interfaces over the last few seconds. We have introduced a **show** command to view the counters and rate information for the interfaces. The feature introduces these: • CLI: **show interfaces counter rates physical** • YANG Data Model: **Cisco-IOS-XR-infra-statsd-oper.yang** (see GitHub under the 754 folder.)

The new **show** command displays a snapshot of statistics for all the interfaces at a given instant for your quick reference. Here, the display is in a tabular format for easy analysis.

Run the **show interfaces counter rates physical** command to view statistics of all physical interfaces.

View the statistics

```
Router#show interfaces counters rates physical

    InterfaceName          Intvl    InMbps    InBW%   InKpps   OutMbps   OutBW%    OutKpps
    GigabitEthernet0/2/0/0  0:05     0.0      0.0%     0.0       0.0      0.0%        0.0
    GigabitEthernet0/2/0/1  0:05     0.0      0.0%     0.0       0.0      0.0%        0.0
    GigabitEthernet0/2/0/2  0:05     0.0      0.0%     0.0       0.0      0.0%        0.0
    GigabitEthernet0/2/0/3  0:05   235.0     22.0%    23.5      87.0      9.5%        7.2
    GigabitEthernet0/3/0/0  0:05    88.0      9.3%     7.0     100.0     10.0%       10.5
    GigabitEthernet0/3/0/1  0:05     0.0      0.0%     0.0       0.0      0.0%        0.0
```

The statistics for each physical interface is calculated for the time interval of 5 sec. Hence, the input and output rate (in Mbps and Kpps) is the real-time statistics.

Note The traffic rate displayed is the real-time link utilization of the time interval. The time interval is determined by the system and may vary based on the system processing load. The time interval increases during events where the system is handling, for example, performing routing updates.

How to Configure Interfaces in Breakout Mode

Information About Breakout

The router supports transmission of traffic in the breakout mode. The breakout mode enables a 40 Gigabit Ethernet port to be split into four independent and logical 10 Gigabit Ethernet ports. The 4x10 breakout mode is supported on the following types of 40G modules:

- QSFP-4x10-LR-S
- QSFP-40G-SR4

Guidelines and Restrictions for Breakout Mode

- The native 40G mode on QSFP-40G-SR4 is not supported.
- The 36-port QSFP56-DD 400 GbE Line Card does not support the 4x10G breakout.
- If you're using a Q100-based Cisco 8200 Series Router and want to set up a 4x10G breakout configuration, you need to use even numbered ports from 24 to 35. These include ports 24, 26, 28, 30, 32, and 34. Once you do this, the system automatically disables the odd numbered ports in this range - ports 25, 27, 29, 31, 33, and 35.
- Use the *hw-module port-range* command to set the port range for the breakout configuration in the global configuration.
- To remove the global *hw-module port-range* configuration, you must first remove the 'breakout 4x10' configuration under the controller.
- For 4x10G breakout on 48-port Line Card, only QSFP-4x10-LR-S module is supported.
- For the 88-LC0-34H14FH line card, you must breakout only 3 ports instead of 4 ports to avoid the `QOS-DPA_QOSEA-2-TMPORT_PROG_ERROR` issue, which creates partial interfaces during configuration mode.

Configure Breakout in a Port

Configuring breakout in a port:

```
RP/0/RP0/CPU0:uut# configure
Fri Oct 11 23:58:47.165 UTC
RP/0/RP0/CPU0:uut(config)# controller optics 0/1/0/28
RP/0/RP0/CPU0:uut(config-Optics)# breakout 4x10
RP/0/RP0/CPU0:uut(config-Optics)# commit
Fri Oct 11 23:59:51.261 UTC
RP/0/RP0/CPU0:uut(config-Optics)# end
RP/0/RP0/CPU0:uut#
```

Remove the Breakout Configuration

Removing the breakout configuration:

```
RP/0/RP0/CPU0:uut# configure
Sat Oct 12 00:01:38.673 UTC
RP/0/RP0/CPU0:uut(config)# controller optics 0/1/0/28
RP/0/RP0/CPU0:uut(config-Optics)# no breakout 4x10
RP/0/RP0/CPU0:uut(config-Optics)# commit
Sat Oct 12 00:01:55.864 UTC
RP/0/RP0/CPU0:uut(config-Optics)# end
```

Verify a Breakout Configuration

Verifying a breakout configuration:

```
RP/0/RP0/CPU0:uut# show running-config controller optics 0/1/0/28
Sat Oct 12 00:11:33.962 UTC
controller Optics0/1/0/28
breakout 4x10
!

RP/0/RP0/CPU0:uut# show int br location 0/1/CPU0 | i Te
Sat Oct 12 00:11:38.609 UTC
       Te0/1/0/27/0          up          up               ARPA 10000   10000000
       Te0/1/0/27/1          up          up               ARPA 10000   10000000
       Te0/1/0/27/2          up          up               ARPA 10000   10000000
       Te0/1/0/27/3          up          up               ARPA 10000   10000000
       Te0/1/0/28/0          up          up               ARPA 10000   10000000
       Te0/1/0/28/1          up          up               ARPA 10000   10000000
       Te0/1/0/28/2          up          up               ARPA 10000   10000000
       Te0/1/0/28/3          up          up               ARPA 10000   10000000

.
```

Ethernet Interface Route Statistics

Table 13: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet Interface Route Statistics	Release 7.3.4	You can view statistics on the number of packets and bytes sent and received in unicast, multicast, and broadcast routes. These statistics help you to monitor the network performance and measure your bandwidth.

Ethernet interface route statistics provide the following information about the unicast, multicast, and broadcast routes:

- The number of packets or bytes received and transmitted.
- The total number of packets or bytes passing through the Ethernet interface.

These statistics are available on Cisco 8000 routers that are built with Cisco Silicon One Q100 and Q200 processors and all Network Processor Unit (NPU) 2.0 devices.

Ethernet interface route statistics may be useful for monitoring the network devices and their traffic. For example, if you are not able to connect to the internet or use some cloud-based applications, these route statistics can help you understand the problems in the network and where they occur.

Viewing the Interface Statistics

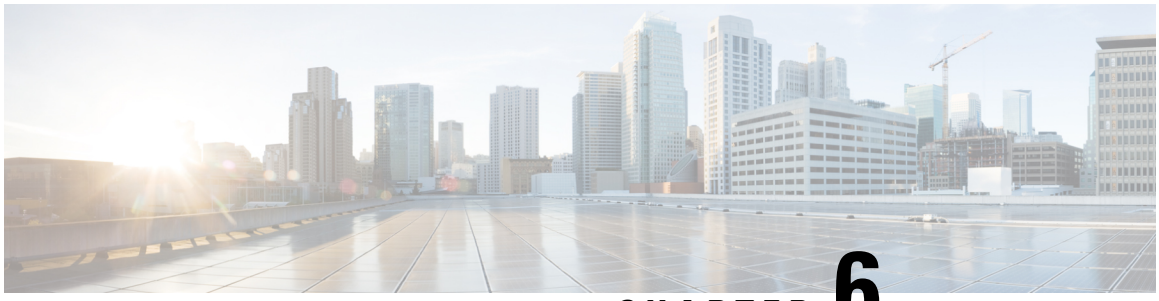
Use the `show interface` and the `show controller interface` commands to view these Ethernet interface route statistics. The following is a sample showing both commands.

```
Router#show interfaces HundredGigE 0/0/0/0

<Timestamp>

HundredGigE0/0/0/0 is up, line protocol is up

  Interface state transitions: 93

  Hardware is HundredGigE, address is acbc.d975.0500 (bia acbc.d975.0500)

  Internet address is 100.0.1.1/24

  MTU 1514 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)

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

  Encapsulation ARPA,

  Full-duplex, 100000Mb/s, 100GBASE-SR4, link type is force-up

  output flow control is off, input flow control is off

  Carrier delay (up) is 10 msec

  loopback not set,

  Last link flapped 01:03:01

  ARP type ARPA, ARP timeout 04:00:00

  Last input 3d09h, output 01:02:41

  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
```

```
     2959131434 packets input, 757537646912 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

     2958525230 packets output, 757382468319 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

     93 carrier transitions
```

```
Router#show controllers HundredGigE0/0/0/0 stats

<Timestamp>

Statistics for interface HundredGigE0/0/0/0 (cached values):



Ingress:

    Input total bytes           = 757537646912

    Input good bytes            = 757537646912


    Input total packets         = 2959131434
    Input 802.1Q frames         = 0
    Input pause frames          = 0
    Input pkts 64 bytes         = 1
    Input pkts 65-127 bytes     = 0
    Input pkts 128-255 bytes    = 0
    Input pkts 256-511 bytes    = 2959131433
    Input pkts 512-1023 bytes   = 0
    Input pkts 1024-1518 bytes  = 0
    Input pkts 1519-Max bytes   = 0

    Input good pkts             = 2959131434
```

```
    Input unicast pkts          = 0
    Input multicast pkts        = 0
    Input broadcast pkts        = 0

    Input drop overrun          = 0
    Input drop abort            = 0
    Input drop invalid VLAN     = 0
    Input drop invalid DMAC     = 0
    Input drop invalid encap    = 0
    Input drop other            = 0

    Input error giant           = 0
    Input error runt            = 0
    Input error jabbers         = 0
    Input error fragments       = 0
    Input error CRC             = 0
    Input error collisions      = 0
    Input error symbol          = 0
    Input error other           = 0

    Input MIB giant             = 0
    Input MIB jabber            = 0
    Input MIB CRC               = 0

Egress:
    Output total bytes          = 757382468319
    Output good bytes           = 757382468319

    Output total packets        = 2958525230
    Output 802.1Q frames        = 0
    Output pause frames         = 0
    Output pkts 64 bytes        = 41
    Output pkts 65-127 bytes    = 296
    Output pkts 128-255 bytes   = 746
    Output pkts 256-511 bytes   = 2958524147
    Output pkts 512-1023 bytes  = 0
    Output pkts 1024-1518 bytes = 0
    Output pkts 1519-Max bytes  = 0

    Output good pkts            = 2958525230
    Output unicast pkts         = 0
    Output multicast pkts       = 0
    Output broadcast pkts       = 0

    Output drop underrun        = 0
    Output drop abort           = 0
    Output drop other           = 0

    Output error other          = 0
```



CHAPTER 6

Configuring Ethernet OAM

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

Table 14: Feature Information Table

Release	Modification
Release 7.3.1	Support for Ethernet Link OAM was introduced.

Information About Configuring Ethernet OAM

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

Ethernet Link OAM

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet Link OAM	Release 7.3.1	This feature allows service providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events, and take actions on events. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet 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, and take actions on events. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet link OAM can be configured in the following ways:

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

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

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

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

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

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

Neighbor Discovery

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

EFD

Ethernet Fault Detection (EFD) is a mechanism that allows Ethernet OAM protocols to control the `line protocol` state of an interface.

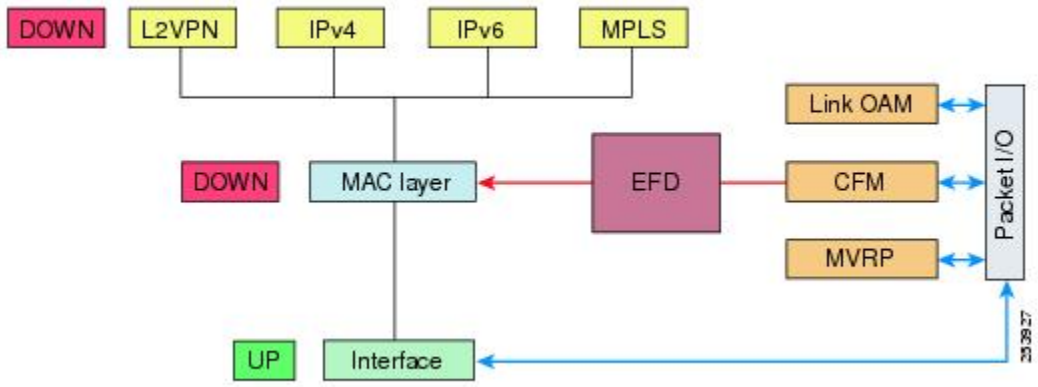
Unlike many other interface types, Ethernet interfaces do not have a line protocol, whose state is independent from that of the interface. For Ethernet interfaces, this role is handled by the physical-layer Ethernet protocol itself, and therefore if the interface is physically up, then it is available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

EFD changes this to allow EOAM to act as the line protocol for Ethernet interfaces. This allows EOAM to control the interface state so that if a EOAM defect (such as AIS or loss of continuity) is detected with an expected peer MEP, the interface can be shut down. This not only stops traffic flow, but also triggers actions in any higher-level protocols to route around the problem. For example, in the case of Layer 2 interfaces, the MAC table would be cleared and MSTP would reconverge. For Layer 3 interfaces, the ARP cache would be cleared and potentially the IGP would reconverge.

Note EFD can only be used for down MEPs. When EFD is used to shut down the interface, the EOAM frames continue to flow. This allows EOAM to detect when the problem has been resolved, and thus bring the interface backup automatically.

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

Figure 1: EOAM Error Detection and EFD Trigger



MIB Retrieval

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

Miswiring Detection (Cisco-Proprietary)

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

SNMP Traps

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

Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

Configuring Ethernet OAM Features on an Individual Interface: Example

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

```
configure
 interface TenGigE 0/1/0/0
  ethernet oam
  connection timeout 30
  mib-retrieval
  link-monitor
   frame window milliseconds 60000
   frame threshold low 10000000 high 60000000
   frame-period window milliseconds 60000
   frame-period threshold ppm low 100 high 120000
   frame-seconds window milliseconds 900000
   frame-seconds threshold low 3 high 900
   symbol-period window milliseconds 60000
   symbol-period threshold ppm low 1000000 high 1000000
  exit
  require-remote
   mode active
   mib-retrieval
  exit
  action
   critical-event error-disable-interface
   dying-gasp error-disable-interface
   capabilities-conflict error-disable-interface
   wiring-conflict error-disable-interface
   discovery-timeout error-disable-interface
   session-down error-disable-interface
   commit
```

Configuring an Ethernet OAM Profile Globally: Example

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

```
configure
ethernet oam profile Profile_1
 connection timeout 30
 mib-retrieval
 link-monitor
  frame window milliseconds 60000
```

```
  frame threshold low 10000000 high 60000000
  frame-period window milliseconds 60000
  frame-period threshold ppm low 100 high 1000000
  frame-seconds window milliseconds 900000
  frame-seconds threshold low 3 high 900
  symbol-period window milliseconds 60000
  symbol-period threshold ppm low 100000 high 1000000
 exit
 require-remote
  mode active
  mib-retrieval
 exit
 action
  critical-event error-disable-interface
  dying-gasp error-disable-interface
  capabilities-conflict error-disable-interface
  wiring-conflict error-disable-interface
  discovery-timeout error-disable-interface
  session-down error-disable-interface
  commit
```

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

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

```
configure
 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

    commit

configure
 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

      commit
```

Recovering from error-disable: Example

You can recover an error-disabled interface due to session-down using one of these methods:

- Manually clear the error-disable using the **clear** command.

```
Router# configure
Router(config)# ethernet oam profile Profile_1
Router(config-eoam)# action
Router(config-eoam-action)# clear session-down error-disable-interface
```

- Disable and then re-enable the network link using administrative shutdown commands to reset the connection.

```
Router# configure
Router(config)# interface TenGigE 0/1/0/0
Router(config-if)# shutdown
Router(config-if)# commit
Router(config-if)# no shutdown
Router(config-if)# commit
```

- Configure an auto-recovery timer for this error-disable reason.

```
Router# configure
Router(config)# error-disable recovery cause link-oam-session-down interval 30
Router(config)# commit
```

Clearing Ethernet OAM Statistics on an Interface: Example

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

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

Enabling SNMP Server Traps on a Router: Example

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

```
configure
 snmp-server traps ethernet oam events
```

Ethernet CFM

Table 16: Feature History Table

Feature name	Release	Description
CFM on bundle member link for connectivity check	Release 7.3.15	This feature introduces support for Connectivity Fault Management (CFM) on bundle members. Earlier, network administrators managed networks by using the fault, configuration, account, performance, security model. CFM is one of a suite of the Ethernet OAM protocols, which uses a combination of keepalive packets and MAC-based pings, and traceroutes to detect faults in a network. With the CFM feature, you: • reduce operating expenses for service operators by reducing network faults and errors • provide end-to-end maintenance of networks
Up MEP and down MEP support in CFM	Release 7.3.15	This feature introduces Maintenance End Points (MEP) entities that you can configure in a domain. MEPs send either CFM frames from the interface where they are configured or CFM frames that are received on other interfaces. MEPs allow you to perform fault management and carry out performance checks.

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

CFM is defined in two standards:

- IEEE 802.1ag—Defines the core features of the CFM protocol.

- ITU-T Y.1731—Redefines, but maintains compatibility with the features of IEEE 802.1ag, and defines some additional features.

Ethernet CFM supports these functions of ITU-T Y.1731:

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

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

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

Limitations and restrictions

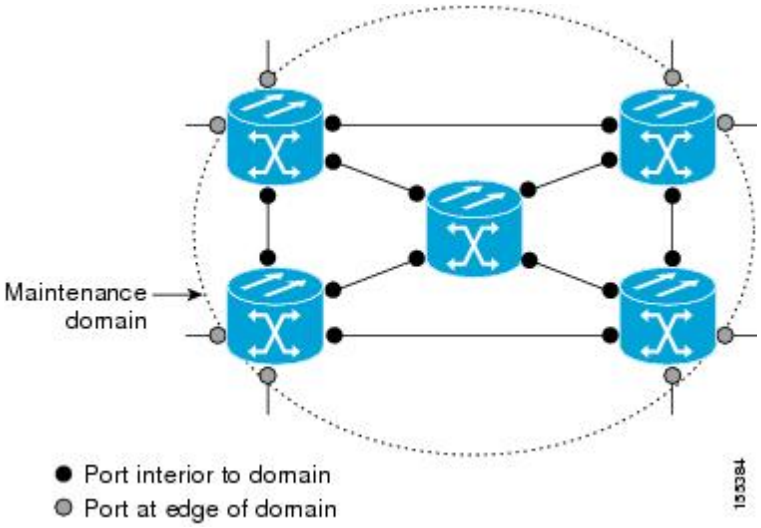
- The system supports only cross-connect.
- MIPs are not supported.
- Supports timer of 1s, 10s, 1m, 10m.
- Supports timer of 100ms, 1s, 10s, 1m, 10m for bundle members.
- L3 interfaces are not supported except for bundle members.
- Down MEPs are only supported for L2 cross-connect and bundle members.
- Multiple MEPs of different directions are not supported on the same interface or Xconnect.
- CFM is not supported on L2 subinterfaces with default encapsulation.
- When configuring CFM down MEP on an interface, ensure that the interface is included in an L2VPN.

Maintenance Domains

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

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

Figure 2: CFM Maintenance Domain



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

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

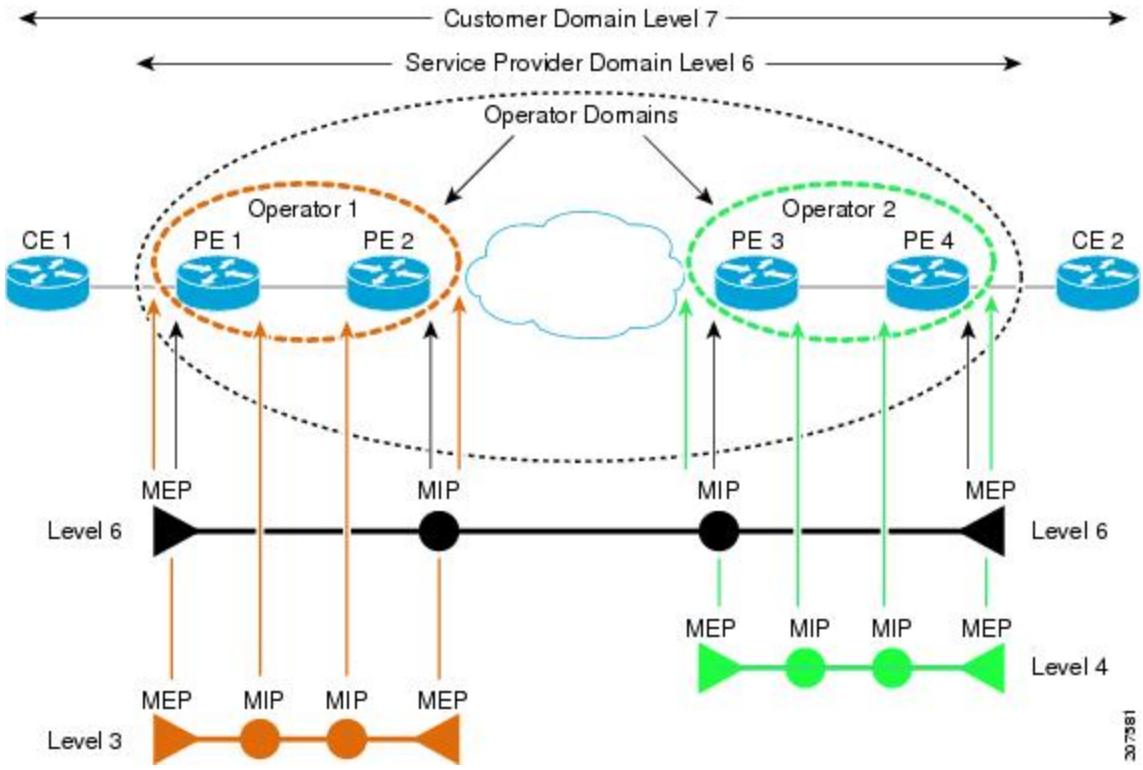
- The customer can use CFM between their CE devices, to verify and manage connectivity across the whole network.
- The service provider can use CFM between their PE devices, to verify and manage the services they are providing.
- Each operator can use CFM within their operator network, to verify and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

Each organization uses a different CFM maintenance domain.

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

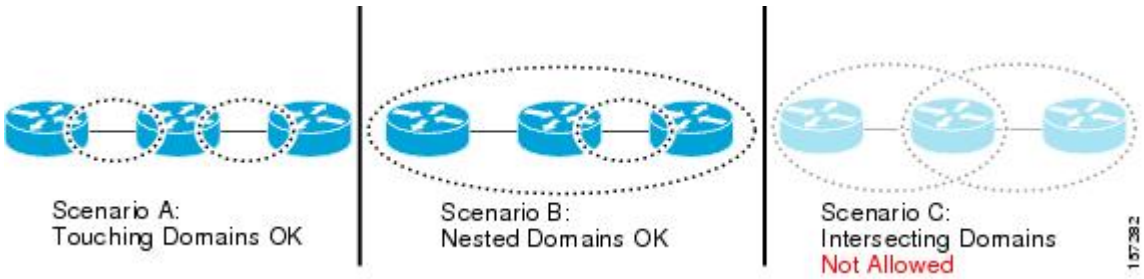
Note In CFM diagrams, the conventions are that triangles represent MEPs, pointing in the direction that the MEP sends CFM frames, and circles represent MIPs.

Figure 3: Different CFM Maintenance Domains Across a Network



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

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



Services

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

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

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

Maintenance Points

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

A maintenance point is always associated with a particular CFM service, and therefore with a particular maintenance domain at a particular level. Maintenance points generally only process CFM frames at the same level as their associated maintenance domain. Frames at a higher maintenance level are always forwarded transparently, while frames at a lower maintenance level are normally dropped. This helps enforce the maintenance domain hierarchy, and ensures that CFM frames for a particular domain cannot leak out beyond the boundary of the domain.

There are following type(s) of MP(s):

- Maintenance End Points (MEPs)—Created at the edge of the domain. Maintenance end points (MEPs) are members of a particular service within a domain and are responsible for sourcing and sinking CFM frames. They periodically transmit continuity check messages and receive similar messages from other MEPs within their domain. They also transmit traceroute and loopback messages at the request of the administrator. MEPs are responsible for confining CFM messages within the domain.

MEP and CFM Processing Overview

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

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

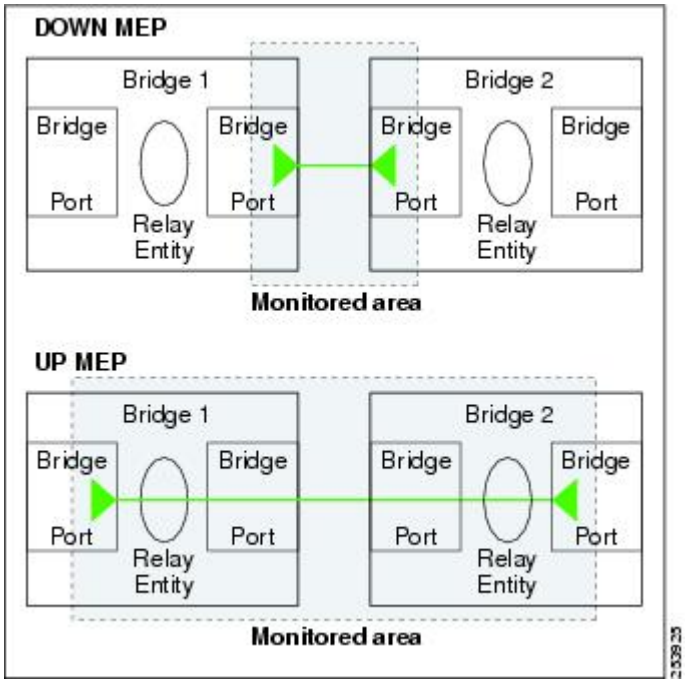
AIS packets are only sent when there is a MIP configured on the same interface as the MEP and at the level of the MIP.

Note

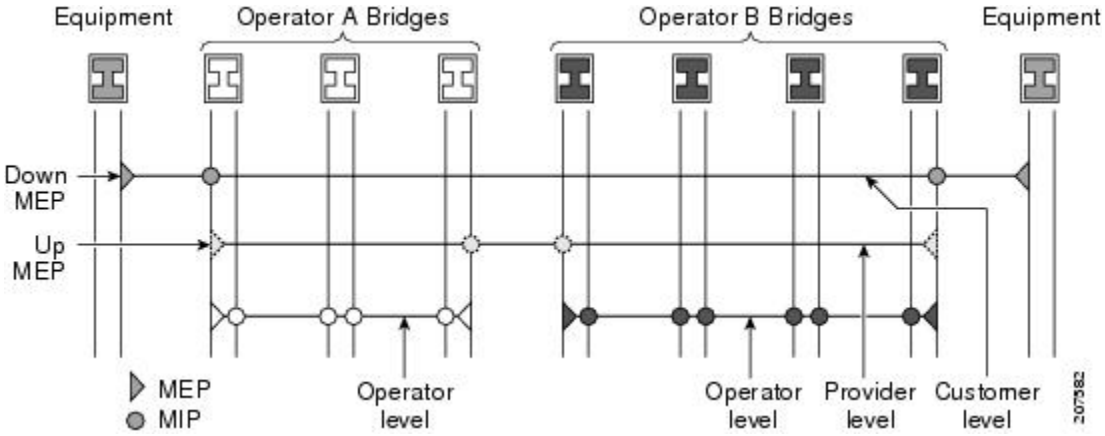
- The terms *Down MEP* and *Up MEP* are defined in the IEEE 802.1ag and ITU-T Y.1731 standards, and refer to the direction that CFM frames are sent from the MEP. The terms should not be confused with the operational status of the MEP.
- The router only supports the "Down MEP level < Up MEP level" configuration.

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

Figure 4: Monitored Areas for Down and Up MEPs



This figure shows maintenance points at different levels. Because domains are allowed to nest but not intersect, a MEP at a low level often corresponds with a MEP at a higher level.



Up MEPs can only exist on switched (Layer 2) interfaces, because they send and receive frames from the bridge relay function. Down MEPs can be created on switched (Layer 2) interfaces.

MEPs continue to operate normally if the interface they are created on is blocked by the Spanning Tree Protocol (STP); that is, CFM frames at the level of the MEP continue to be sent and received, according to the direction of the MEP. MEPs never allow CFM frames at the level of the MEP to be forwarded, so the STP block is maintained.

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

CFM Protocol Messages

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

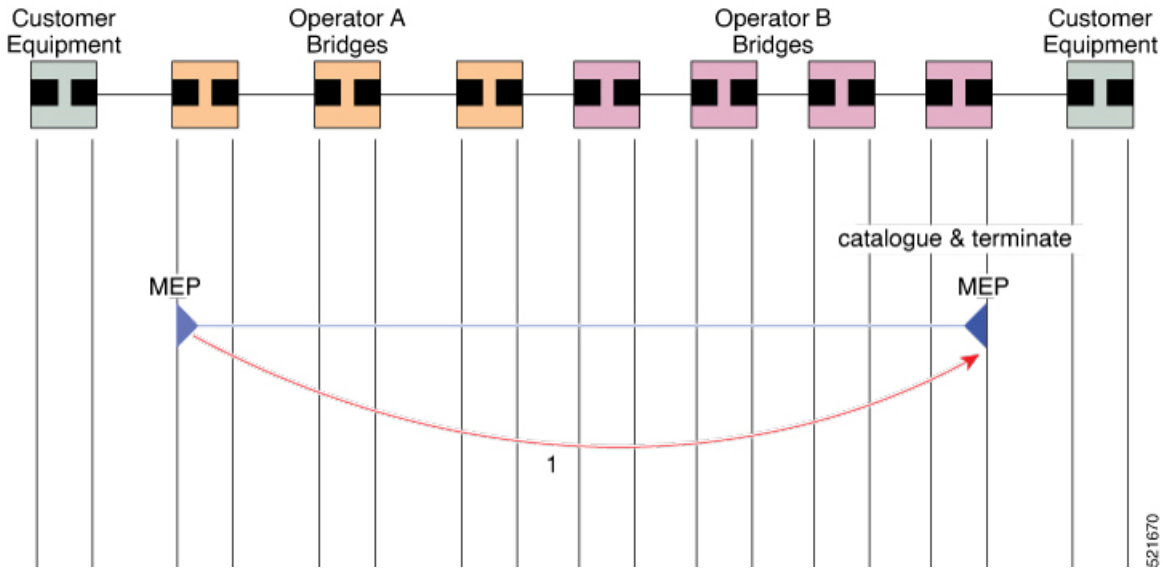
This section describes the following CFM messages:

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

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

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

Figure 5: Continuity Check Message Flow



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

- 100 ms (only supported on bundle members)
- 1 s
- 10 s
- 1 minute
- 10 minutes

A MEP detects a loss of connectivity with one of its peer MEPs when some number of CCMs are missed. This occurs when sufficient time has passed during which a certain number of CCMs were expected, given the CCM interval. This number is called the *loss threshold*, and is usually set to 3.

With the exception of bundle members, CFM is supported only on interfaces that have Layer 2 transport feature enabled.

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

- A configured identifier for the domain of the transmitting MEP. This is referred to as the Maintenance Domain Identifier (MDID).
- A configured identifier for the service of the transmitting MEP. This is referred to as the Short MA Name (SMAN). Together, the MDID and the SMAN make up the Maintenance Association Identifier (MAID). The MAID must be configured identically on every MEP in the service.
- These are restrictions on the type of MAID that are supported for sessions with time interval of less than 1 minute. The MAID supports two types of formats on offloaded MEPs:
    - No Domain Name Format
        - MD Name Format = 1-NoDomainName

- Short MA Name Format = 3 - 2 bytes integer value
- Short MA NAme Length = 2 - fixed length
- Short MA Name = 2 bytes of integer

- 1731 Maid Format
  - MD Name Format = 1-NoDomainName
  - MA Name Format(MEGID Format) = 32
  - MEGID Length = 13 - fixed length
  - MEGID(ICCCode) = 6 Bytes
  - MEGID(UMC) = 7 Bytes
  - ITU Carrier Code (ICC) - Number of different configurable ICC code - 15 (for each NPU)
  - Unique MEG ID Code (UMC) - 4

Maintenance Association Identifier (MAID) comprises of the Maintenance Domain Identifier (MDID) and Short MA Name (SMAN). MDID only supports **null** value and SMAN only supports ITU Carrier Code (ICC) or a numerical. No other values are supported.

- An example for configuring domain ID null is: **ethernet cfm domain SMB level 3 id null**
- An example for configuring SMAN is: **ethernet cfm domain SMB level 3 id null service 901234AB xconnect group 99999 p2p 99999 id number 1**

- A configured numeric identifier for the MEP (the MEP ID). Each MEP in the service must be configured with a different MEP ID.
- Dynamic Remote MEPs are not supported for MEPs with less than 1 min interval. You must configure MEP CrossCheck for all such MEPs.
- Sequence numbering is not supported for MEPs with less than 1 minute interval.
- In a Remote Defect Indication (RDI), each MEP includes this in the CCMs it is sending, if it has detected a defect relating to the CCMs it is receiving. This notifies all the MEPs in the service that a defect has been detected somewhere in the service.
- The interval at which CCMs are being transmitted.
- CCM Tx/Rx statistics counters are not supported for MEPs with less than1 minute intervals.
- Sender TLV and Cisco Proprietary TLVs are not supported for MEPs with less than 1 minute intervals.
- The status of the interface where the MEP is operating, for example, whether the interface is up, down, STP blocked, and so on.

Note The status of the interface (up/down) should not be confused with the direction of any MEPs on the interface (Up MEPs/Down MEPs).

These defects can be detected from the received CCMs:

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

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

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

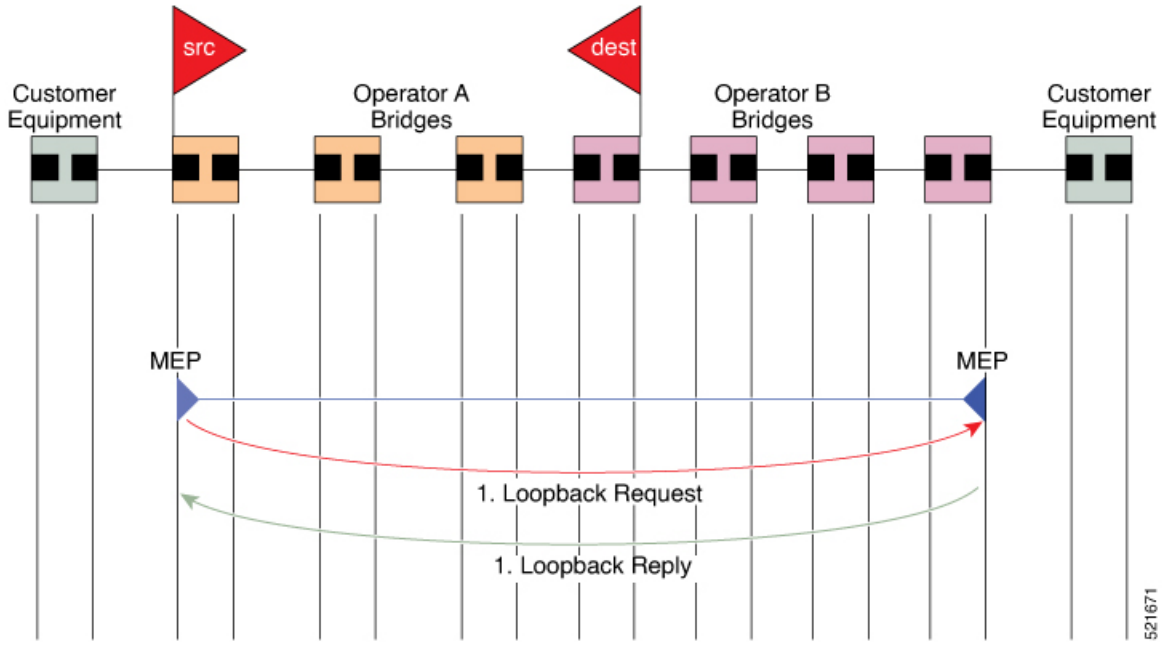
Loopback (IEEE 802.1ag and ITU-T Y.1731)

Loopback Messages (LBM) and Loopback Replies (LBR) are used to verify connectivity between a local MEP and a particular remote MP. At the request of the administrator, a local MEP sends unicast LBMs to the remote MP.

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

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

Figure 6: Loopback Messages



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

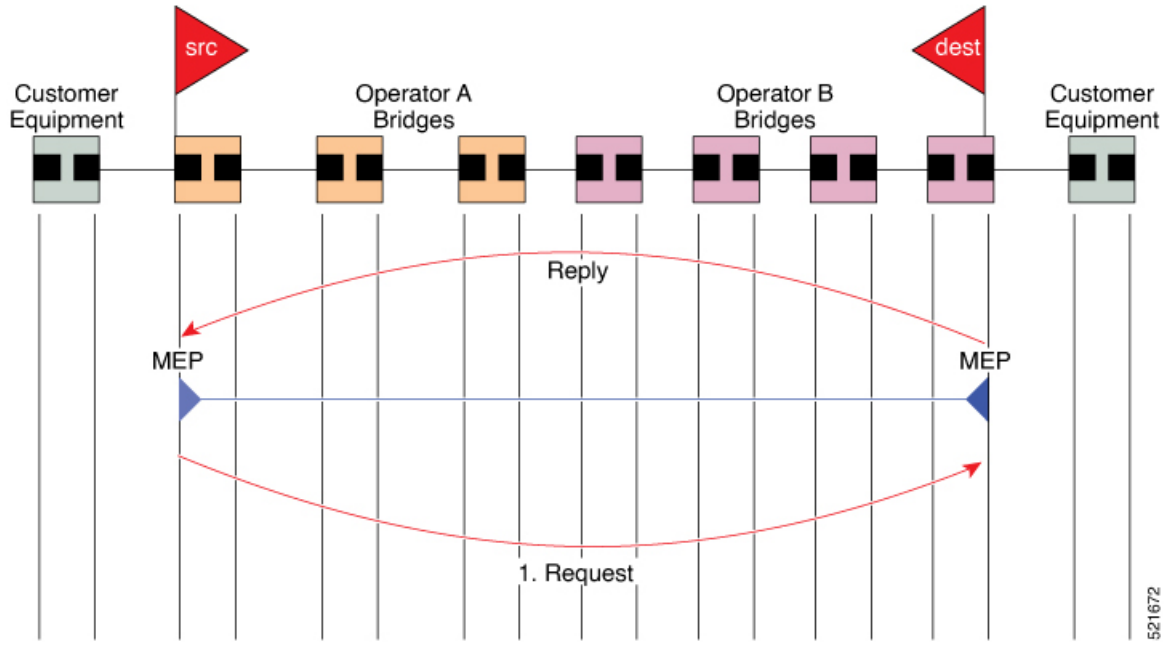
Linktrace (IEEE 802.1ag and ITU-T Y.1731)

Linktrace Messages (LTM) and Linktrace Replies (LTR) are used to track the path (hop-by-hop) to a unicast destination MAC address.

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

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

Figure 7: Linktrace Message Flow



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

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

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

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

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

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

Configurable Logging

CFM supports logging of various conditions to syslog. Logging can be enabled independently for each service, and when the following conditions occur:

- New peer MEPs are detected, or loss of continuity with a peer MEP occurs.
- Changes to the CCM defect conditions are detected.
- Cross-check “missing” or “unexpected” conditions are detected.
- AIS condition detected (AIS messages received) or cleared (AIS messages no longer received).
- EFD used to shut down an interface, or bring it back up.

How to Configure Ethernet OAM

This section provides these configuration procedures:

Configuring Ethernet OAM

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

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

Configuring an Ethernet OAM Profile

Perform these steps to configure an Ethernet OAM profile.

Procedure

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

	Command or Action	Purpose
Step 2	**ethernet oam profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config)# ethernet oam profile Profile_1`	Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.
Step 3	**link-monitor** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# link-monitor`	Enters the Ethernet OAM link monitor configuration mode.
Step 4	**symbol-period window** *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 low 10000000 high 60000000`	(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		
Step 7	**frame window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame window 60`	(Optional) Configures the frame window size (in milliseconds) of an OAM frame error event. The range is from 1000 to 60000. The default value is 1000.
Step 8	**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 9	**frame-period window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window 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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window milliseconds 60000`	interface speed. Note that the conversion assumes that all frames are of the minimum size. The range is from 100 to 60000. The default value is 1000. **Note** The only accepted values are multiples of the line card-specific polling interval, that is, 1000 milliseconds for most line cards.
Step 10	**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 0 to 1000000. The default low threshold is 1. To obtain the number of frames, the configured time interval is converted to a window size in frames using the interface speed. For example, for a 1Gbps interface, the IEEE defines minimum frame size as 512 bits. So, we get a maximum of approximately 1.5 million frames per second. If the window size is configured to be 8 seconds (8000ms) then this would give us a Window of 12 million frames in the specification's definition of Errored Frame Window. The thresholds for frame-period are measured in errors per million frames. Hence, if you configure a window of 8000ms (that is a window of 12 million frames) and a high threshold of 100, then the threshold would be crossed if there are 1200 errored frames in that period (that is, 100 per million for 12 million).
Step 11	**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 6000. **Note** The only accepted values are multiples of the line card-specific polling interval, that is, 1000 milliseconds for most line cards.

	Command or Action	Purpose
Step 12	**frame-seconds threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 threshold 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 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# exit`	Exits back to Ethernet OAM mode.
Step 14	**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 15	**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 16	**hello-interval 1s** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# hello-interval 1s`	Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (**1s**).
Step 17	**mode** {**active**|**passive**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mode passive`	Configures the Ethernet OAM mode. The default is active.
Step 18	**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 19	**require-remote mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval`	Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.

	Command or Action	Purpose
Step 20	**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 21	**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 22	**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 23	**action dying-gasp** {**disable** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface`	Specifies the action that is taken on an interface when a dying-gasp notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 24	**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

	Command or Action	Purpose
		mode to override the profile setting and take no action at the interface when the event occurs.
Step 25	**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 26	**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 27	**action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**}	Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 28	**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 29	**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.

	Command or Action	Purpose
Step 30	**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 31	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Attaching an Ethernet OAM Profile to an Interface

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

Procedure

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

	Command or Action	Purpose
Step 6	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

Verifying the Ethernet OAM Configuration

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

```
RP/0/RP0/CPU0:router# show ethernet oam configuration
Thu Aug  5 22:07:06.870 DST
GigabitEthernet0/4/0/0:
  Hello interval:                                          1s
  Mib retrieval enabled:                                    N
  Uni-directional link-fault detection enabled:             N
  Configured mode:                                     Active
  Connection timeout:                                       5
  Symbol period window:                                     0
  Symbol period low threshold:                              1
  Symbol period high threshold:                          None
  Frame window:                                          1000
  Frame low threshold:                                      1
  Frame high threshold:                                  None
  Frame period window:                                   1000
  Frame period low threshold:                               1
  Frame period high threshold:                           None
```

```
Frame seconds window:                                60000
Frame seconds low threshold:                             1
Frame seconds high threshold:                         None
High threshold action:                                None
Link fault action:                                     Log
Dying gasp action:                                     Log
Critical event action:                                 Log
Discovery timeout action:                              Log
Capabilities conflict action:                          Log
Wiring conflict action:                      Error-Disable
Session up action:                                     Log
Session down action:                                   Log
Require remote mode:                                Ignore
Require remote MIB retrieval:                            N
```

Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:

Note CFM is not supported for the following:

- L3 Interfaces and Sub-Interfaces
- Bridge Domain, Release 7.3.1 and earlier
- VPLS, Release 7.3.1 and earlier

Configuring a CFM Maintenance Domain

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

Configuring services for a CFM maintenance domain

You can configure up to 50 CFM sessions per line card or 50 CFM sessions per fixed-port router. The system supports 50 CFM sessions on bundles.

Starting Cisco IOS XR Release 7.3.2 and later, 100 CFM sessions are supported for every system.

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

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*]
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations at a specified maintenance level, and enters CFM domain configuration mode. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps	xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*]	[[**number** *number*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs. The **id** sets the short MA name.
Step 5	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before`

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

Enabling and Configuring Continuity Check for a CFM Service

To configure Continuity Check for a CFM service, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*]
5. **continuity-check interval** *time* [**loss-threshold** *threshold*]
6. **continuity-check archive hold-time** *minutes*
7. **continuity-check loss auto-traceroute**
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:**	Enters Ethernet Connectivity Fault Management (CFM) configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# ethernet cfm`	
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with a xconnect where up MEPs will be created. The **id** sets the short MA name.
Step 5	**continuity-check interval** *time* [**loss-threshold** *threshold*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10`	(Optional) Enables Continuity Check and specifies the time interval at which CCMs are transmitted or to set the threshold limit for when a MEP is declared down.
Step 6	**continuity-check archive hold-time** *minutes* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100`	(Optional) Configures how long information about peer MEPs is stored after they have timed out.
Step 7	**continuity-check loss auto-traceroute** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute`	(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.
Step 8	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

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

Configuring Cross-Check on a MEP for a CFM Service

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router# ethernet cfm`	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]	Creates and names a container for all domain configurations and enters the CFM domain configuration mode.

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with a xconnect where up MEPs will be created. The **id** sets the short MA name.
Step 5	**mep crosscheck** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10`	Enters CFM MEP crosscheck configuration mode.
Step 6	**mep-id** *mep-id-number mep-id-number* [**mac-address** *mac-address*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10`	Enables cross-check on a MEP. **Note** • Repeat this command for every MEP that you want included in the expected set of MEPs for cross-check.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Other Options for a CFM Service

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router# ethernet cfm`	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps	xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*]	[**string** *text*]	[**number** *number*]	[**vlan-id** *id-number*]	[**vpn-id** *oui-vpnid*]]	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where up MEPs will be created. The **id** sets the short MA name.

	Command or Action	Purpose
Step 5	**maximum-meps** *number* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000`	(Optional) Configures the maximum number (2 to 8190) of MEPs across the network, which limits the number of peer MEPs recorded in the database.
Step 6	**log** {**ais**|**continuity-check errors**|**continuity-check mep changes**|**crosscheck errors**|**efd**} **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors`	(Optional) Enables logging of certain types of events.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring CFM MEPs

- For every subinterface configured under a Layer 3 parent interface, you must associate a unique 802.1Q or 802.1ad tag. Else, it leads to unknown network behavior.

SUMMARY STEPS

1. **configure**
2. **interface** {**HundredGigE | TenGigE**} *interface-path-id*
3. **interface** {**HundredGigE | TenGigE | Bundle-Ether**} *interface-path-idl2transport*
4. **ethernet cfm**
5. **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*

6. **cos** *cos*
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** {**HundredGigE	TenGigE**} *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1`	Type of Ethernet interface on which you want to create a MEP. Enter **HundredGigE** or **TenGigE** and the physical interface or virtual interface. **Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router. • L3 interfaces are only supported for bundle member interfaces. Else, you must enable l2transport.
Step 3	**interface** {**HundredGigE	TenGigE	Bundle-Ether**} *interface-path-idl2transport* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1`	Type of Ethernet interface on which you want to create a MEP. Enter **HundredGigE, TenGigE,** or **Bundle-Ether** and the physical interface or virtual interface followed by the l2transport. L2transport configures the interface as an L2 interface. Naming convention is *interface-path-id.subinterface*. The period in front of the subinterface value is required as part of the notation.
Step 4	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet cfm`	Enters interface Ethernet CFM configuration mode.
Step 5	**mep domain** *domain-name* **service** *service-name* **mep-id** *id-number* **Example:** `RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1`	Creates a maintenance end point (MEP) on an interface and enters interface CFM MEP configuration mode.
Step 6	**cos** *cos* **Example:** `RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7`	(Optional) Configures the class of service (CoS) (from 0 to 7) for all CFM packets generated by the MEP on an interface. If not configured, the CoS is inherited from the Ethernet interface.

	Command or Action	Purpose
		Note For Ethernet interfaces, the CoS is carried as a field in the VLAN tag. Therefore, CoS only applies to interfaces where packets are sent with VLAN tags. If the **cos (CFM)** command is executed for a MEP on an interface that does not have a VLAN encapsulation configured, it will be ignored.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Y.1731 AIS

This section has the following step procedures:

Configuring AIS in a CFM Domain Service

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM global configuration mode.
Step 3	**domain** *name* **level** *level* **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1`	Specifies the domain and domain level.
Step 4	**service** *name* **xconnect group** *xconnect-group-name* **p2p** *xconnect-name* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 xconnect group XG1 p2p X2`	Specifies the service and cross-connect group and name.
Step 5	**ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7`	Configures Alarm Indication Signal (AIS) transmission for a Connectivity Fault Management (CFM) domain service.
Step 6	**log ais** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais`	Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

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

Configuring AIS on a CFM Interface

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

SUMMARY STEPS

1. **configure**
2. **interface gigabitethernet** *interface-path-id*
3. **ethernet cfm**
4. **ais transmission up interval 1m cos** *cos*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface gigabitethernet** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# interface TenGigE 0/0/0/2`	Enters interface configuration mode.
Step 3	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM interface configuration mode.
Step 4	**ais transmission up interval 1m cos** *cos* **Example:**	Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7`	
Step 5	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the CFM Configuration

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

show ethernet cfm configuration-errors [**domain** *domain-name*] [**interface** *interface-path-id*]	Displays information about errors that are preventing configured CFM operations from becoming active, as well as any warnings that have occurred.
show ethernet cfm local maintenance-points domain *name* [**service** *name]* | **interface** *type interface-path-id*] [**mep** | **mip**]	Displays a list of local maintenance points.

Note After you configure CFM, the error message, *cfmd[317]: %L2-CFM-5-CCM_ERROR_CCMS_MISSED : Some received CCMs have not been counted by the CCM error counters*, may display. This error message does not have any functional impact and does not require any action from you.

CFM Over Bundles

CFM over bundle supports the following:

- CFM Maintenance Points — UP MEP, Down MEP, which only includes L2 bundle main and sub-interfaces.
- CCM interval of 100 ms, 1s, 10s, 1min, and 10mins.
- RP OIR/VM reload without impacting learnt CFM peer MEPs.
- Process restart without impacting CFM sessions.
- Static MEPs.

Restrictions for Configuration of CFM on Bundles

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

- Only Layer 2 bundle Ethernet interfaces and sub-interfaces are supported, which are part of a L2VPN cross-connect.
- No support for 3.3ms and 10ms CCM interval.
- Supports 5000 pps rates of CCM traffic for bundle interfaces.
- Ethernet Connectivity Fault Management (CFM) is not supported with Maintenance association End Points (MEPs) that are configured on default and untagged encapsulated sub-interfaces that are part of a single physical interface.
- Multiple MEPs of different directions are not supported on the same interface or Xconnect.
- CFM does not support fast failover, which may result in session flaps on bundle interfaces. Use offload for virtual interfaces to avoid flaps on faster CCM intervals.

Ethernet SLA Statistics Measurement in a Profile

Table 17: Feature History Table

Feature Name	Release Information	Feature Description
Enhancement to Ethernet SLA Statistics Measurement	Release 7.7.1	You can now configure the size of bins for the delay and jitter measurement in Ethernet SLA statistics with a **width** value ranging from 1 to 10000000 microseconds. This enhancement provides granularity to store more accurate results of SLA statistics in the aggregate bins. In earlier releases, you could only configure the **width** value for the delay and jitter measurement in milliseconds. This feature introduces the **usec** keyword in the **aggregate** command.

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

Ethernet SLA statistics measurement for network performance is performed by sending packets and storing data metrics such as:

- Round-trip delay time—The time for a packet to travel from source to destination and back to source again.
- Round-trip jitter—The variance in round-trip delay time (latency).
- One-way delay and jitter—The router also supports measurement of one-way delay or jitter from source to destination, or from destination to source.
- One-way frame loss—The router also supports measurement of one-way frame loss from source to destination, or from destination to source.

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

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

Counters for packet loss, corruption, and, out-of-order packets are kept for each bucket, and in each case, a percentage of the total number of samples for that bucket is reported (for example, 4% packet corruption).

For delay, jitter, and loss statistics, the minimum, maximum, mean and standard deviation for the whole bucket are reported, as well as the individual samples or aggregated bins. Also, the overall FLR for the bucket, and individual FLR measurements or aggregated bins are reported for synthetic loss measurement statistics. The packet loss count is the overall number of measurement packets lost in either direction and the one-way FLR measures the loss in each direction separately.

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

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

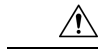
Frame Loss Ratio (FLR) is a primary attribute that can be calculated based on loss measurements. FLR is defined by the ratio of lost packets to sent packets and expressed as a percentage value. FLR is measured in each direction (source to destination and destination to source) separately. Availability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

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

For valid one-way delay results, you must have both local and remote devices time synchronized. In order to do this, you must select sources for frequency and time-of-day (ToD).

Frequency selection can be between any source of frequency available to the router, such as: BITS, GPS, SyncE, or PTP. The ToD selection is between the source selected for frequency and PTP or DTI. Note that NTP is not sufficient.

Configuration Guidelines



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

Before you configure Ethernet SLA, consider the following guidelines:

- Aggregation—Use of the **aggregate none** command significantly increases the amount of memory required because each individual measurement is recorded, rather than just counts for each aggregation bin. When you configure aggregation, consider that more bins will require more memory.
- Buckets archive—When you configure the **buckets archive** command, consider that the more history that is kept, the more memory will be used.
- Measuring two statistics (such as both delay and jitter) will use approximately twice as much memory as measuring one.
- Separate statistics are stored for one-way source-to-destination and destination-to-source measurements, which consumes twice as much memory as storing a single set of round-trip statistics.

- You must define the schedule before you configure SLA probe parameters to send probes for a particular profile. It is recommended to set up the profile—probe, statistics, and schedule before any commit.

Restrictions

One-way delay and jitter measurements are not supported by cfm-loopback profile types.

Configure Ethernet SLA Statistics Measurement in a Profile

To configure SLA statistics measurement in a profile, perform these steps:

1. Enter the Ethernet SLA configuration mode, using the **ethernet sla** command in Global Configuration mode.
2. Create an SLA operation profile with the **profile** *profile-name* **type cfm-delay-measurement** command.
3. Enable the collection of SLA statistics using the **statistics measure** {**one-way-delay-ds | one-way-delay-sd | one-way-jitter-ds | one-way-jitter-sd | round-trip-delay | round-trip-jitter | one-way-loss-ds | one-way-loss-sd**} command.
4. Configure the size and number of bins into which to aggregate the results of statistics collection. For delay measurements and data loss measurements, the default is that all values are aggregated into 1 bin. For synthetic loss measurements, by default the aggregation is disabled. Use the **aggregate** {**bins** *count* **width** [**usec**] *width* | **none**} command to configure the bins.
   - For delay and jitter measurements, you can configure a width value from 1 to 10000 milliseconds, if the number of bins is at least 2. To configure the width value in microseconds, use the **usec** option. You can configure the width value from 1 to 10000000 microseconds.
   - For data loss and synthetic loss measurements, you can configure a width value from 1 to 100 percentage points, if the number of bins is at least 2.
5. Configure the size of the buckets in which statistics are collected, using the **buckets size** *number* **probes** command.
6. Configure the number of buckets to store in memory using the **buckets archive** *number* command.
7. Save the configuration changes using the **end** or **commit** command.

Configuration Example

This example shows configuration of round-trip-delay statistics measurement in 5 bins each with a range of 123 microseconds:

```
Router(config)# ethernet sla
Router(config-sla)# profile test type cfm-delay-measurement
Router(config-sla-prof)# statistics measure round-trip-delay
Router(config-sla-prof-stat-cfg)# aggregate bins 5 width usec 123
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 50
Router(config-sla-prof-stat-cfg)# commit
```

This example shows configuration of round-trip-delay statistics measurement in 5 bins each with a range of 10 milliseconds:

```
Router(config)# ethernet sla
Router(config-sla)# profile test type cfm-delay-measurement
Router(config-sla-prof)# statistics measure round-trip-delay
Router(config-sla-prof-stat-cfg)# aggregate bins 5 width 10
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 50
Router(config-sla-prof-stat-cfg)# commit
```

Verification

This example displays aggregate bins configured with a range of 123 microseconds:

```
Router# show ethernet sla statistics detail
Tue Sep 28 07:59:22.340 PDT
Source: Interface GigabitEthernet0/0/0/2, Domain dom1
Destination: Target MAC Address 0012.0034.0056
================================================================================
Profile 'test', packet type 'cfm-delay-measurement'
Scheduled to run every 1min first at 00:00:31 UTC for 10s

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

No stateful thresholds.

Bucket started at 07:56:31 PDT Tue 28 September 2021 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                  Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
    Min: 0.000ms, occurred at 07:56:32 PDT Tue 28 September 2021
    Max: 1.000ms, occurred at 07:56:31 PDT Tue 28 September 2021
    Mean: 0.100ms; StdDev: 0.300ms

    Bins:
    Range                 Samples   Cum. Count     Mean
    -----------------  ----------  -----------  -------
        0 to 0.123 ms  9  (90.0%)   9  (90.0%)  0.000ms
    0.123 to 0.246 ms  0   (0.0%)   9  (90.0%)        -
    0.246 to 0.369 ms  0   (0.0%)   9  (90.0%)        -
    0.369 to 0.492 ms  0   (0.0%)   9  (90.0%)        -
    > 0.492        ms  1  (10.0%)  10 (100.0%)  1.000ms
```

This example displays aggregate bins configured with a range of 10 milliseconds:

```
Router# show ethernet sla statistics detail
Tue Sep 28 08:00:57.527 PDT
Source: Interface GigabitEthernet0/0/0/2, Domain dom1
Destination: Target MAC Address 0012.0034.0056
================================================================================
Profile 'test', packet type 'cfm-delay-measurement'
Scheduled to run every 1min first at 00:00:31 UTC for 10s

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

No stateful thresholds.

Bucket started at 08:00:32 PDT Tue 28 September 2021 lasting 10s
    Pkts sent: 9; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                  Misordered: 1 (11.1%); Duplicates: 0 (0.0%)
    Result count: 9
```

```
Min: 0.000ms, occurred at 08:00:32 PDT Tue 28 September 2021
Max: 0.000ms, occurred at 08:00:32 PDT Tue 28 September 2021
Mean: 0.000ms; StdDev: 0.000ms

Results suspect due to a probe starting mid-way through a bucket

Bins:
Range           Samples  Cum. Count     Mean
-----------  ----------  ----------  -------
 0 to 10 ms  9 (100.0%)  9 (100.0%)  0.000ms
10 to 20 ms  0   (0.0%)  9 (100.0%)        -
20 to 30 ms  0   (0.0%)  9 (100.0%)        -
30 to 40 ms  0   (0.0%)  9 (100.0%)        -
> 40     ms  0   (0.0%)  9 (100.0%)        -
```

Ethernet frame delay measurement for L2VPN services

Ethernet frame delay measurement complies with the ITU-T Y.1731 standard, which provides comprehensive fault management and performance monitoring recommendations. Delay Measurement Message (DMM) and Delay Measurement Reply (DMR) are used to periodically measure one-way or two-way frame delay and frame delay variation between a pair of point-to-point MEPs. Measurements are made between two MEPs belonging to the same domain and Maintenance Association (MA).

Table 18: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet frame delay measurement for L2VPN services	Release 7.5.3	You can now monitor L2VPN networks and avoid impact to your customers' operations by accurately measuring frame round-trip delays and jitters between two maintenance endpoints (MEPs). This feature lets you detect end-to-end connectivity, loopback, and link trace on MEPs. It reports service performance to your end customers, helping improve technical and operational tasks such as troubleshooting and billing. This feature introduces the **cfm-delay-measurement probe** command.

You can measure frame delay in the Layer 2 networks to detect end-to-end connectivity, loopback, and link trace on Maintenance End Points (MEPs) and also report service performance that helps to improve technical and operational tasks such as troubleshooting, billing, and so on. Frame delay is the duration between the time the source node transmits the first bit of a frame and the time the same source node receives the last bit of the frame.

The frame delay measurement uses the following two protocol data units (PDUs):

- Delay Measurement Message (DMM)—DMM is used to measure frame delay and frame delay variation between a pair of point-to-point Maintenance End Points (MEPs).
- Delay Measurement Response (DMR)—DMR is the delay measurement response sent by the destination MEP. When an MEP receives a DMM frame, the responder MEP responds with a DMR frame. The DMR frame carries a reply information and a copy of the timestamp contained in the DMM frame.

Note DMM sessions (using CFM) are not supported with MACsec enabled on the core interface, as this requires pre-encryption timestamping in the interface group.

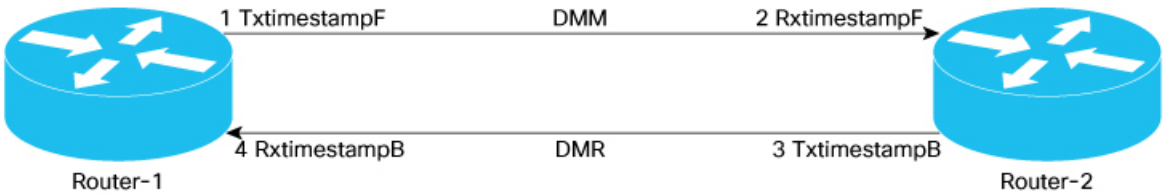
We support one-way and two-way frame delay measurement.

Frame Delay Measurement	Description
One-way frame delay measurement (1DM)	• Measures the frame delay on a unidirectional link between the MEPs. • 1DM requires that clocks at both the transmitting MEP and the receiving MEPs are synchronized. • Measuring frame-delay variation does not require clock synchronization and the variation can be measured using 1DM and DMR frame combination.
Two-way frame delay measurement	• Measures the frame delay on a bidirectional link between the MEPs. • Two-way delay measurement does not require the clocks at both the transmitting MEP and the receiving MEPs to be synchronized. • The two-way frame delay is measured using only DMM and DMR frames.

For more information about CFM, see Configuring Ethernet OAM, on page 67.

Topology

Let's see how a round-trip frame delay is measured with the following sample topology.



- The sender MEP (Router-1) transmits a frame containing delay measurement request information and the timestamp at the which router sends the DMM.

- When packets pass through each interface, timestamps are written into DMMs and DMRs at both local and peer MEPs.
- When the DMM leaves the local interface, the TX timestamp is added to the packet.
- When the receiver MEP (Router-2) receives the frame, records the timestamp at which the receiver MEP receives the frame with the delay measurement request information and the remote MEP (Router-2) responds with an DMR adding the remote TX timestamp to the packet as it leaves the remote interface.

To measure a round-trip delay for a traffic exchange between Router-1 and Router-2, four timestamps get populated as the packet moves through the network.

- Router-1 adds the TxTimestampF when DMM packet is transmitted.
- Router-2 adds RxTimestampF when DMM packet is received by it.
- Router-2 adds TxTimestampB when DMR packet it transmitted.
- Router-1 adds RxTimestampB when DMR is received by it

The round-trip delay is calculated using the following formula:

```
Delay = (RxTimestampB – TxTimestampF) - (TxTimestampB – RxTimestampF)
    = RxTimestampB – TxTimestampF - TxTimestampB + RxTimestampF
    = (RxTimestampF – TxTimestampF) - (TxTimestampB – RxTimestampB)
```

Configure Ethernet Frame Delay Measurement for L2VPN Services

Perform the following tasks to configure Ethernet Frame Delay Measurement for L2VPN Services:

1. Configure L2VPN service.
2. Enable CFM service continuity check.
3. Enable CFM on the interface.
4. Configure Ethernet frame delay measurement.

```
/* Configure L2VPN service */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group evpn_vpws_203
Router(config-l2vpn-xc)# p2p evpn_vpws_phy-100
Router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/2.100
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 30001 target 30001 source 50001
Router(config-l2vpn-xc-p2p)# commit

/* Enable CFM service continuity check */
Router# ethernet cfm
Router(config-cfm# domain xcup1 level 7 id null
Router(config-cfm-dmn)# service xcup1 xconnect group evpn_vpws_Bund
Router(config-cfm-dmn-svc)# mip auto-create all ccm-learning
Router(config-cfm-dmn-svc)# continuity-check interval 1s
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-dmn-svc)# mep-id 4001
Router(config-cfm-dmn-svc)# commit

/* Enable CFM on the interface */
Router(config)# interface GigabitEthernet0/0/0/2.100 l2transport
```

```
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# rewrite ingress tag pop 1 symmetric
Router(config-subif)# mtu 9100
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain bd-domain service bd-service mep-id 4001
Router(config-if-cfm-mep)# sla operation profile test-profile1 target mep-id 1112
Router(config-if-cfm-mep)# commit

/* Configure Ethernet frame delay measurement */
Router(config)# ethernet sla
Router(config-sla)# profile EVC-1 type cfm-delay-measurement
Router(config-sla-prof)# probe
Router(config-sla-prof-pb)# send packet every 1 seconds
Router(config-sla-prof-pb)# schedule
Router(config-sla-prof-schedule)# every 3 minutes for 120 seconds
Router(config-sla-prof-schedule)# statistics
Router(config-sla-prof-stat)# measure round-trip-delay
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 5
Router(config-sla-prof-stat-cfg)# commit
```

Running Configuration

This section shows the Ethernet frame delay measurement running configuration.

```
/* Configure L2VPN service */
l2vpn
xconnect group evpn_vpws_203
p2p evpn_vpws_phy-100
interface GigabitEthernet0/0/0/2.100
neighbor evpn evi 30001 target 30001 source 50001
!
/* Enable CFM service continuity check */
ethernet cfm
 domain xcup1 level 7 id null
  service xcup1 xconnect group evpn_vpws_Bundle_ether203 p2p evpn_vpws-100 id number 4001
   mip auto-create all ccm-learning
   continuity-check interval 1s
   mep crosscheck
    mep-id 4001
!
/* Enable CFM on the interface */
interface GigabitEthernet0/0/0/2.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 mtu 9100
 ethernet cfm
  mep domain bd-domain service bd-service mep-id 4001
   sla operation profile test-profile1 target mep-id 1112
!
/* Configure Ethernet SLA */
ethernet sla
 profile EVC-1 type cfm-delay-measurement
  probe
   send packet every 1 seconds
  !
  schedule
   every 3 minutes for 120 seconds
  !
  statistics
   measure round-trip-delay
    buckets size 1 probes
```

```
    buckets archive 5
!
```

Verification

Verify the frame delay measurement. In the following example, you observe that the sent and received DMM and DMR packets are same. So there is no delay in frame transimission.

```
Router# show ethernet cfm local meps interface GigabitEthernet0/0/0/2.100 verbose

Up MEP on GigabitEthernet0/0/0/2.100 MEP-ID 4001
==============================================================================
  Interface state: Up      MAC address: 0c11.6752.3af8
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  Yes, 10s (Remote Defect detected: No)
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No
  Sending CSF:             No
  Receiving CSF:           No

  Packet         Sent      Received
  ------  ----------  ----------------------------------------------------------
   CCM           19       9 (out of seq: 0)
DMM           473      0
DMR           0        473
```

Link loss forwarding

Link loss forwarding (LLF) is a mechanism used in networking to propagate the status of a network link to other connected devices. When a link experiences a failure or goes down, LLF ensures that this information is forwarded to other network devices, which can then take appropriate actions to maintain network stability and performance.

Table 19: Feature History Table

Feature Name	Release Information	Feature Description
Link loss forwarding	Release 7.9.1	We have now enabled high availability between two bridged interfaces by disabling both interfaces if any one of them fails. Such high availability is enabled because the functionality allows a fault detected on one side of a CFM-protected network to propagate to the other, allowing the device to re-route around the failure. In earlier releases, a failure on one bridged interface did not disable the other interface, and connected devices remained unaware of the link loss. The feature introduces these changes: • **CLI**: New **propagate-remote-status** command • **YANG Data Model**: New XPaths for **Cisco-IOS-XR-um-ethernet-cfm-cfg.yang** (see GitHub, YANG Data Models Navigator)

LLF uses Connectivity Fault Management (CFM) to transmit notification of a signal loss or fault across the network. When there is a fault on a link to a device on one side of the network, the connection to the port on the other side needs to be shutdown so that the device re-routes the traffic.

Link State Monitor and Propagation by CFM

Link State Monitoring involves tracking the status of network links to ensure they are operational and performing as expected. This can include monitoring for link failures, degradations, or other issues that might affect network performance. When a link state changes, this information needs to be propagated throughout the network so that other devices can adjust their routing tables and network operations accordingly.

When there is a fault on a link to a device on one side of the network, the connection to the port on the other side needs to be shutdown so that the device re-routes the traffic. This requires the interface to be TX-disabled.

Link Loss Forwarding (LLF) uses Connectivity Fault Management (CFM) to transmit notification of a signal loss or fault across the network. If a local attachment circuit (AC) on a bridged interface fails, one of the following signals or packet types are sent to the neighboring device:

- Continuity Check Message (CCM) – The CCMs are heartbeat messages exchanged periodically between all the Maintenance End Points (MEPs) in a service. MEPs are members of a particular service within a domain and are responsible for sourcing and sinking CFM frames. Each MEP sends out multicast CCMs, and receives CCMs from all the other MEPs in the service. This allows each MEP to discover its peer MEPs, and to verify that there is connectivity between them.
- Alarm Indication Signal (AIS) – These are messages sent periodically by MEPs that have detected a fault, to the MEPs in the next highest maintenance domain level.
- Client Signal Fail (CSF) – A mechanism for error detection. When a MEP detects an issue, the MEP sends CSF packets to its peer MEPs.

For more information on MEPs, see Maintenance Points, on page 77.

Connectivity Fault Management Daemon (CFMD) and Ether-MA are processes that run on the control plane of the router. Ether-MA handles owner channel communication and resyncs from CFMD, L2VPN, and other Ether MA processes. This module handles the TX-disable and TX-enable events, based on the notifications from CFMD.

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

When an interface receives a fault notification, the transitions are handled as follows:

- The interface is transitioned to TX-disable state.
- A restore or damping timer with a 3.5 * packet interval duration is started.
- If no other fault packets are received after the restore timer ends, the TX-disable state is cleared and the interface is transitioned to TX-enable state.

Restrictions for Link Loss Forwarding for CFM

- Link loss forwarding is not permitted on subinterfaces.
- Link loss forwarding is permitted only on UP MEPs. The UP MEPs send the frames into the bridge relay function and not through the wire connected to the port where the MEP is configured. For more information on UP MEPs, see MEP and CFM Processing Overview, on page 77.

- A damping or restore timer governs transitions of an interface from TX-disabled state to TX-enabled state. The period of the damping timer is calculated by three times the configured CCM interval. You cannot configure the damping timer.
- The damping timer is not provided for transitions of an interface from TX-enabled state to TX-disabled state.
- Link loss forwarding does not work on bundle interfaces configured with LACP.

Configure Link Loss Forwarding for CFM

To configure LLF on a network:

1. Configure a Connectivity Fault Management (CFM) domain and service.
2. Configure a Maintenance End Point (MEP) under the CFM domain and service.
3. Configure continuity check message (CCM) interval on the MEP. The restore timer for a CCM notification is calculated based on the configured CCM interval.
4. Configure Client Signal Fail (CSF) transmission on the MEP, to enable CSF transmission.
5. Configure CSF logging on the MEP, to enable logging on receiving a CSF packet.

Note The CSF configuration is optional and is not required when both the devices in CFM-protected network are running with IOS-XR. This configuration is required for inter-operation with certain client-end setups that contain devices from other clients.

6. Enable LLF on an interface using the **propagate-remote-status** command. This command triggers the interface to be TX-disabled on fault detection.

Configuration Example

```
/* Configure CFM domain, service, and MEP */

Router# configure
Router(config)# ethernet cfm
Router(config-cfm)# domain dom1 level 1 service ser1 bridge group up-meps bridge-domain
up-mep

/* Configure CCM interval */


Router(config-cfm-dmn-svc)# continuity-check interval 1m

/* (Optional) Configure CSF */

Router(config-cfm-dmn-svc)# csf interval 1m cos 4
Router(config-cfm-dmn-svc)# csf-logging
Router(config-cfm-dmn-svc)# commit

/* Enable LLF on an interface */

Router# configure
Router(config)# interface GigabitEthernet0/2/0/0
Router(config-if)# ethernet cfm
Router(config-if-cfm)# mep domain dom1 service ser1 mep-id 1
```

```
Router(config-if-cfm-mep)# propagate-remote-status
Router(config-if-cfm-mep)# commit
```

Running Configuration

```
ethernet cfm
 domain dom1 level 1
  service ser1 bridge group up-meps bridge-domain up-mep
   continuity-check interval 1m
   csf interval 1m cos 4
   csf-logging
  !
 !
!
interface GigabitEthernet0/2/0/0
 ethernet cfm
  mep domain dom1 service ser1 mep-id 1
   propagate-remote-status
  !
 !
!
```

Verification

The following output shows LLF configuration and fault state for each interface:

```
Router# show ethernet cfm interfaces llf location 0/RP0/CPU0
Defects (from at least one peer MEP):
 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down              F - CSF received


GigabitEthernet0/1/0/0
  MEP Defects                           Restore Timer
----- ---------------------------------- --------------
  100 R                                 Not running
  101 None                              10s remaining
  102 RPF                               Not running

GigabitEthernet0/1/0/1
  MEP Defects                           Restore Timer
----- ---------------------------------- --------------
  110 None                              3s remaining

GigabitEthernet0/1/0/2
  MEP Defects                           Restore Timer
----- ---------------------------------- --------------
  120 P                                 Not running
```

The following output shows that the interface received a single CSF packet at 1 minute interval, so that the interface is TX-disabled with a damping timer of 3.5 minutes.

```
Router# show ethernet cfm local meps detail
Domain dom1 (Level 1), Service ser1
UP MEP on GigabitEthernet0/1/0/0 MEP-ID 1
================================================================================
    Interface state: UP    MAC address: 0204.3dbe.c93b
    Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)
    Cross-check errors: 0 missing, 0 unexpected
```

```
    CCM generation enabled: No
    AIS generation enabled: No
    Sending AIS:            No
    Receiving AIS:          No
    Sending CSF:            No
    Receiving CSF:          Yes (Interval: 1min, started 00:03:29 ago)
    TX Disable triggered:   Yes (restore timer not running)
```

The following output shows that the interface received a CCM notification that the peer MEP port is down, so that the interface is TX-disabled.

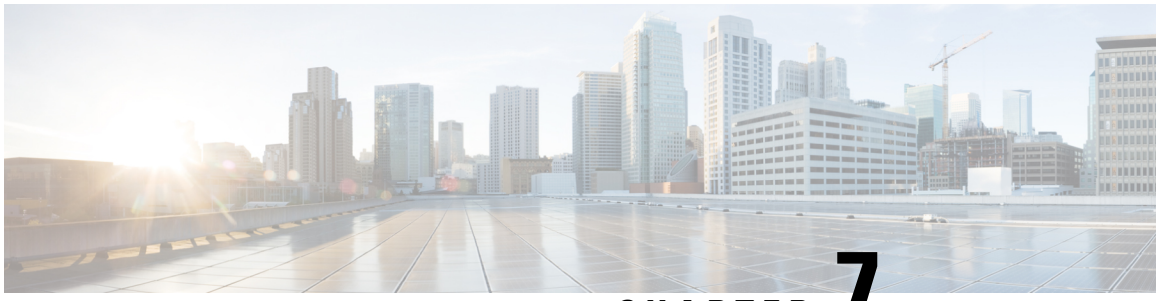
```
Router# show ethernet cfm local meps detail
Domain dom1 (Level 1), Service ser1
UP MEP on GigabitEthernet0/1/0/0 MEP-ID 1
================================================================================
    Interface state: UP    MAC address: 0204.3dbe.c93b
    Peer MEPs: 1 up, 1 with errors, 0 timed out (archived)
    Cross-check errors: 0 missing, 0 unexpected

    CCM generation enabled: Yes, 1min (Remote Defect detected: Yes)
    CCM defects detected:   P - peer port down
    AIS generation enabled: No
    Sending AIS:            No
    Receiving AIS:          No
    Sending CSF:            No
    Receiving CSF:          No
    TX Disable triggered:   Yes (restore timer not running)
```

The following output shows that the interface received CCM notification that the peer MEP port is up, and restore timer is started for the TX-disabled interface.

```
Router# show ethernet cfm local meps detail
Domain dom1 (Level 1), Service ser1
UP MEP on GigabitEthernet0/1/0/0 MEP-ID 1
================================================================================
    Interface state: UP    MAC address: 0204.3dbe.c93b
    Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
    Cross-check errors: 0 missing, 0 unexpected

    CCM generation enabled: Yes, 1min (Remote Defect detected: No)
    AIS generation enabled: No
    Sending AIS:            No
    Receiving AIS:          No
    Sending CSF:            No
    Receiving CSF:          No
    TX Disable triggered:   Yes (restore timer running, 1183ms remaining)
```

The following output shows Ether-MA configured bundles and their members:

```
Router# show ethernet infra internal ether-ma bundles
Bundle interface: Bundle-Ether1 (TX disabled)
Bundle members:
    GigabitEthernet0/1/0/1
    GigabitEthernet0/1/0/2

Bundle interface: Bundle-Ether2
Bundle members:
    GigabitEthernet0/2/0/1
```



CHAPTER 7

IP Event Dampening

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping.

Guidelines and Limitations

See the following guidelines and limitations before configuring IP Event Dampening feature:

- Due to changes in the netstack-IP component, all IP clients observe the impact of interface dampening.
- When dampening is enabled, a penalty value is assigned to an interface. This value starts at 0 and increases by 1000 each time the interface state transitions from up to down.
- For each flap of the interface, a certain penalty is added. The penalty decays exponentially when parameters are configured.
- When the penalty exceeds a certain high level, the interface is dampened. It is unsuppressed when the penalty decays below a low level.
- When an interface is dampened, the IP address and the static routes are removed from the interface. All the clients of IP get an IP delete notification.
- When an interface is unsuppressed, the IP address and the relevant routes are added back. All the clients of IP get an IP address add notification for all the IP addresses of the interface.
- All Layer 3 interfaces that are configured on the Ethernet interface, port changes, and SVI support this feature.

IP Event Dampening Overview

Interface state changes occur when interfaces are administratively brought up or down or if an interface changes state. When an interface changes state or flaps, routing protocols are notified of the status of the routes that are affected by the change in state. Every interface state change requires all affected devices in the network to recalculate best paths, install or remove routes from the routing tables, and then advertise valid

routes to peer routers. An unstable interface that flaps excessively can cause other devices in the network to consume substantial amounts of system processing resources and cause routing protocols to lose synchronisation with the state of the flapping interface.

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping. Dampening an interface removes the interface from the network until the interface stops flapping and becomes stable. Configuring the IP Event Dampening feature improves convergence times and stability throughout the network by isolating failures so that disturbances are not propagated. This, in turn, reduces the utilisation of system processing resources by other devices in the network and improves overall network stability.

Interface State Change Events

This section describes the interface state change events of the IP Event Dampening feature. This feature employs a configurable exponential decay mechanism that is used to suppress the effects of excessive interface flapping or state changes. When the IP Event Dampening feature is enabled, flapping interfaces are dampened from the perspective of the routing protocol by filtering excessive route updates. Flapping interfaces are identified, assigned penalties, suppressed if necessary, and made available to the network when the interface stabilizes.

Suppress Threshold

The suppress threshold is the value of the accumulated penalty that triggers the router to dampen a flapping interface. The flapping interface is identified by the router and assigned a penalty for each up and down state change, but the interface is not automatically dampened. The router tracks the penalties that a flapping interface accumulates. When the accumulated penalty reaches the default or preconfigured suppress threshold, the interface is placed in a dampened state.

Half-Life Period

The half-life period determines how fast the accumulated penalty can decay exponentially. When an interface is placed in a dampened state, the router monitors the interface for additional up and down state changes. If the interface continues to accumulate penalties and the interface remains in the suppress threshold range, the interface will remain dampened. If the interface stabilises and stops flapping, the penalty is reduced by half after each half-life period expires. The accumulated penalty will be reduced until the penalty drops to the reuse threshold. The configurable range of the half-life period timer is from 1 to 45 minutes. The default half-life period timer is 1 minute.

Reuse Threshold

When the accumulated penalty decreases until the penalty drops to the reuse threshold, the route is unsuppressed and made available to other devices in the network. The range of the reuse value is from 1 to 20000 penalties. The default value is 750 penalties.

Maximum Suppress Time

The maximum suppress time represents the maximum time an interface can remain dampened when a penalty is assigned to an interface. The maximum suppress time can be configured from 1 to 255 seconds. The maximum penalty is truncated to maximum 20000 unit. The maximum value of the accumulated penalty is calculated based on the maximum suppress time, reuse threshold, and half-life period.

Affected Components

When an interface is not configured with dampening, or when an interface is configured with dampening but is not suppressed, the routing protocol behavior as a result of interface state transitions is not changed by the IP Event Dampening feature. However, if an interface is suppressed, the routing protocols and routing tables are immune to any further state transitions of the interface until it is unsuppressed.

Route Types

The following interfaces are affected by the configuration of this feature:

- Connected routes:
  - The connected routes of dampened interfaces are not installed into the routing table.
  - When a dampened interface is unsuppressed, the connected routes will be installed into the routing table if the interface is up.
- Static routes:
  - Static routes assigned to a dampened interface are not installed into the routing table.
  - When a dampened interface is unsuppressed, the static route will be installed into the routing table if the interface is up.

Note Only the primary interface can be configured with this feature, and all subinterfaces are subject to the same dampening configuration as the primary interface. IP Event Dampening does not track the flapping of individual subinterfaces on an interface.

Supported Protocols

All the protocols that are used are impacted by the IP Event Dampening feature. The IP Event Dampening feature supports Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), Hot Standby Routing Protocol (HSRP), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), and VRRP. Ping and SSH to the concerned interface IP address does not work.

Note The IP Event Dampening feature has no effect on any routing protocols if it is not enabled or an interface is not dampened.

How to Configure IP Event Dampening

Enabling IP Event Dampening

The `dampening` command is entered in interface configuration mode to enable the IP Event Dampening feature. If this command is applied to an interface that already has dampening configured, all dampening states are reset and the accumulated penalty will be set to 0. If the interface has been dampened, the accumulated penalty will fall into the reuse threshold range, and the dampened interface will be made available to the network. The flap counts, however, are retained.

Table 20: Procedure

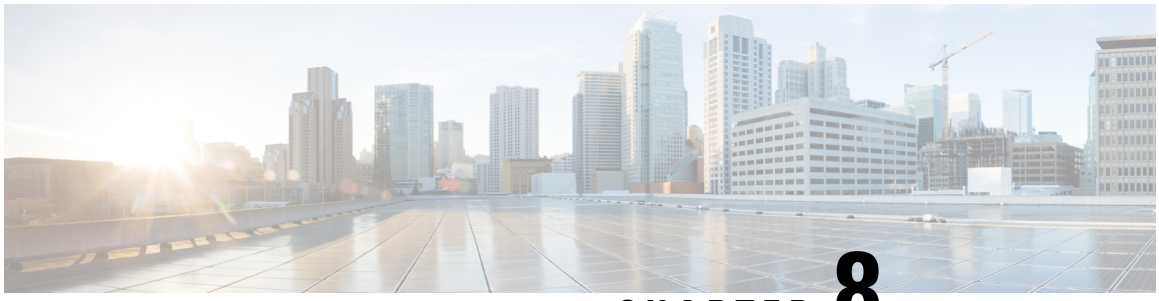
Steps	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type number*	Enters interface configuration mode and configures the specified interface.
Step 3	**dampening** [*half-life-period reuse-threshold*] [*suppress-threshold max-suppress* [*restart-penalty*]]	Enables interface dampening. • Entering the `dampening` command without any arguments enables interface dampening with default configuration parameters. • When manually configuring the timer for the *restart-penalty* argument, the values must be manually entered for all arguments.
Step 4	**end**	Exits interface configuration mode.

Verifying IP Event Dampening

Use the `show dampening interface` or `show interface dampening` commands to verify the configuration of the IP Event Dampening feature.

Table 21: Procedure

Steps	Command or Action	Purpose
Step 1	**show dampening interface**	Displays dampened interfaces.
Step 2	**show interface dampening**	Displays dampened interfaces on the local router.



CHAPTER 8

Configure Link Bundling

Table 22: Feature History Table

Feature Name	Release	Description
1023 Ethernet Bundle Interfaces Support	Release 7.3.15	With the introduction of this enhancement, the maximum system-wide bundle interface scale has increased from 512 to 1023 bundle interfaces. The default value remains at 64-member links for each bundle.
64-bit Bandwidth Support	Release 7.3.15	With this release, the Cisco 8000 Series Router supports 64-bit bandwidth field, as opposed to the previous 32-bit bandwidth field. 64-bit bandwidth enables the system to support interface bandwidths greater than 4.2T.

This module describes the configuration of link bundle interfaces on the Cisco 8000 Series Router.

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

Each bundle has a single MAC, a single IP address, and a single configuration set (such as ACLs).

Note The router supports both Layer 2 and Layer 3 Link Bundles. If the Link Bundle is a Layer 3 interface, the system requires an IP address. If the Link Bundle is a Layer 2 interface, the system does not require an IP address. A Link Bundle on the router may contain Layer 2 and Layer 3 subinterfaces within it. In which case, the Layer 3 subinterfaces require IP addresses, but the Link Bundle interface does not require an IP address.

The router supports bundling for these types of interfaces:

- Ethernet interfaces

Feature History for Configuring Link Bundling

Release	Modification
Release 7.0.11	Support for this feature added on the router.

Release 7.2.1	Mixed speed bundle members feature added on the router.

Limitations and Compatible Characteristics of Ethernet Link Bundles

This list describes the properties and limitations of ethernet link bundles:

- Starting with Cisco IOS XR Release 7.2.1, the router supports mixed speed bundles, allowing member links with different bandwidths to be included in the same bundle. The traffic distribution across bundle members is based on the bandwidth of each link. Mixed speed bundles are subject to a maximum bandwidth ratio of 10:1 between the fastest and slowest member links.

  For example, you can combine a 10 Gbps and a 100 Gbps link or a 100 Gbps and a 40 0Gbps link in the same bundle; however, a 10 Gbps and a 400 Gbps link cannot be bundled together. Load balancing is performed in proportion to the bandwidth of each member link. Typical valid combinations include:

  - 400G, 100G
  - 400G, 40G
  - 400G, 100G, 40G
  - 100G, 40G
  - 100G, 10G
  - 100G, 40G, 10G
  - 40G and 10G

  Additionally, the total weight of the bundle must not exceed 64.

- The weight of each bundle member is the ratio of its bandwidth to the lowest bandwidth member. Total weight of the bundle is the sum of weights or relative bandwidth of each bundle member. Since the weight for a bundle member is greater than or equal to 1 and less than or equal to 10, the total member of links in a bundle is less than 64 in mixed bundle case.
- Any type of Ethernet interfaces can be bundled, with or without the use of Link Aggregation Control Protocol (LACP).
- With Cisco IOS XR Release 7.3.15, a single router can support up to 1023 bundle interfaces, with each bundle accommodating up to 64 member links.

  If adding a new line card causes these limits to be exceeded, the system will experience continuous Out of Resource (OOR) failures. To resolve these errors, you must either reduce the scale or disable the affected line card.

- Physical layer and link layer configuration are performed on individual member links of a bundle.

- Configuration of network layer protocols and higher layer applications is performed on the bundle itself.
- IPv4 and IPv6 addressing is supported on ethernet link bundles.
- A bundle can be administratively enabled or disabled.
- Each individual link within a bundle can be administratively enabled or disabled.
- Ethernet link bundles are created in the same way as Ethernet channels, where the user enters the same configuration on both end systems.
- The MAC address that is set on the bundle becomes the MAC address of the links within that bundle.
- Load balancing (the distribution of data between member links) is done by flow instead of by packet. Data is distributed to a link in proportion to the bandwidth of the link in relation to its bundle.
- QoS is supported and is applied proportionally on each bundle member.
- All links within a single bundle must terminate on the same two systems.
- Bundled interfaces are point-to-point.
- A link must be in the up state before it can be in distributing state in a bundle.
- Only physical links can be bundle members.

Prerequisites for Configuring Link Bundling on a Router

Before configuring Link Bundling, ensure that you meet the following tasks and conditions:

- You know the interface IP address (Layer 3 only).
- You know the links that you must include in the bundle that you are configuring.
- If you are configuring an Ethernet link bundle, you must install Ethernet line cards on the router.

Note For more information about physical interfaces, PLIMs, and modular services cards, refer to the *Cisco 8000 Series Router Hardware Installation Guide*.

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. The system assigns a virtual interface to the bundled link. You can dynamically add and delete component links 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 that the link bundle uses. Packets sent to the link bundle are forwarded to one of the links in the bundle.

A link bundle is a group of ports that the system bundles together and the group then acts as a single link. Following are the advantages of link bundles:

- 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 flow on the available links, if one of the links within a bundle fails. You can add bandwidth without interrupting the packet flow.

Prior to Cisco IOS XR Software Release 7.3.15, the interface bandwidth was stored and processed as a 32-bit value, which supported bundles with an aggregate bandwidth of up to 4.2 Gbps (the sum of its members). Starting with Cisco IOS XR Software Release 7.3.15, the interface bandwidth is stored and processed as a 64-bit value. The 64-bit value supports significantly larger aggregate bandwidths, accommodating bundles with high-bandwidth members whose combined bandwidth can exceed 4.2 Tbps.

All the individual links within a single bundle must be of the same type.

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. The system automatically removes the links from a bundle that are incompatible or have failed.

Link Aggregation Through LACP

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. For the router, 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. The system analyzes these frames to ensure that both the 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 a 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 by using 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. This determines the compatibility of the links that are configured to be members of a bundle.

Bundle MAC addresses in the router come from a set of reserved MAC addresses in the backplane. This MAC address stays with the bundle as long as the bundle interface exists. The bundle uses this MAC address until you configure a different MAC address. The member links use the bundle MAC address when passing the 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.

Configuring LACP Fallback

This section describes how to configure the LACP Fallback feature.

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3
```

Creates and names a new Ethernet link bundle.

This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.

Step 3 **ipv4 address** *ipv4-address mask*

Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0
```

Specifies a primary IPv4 address for an interface.

Step 4 **bundle lacp-fallback timeout 4** *number*

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle lacp-fallback timeout 4
```

Enables the LACP Fallback feature.

Step 5 **end** or **commit**

Example:

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

Saves configuration changes.

Step 6 **show bundle infrastructure database ma bdl-info Bundle-e1010 | inc** *text*

Example:

```
RP/0/RP0/CPU0:router# show bundle infrastructure database ma bdl-info Bundle-e1010 | inc "fallback"
```

(Optional) Shows the MA information of the bundle manager.

Step 7 **show bundle infrastructure database ma bdl-info Bundle-e1015 | inc** *text*

Example:

```
RP/0/RP0/CPU0:router# show bundle infrastructure database ma bdl-info Bundle-e1015 | inc "fallback"
```

(Optional) Shows the MA information of the bundle manager.

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 that distributes traffic over multiple links that are based on certain parameters. The router support load balancing for all links in a bundle using Layer 2, Layer 3, and Layer 4 routing information. Starting with Cisco IOS XR Software Release 7.2.1, bandwidth based load-balancing is applicable to L3 unicast flows.

This section describes the load balancing support on link bundles.

For more information about other forms of load balancing on the router, see the following:

- Per-flow load balancing on non-bundle interfaces using Layer 3 and 4 routing information.
- Pseudowire (PW) Load Balancing beginning in Cisco IOS XR 4.0.1.

Layer 3 Egress Load Balancing on Link Bundles

Layer 3 load balancing support began on the router in Cisco IOS XR 7.0.11 release.

Layer 3 load balancing for link bundles is enabled globally by default.

The ingress linecard does bundle member selection and forwards the packet to the linecard and network processor (NP) corresponding to the selected bundle member. The same hash value is used for both ingress and egress linecards. Therefore, even though the egress linecard also does bundle member selection, it selects the same bundle member that was selected by the ingress linecard.

Multicast IPv4 and IPv6 Traffic

For outbound multicast IPv4 or IPv6 traffic, a set of egress linecards is predetermined by the system. If a bundle interface or bundle subinterface is an outgoing interface, the system selects the bundle member for each outgoing interface in a route based on the multicast group address. This helps with load distribution of multicast routed traffic to different bundle members, while providing traffic sequencing within a specific route.

The egress linecard does NP selection using the same approach, when bundle members are spread across multiple NPs within the egress linecard.

When the packet arrives on an egress NP, it uses the 5-tuple hash to select a bundle member within an NP for each packet. This provides better resiliency for bundle member state changes within an NP.

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.

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface HundredGigE***interface-path*

Example:

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

Creates a Gigabit Ethernet interface and enters interface configuration mode.

Step 3 **bundle id** *number* **mode active**

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active
```

Specifies the bundle interface and puts the member interface in active mode.

Step 4 **lacp period short**

Example:

```
RP/0/RP0/CPU0:router(config-if)# lacp period short
```

Configures a short period time interval for the sending and receiving of LACP packets, using the default time period of 1000 milliseconds or 1 second.

Step 5 **end** or **commit**

Example:

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

or

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

Saves configuration changes.

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

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

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

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

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

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

Example

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

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

Configuring Custom LACP Short Period Time Intervals

This section describes how to configure custom short period time interval for sending and receiving LACP packets on a Gigabit Ethernet interface.

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface HundredGigE***interface-path*

Example:

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

Creates a Gigabit Ethernet interface and enters interface configuration mode.

Step 3 **bundle id** *number* **mode active**

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active
```

Specifies the bundle interface and puts the member interface in active mode.

Step 4 **lacp period***time-interval*

Example:

```
RP/0/RP0/CPU0:router(config-if)# lacp period 300
```

Configures a custom period time interval for the sending and receiving of LACP packets. The interval can be in the range 100 to 1000 ms, in multiples of 100.

Step 5 **end** or **commit**

Example:

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

or

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

Saves configuration changes.

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

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

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

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

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

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

Example

This example shows how to configure the LACP period time interval to the custom time of 300 milliseconds:

```
config
interface HundredGigE 0/1/0/1
   bundle id 1 mode active
   lacp period 300
   commit
```

QoS and Link Bundling

On the router, when the system applies QoS on the bundle for either the ingress or egress direction, QoS is applied at each member interface. For complete information on configuring QoS on link bundles on the router, refer to the *Cisco 8000 Series Aggregation Services Router Modular Quality of Service Configuration Guide* and the *Cisco 8000 Series Aggregation Services Router Modular Quality of Service Command Reference*.

Link Bundle Configuration Overview

The following steps provide a general overview of the link bundle configuration process. Ensure that you clear all previous network layer configuration before adding it to a bundle:

1. In global configuration mode, create a link bundle. To create an Ethernet link bundle, enter the **interface Bundle-Ether** command.
2. Assign an IP address and subnet mask to the virtual interface using the **ipv4 address** command.
3. Add interfaces to the bundle that 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.

Note The system configures a link as a member of a bundle from the interface configuration submode for that link.

Nonstop Forwarding During Card Failover

Cisco IOS XR software supports nonstop forwarding during a failover between active and standby paired RP cards. Nonstop forwarding ensures that there is no change in the state of the link bundles when a failover occurs.

For example, if an active RP fails, the standby RP becomes operational. The system replicates the configuration, node state, and checkpoint data of the failed RP to the standby RP. The bundled interfaces are present when the standby RP becomes the active RP.

Note Failover is always onto the standby RP.

You do not need to configure anything to guarantee that the system maintains the standby interface configurations.

Link Failover

When one member link in a bundle fails, the system redirects the traffic to the remaining operational member links and traffic flow remains uninterrupted.

Link Switchover

By default, a maximum of 64 links in a bundle can actively carry traffic. If one member link in a bundle fails, traffic is redirected to the remaining operational member links.

You can optionally implement 1:1 link protection for a bundle by setting the **bundle maximum-active links** command to 1. By doing so, you designate one active link and one or more dedicated standby links. If the active link fails, a switchover occurs and a standby link immediately becomes active, thereby ensuring uninterrupted traffic.

If the active and standby links are running LACP, you can choose between an IEEE standard-based switchover (the default) or a faster proprietary optimized switchover. If the active and standby links are not running LACP, the proprietary optimized switchover option is used.

Regardless of the type of switchover you are using, you can disable the wait-while timer, which expedites the state negotiations of the standby link and causes a faster switchover from a failed active link to the standby link.

To do so, you can use the **lacp switchover suppress-flaps** command.

LACP Fallback

The LACP Fallback feature allows an active LACP interface to establish a Link Aggregation Group (LAG) port-channel before the port-channel receives the Link Aggregation and Control Protocol (LACP) protocol data units (PDU) from its peer.

With the LACP Fallback feature configured, the router allows the server to bring up the LAG, before receiving any LACP PDUs from the server, and keeps one port active. This allows the server to establish a connection to PXE server over one Ethernet port, download its boot image and then continue the booting process. When the server boot process is complete, the server fully forms an LACP port-channel.

Designate a Member Link as Unviable

Table 23: Feature History Table

Feature Name	Release	Description
Designate a Member Link as Unviable	Release 7.10.1	Earlier, when a member link is added to an interface link bundle, the peer starts using the link as soon as the LACP communication is up. Sometimes, the hardware programming for the data-path does not get complete in this time resulting in packet loss without any notification to the source. You can now mark a member link as unviable to introduce a delay during which the link is treated as standby. By delaying the usage of the member link for data transmission, you can ensure that the link configuration is fully established, which enables successful data transmission. This feature introduces these changes: • **CLI: forwarding-unviable** • **YANG Data Model:** New XPath for `Cisco-IOS-XR-bundlemgr-oper` (see Github, YANG Data Models Navigator).

Link bundling aggregates multiple physical links to a single logical link. When a member link is added to a link bundle, link aggregation control protocol (LACP) communication gets established with the peer to negotiate and control the link aggregation. LACP doesn't have any provision to incorporate a delay before letting data transmission over the link. Therefore, the peer starts using the link when the LACP communication is up. Occasionally, even though the link status is up, and LACP communication is up, the hardware programming for data-path packet forwarding doesn't get complete. In such scenarios, the transmitted data gets lost without any notification or error message to the source or destination of the traffic.

You can now delay the use of such member links, which aren't fully ready to handle data transmission, using the **forwarding-unviable** command. This command configures the link as forwarding-unviable and the member link is considered "standby" for bundle management. As standby member links of a bundle aren't used for data transmission, the usage of forwarding-unviable member links is delayed. When the member link is fully up, that is, the packet forwarding data-path is also completely programmed, you can disable forwarding-unviability of the link using **no forwarding-unviable** command. This removes the forwarding-unviable configuration of the link. Then, the link is treated as an "active" member of the bundle and is used in data transmission and load balancing.

Note It is recommended to wait for a few minutes before running **no forwarding-unviable** command to ensure that the packet forwarding data-path is completely programmed.

Guidelines and Restrictions for Designating Member Links as Unviable

- Forwarding-unviable is disabled on all Ethernet interfaces by default. Therefore, by default, all member links in a bundle are considered "active".
- A link bundle is considered up, only if at least one member link is active. Only the active member links in the link bundle are used for data transmission, load balancing, and redundancy.
- If a link bundle has only one member link, which is forwarding-unviable, the bundle state is considered “down”.
- If all the member links in a bundle are forwarding-unviable, the bundle state is considered “down”.
- Other existing threshold parameters such as minimum-active links, maximum-active links, and maximum-active-bandwidth, which are considered to determine the bundle state, continue to function along with forwarding-unviable functionality. For more details on these parameters, see How to Configure Link Bundling, on page 140.
- There is no effect of forwarding-unviable configuration on individual Ethernet interfaces that are not part of a link bundle. That is, irrespective of the configuration, such non-member interfaces continue to attempt data transmission and reception.

How to Configure Link Bundling

This section contains the following procedures:

Configuring Ethernet Link Bundles

This section describes how to configure an Ethernet link bundle.

Note In order for an Ethernet bundle to be active, you must perform the same configuration on both connection endpoints of the bundle.

Tip You can programmatically perform the configuration using `openconfig-lacp.yang`, `openconfig-if-aggregate.yang` OpenConfig data models, `Cisco-IOS-XR-bundlemgr-oper.yang` Cisco IOS XR native data model 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 8000 Series Routers*.

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3
```

Creates a new Ethernet link bundle with the specified bundle-id. The range is 1 to 65535.

This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface specific configuration commands are entered. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.

Step 3 **ipv4 address** *ipv4-address mask*

Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.

Note
- On the router, only a Layer 3 bundle interface requires an IP address.

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** {**GigabitEthernet** | **TenGigE**} interface-path-id

Example:

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

Enters interface configuration mode for the specified interface.

Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the *rack*/*slot*/*module* format.

Step 10 **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle-id 3
```

Adds the link to the specified bundle.

To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string.

To add the link to the bundle without LACP support, include the optional **mode on** keywords with the command string.

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

Step 11 **bundle port-priority** *priority*

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1
```

(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.

Step 12 **no shutdown**

Example:

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

(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.

Step 13 **exit**

Example:

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

Exits interface configuration submode for the Ethernet interface.

Step 14 **bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **no shutdown exit**

Example:

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/1

RP/0/RP0/CPU0:router(config-if)# bundle id 3

RP/0/RP0/CPU0:router(config-if)# bundle port-priority 2

RP/0/RP0/CPU0:router(config-if)# no shutdown

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

RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/3

RP/0/RP0/CPU0:router(config-if)# bundle id 3

RP/0/RP0/CPU0:router(config-if)# no shutdown

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

(Optional) Repeat Step 8 through Step 11 to add more links to the bundle.

Step 15 **end** or **commit**

Example:

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

or

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

Saves configuration changes.

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

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

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

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

Step 16 **exit**

Example:

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

Exits interface configuration mode.

Step 17 **exit**

Example:

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

Exits global configuration mode.

Step 18 Perform Step 1 through Step 15 on the remote end of the connection.

Brings up the other end of the link bundle.

Step 19 **show bundle Bundle-Ether** *bundle-id*

Example:

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

(Optional) Shows information about the specified Ethernet link bundle.

Step 20 **show lacp bundle Bundle-Ether** *bundle-id*

Example:

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

(Optional) Shows detailed information about LACP ports and their peers.

Configuring VLAN Bundles

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

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

These tasks are described in detail in the procedure that follows.

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

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3
```

Creates and names a new Ethernet link bundle.

This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.

Step 3 **ipv4 address** *ipv4-address mask*

Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.

Step 4 **bundle minimum-active 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 5 **exit**

Example:

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

Exits the interface configuration submode.

Step 6 **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 7 **encapsulation dot1q**

Example:

```
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
```

Sets the Layer 2 encapsulation of an interface.

Step 8 **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 9 **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 10 **exit**

Example:

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

Exits subinterface configuration mode for the VLAN subinterface.

Step 11 Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.

(Optional) Adds more subinterfaces to the bundle.

Step 12 **end** or **commit**

Example:

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

or

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

Saves configuration changes.

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

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

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

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

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

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

Step 13 **exit**

Example:

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

Exits interface configuration mode.

Step 14 **exit**

Example:

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

Exits global configuration mode.

Step 15 **configure**

Example:

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

Enters global configuration mode.

Step 16 **interface** {**GigabitEthernet** | **TenGigE**}*interface-path-id*

Example:

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

Enters interface configuration mode for the Ethernet interface you want to add to the Bundle.

Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the rack/slot/module format.

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

Step 17 **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.

VLANs on an Ethernet Link Bundle

You can configure 802.1Q VLAN subinterfaces on 802.3ad Ethernet link bundles.

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, the system supports all VLAN subinterface configuration on that link bundle.

VLAN subinterfaces can support Ethernet Flow Points (EFPs) and Layer 3 services.

You can configure Layer 3 VLAN subinterfaces as follows:

```
interface bundle-ether instance.subinterface, encapsulation dot1q xxxxx
```

Configuring a Member Link as Unviable

Perform the following task to designate a member link as unviable.

Example Configuration

```
RP/0/RP0/CPU0:ios(config)#interface HundredGigE 0/0/0/34
RP/0/RP0/CPU0:ios(config-if)#forwarding-unviable
RP/0/RP0/CPU0:ios(config-if)#commit
RP/0/RP0/CPU0:ios(config-if)#end
```

Running Configuration

```
RP/0/RP0/CPU0:ios#show running-config interface HundredGigE 0/0/0/34
Thu Apr 20 11:11:55.744 UTC
interface HundredGigE0/0/0/34
 forwarding-unviable
!
```

Verification

Use **show bundle** command to view the forwarding-viable status of LAG members. Here, the interface HundredGigE 0/0/0/34 is added to ethernet bundle 3.

```
RP/0/RP0/CPU0:ios#show bundle
Thu Apr 20 11:29:42.500 UTC

Bundle-Ether3
  Status:                                  Down
  Local links <active/standby/configured>:   0 / 0 / 1
  Local bandwidth <effective/available>:     0 (0) kbps
  MAC address (source):                    78c6.9991.3504 (Chassis pool)
  Inter-chassis link:                      No
  Minimum active links / bandwidth:        1 / 1 kbps
  Maximum active links:                    64
  Wait while timer:                        2000 ms
```

```
Load balancing:
  Link order signaling:                 Not configured
  Hash type:                            Default
  Locality threshold:                   None
LACP:                                   Operational
  Flap suppression timer:               Off
  Cisco extensions:                     Disabled
  Non-revertive:                        Disabled
mLACP:                                  Not configured
IPv4 BFD:                               Not configured
IPv6 BFD:                               Not configured

Port                  Device           State        Port ID         B/W, kbps
--------------------  ---------------  -----------  --------------  ----------
Hu0/0/0/34            Local            Standby      0x8000, 0x0001   100000000
    Link is not forwarding viable and in standby state
```

Configuration Examples for Link Bundling

This section contains the following examples:

Example: Configuring an Ethernet Link Bundle

The following example shows how to join two ports to form an EtherChannel bundle that runs LACP:

```
RP/0/RP0/CPU0:Router(config)# config

RP/0/RP0/CPU0:Router(config-if)# 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)# exit
RP/0/RP0/CPU0:Router(config-if)# 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)# no shutdown
RP/0/RP0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router(config-if)# 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)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# exit
```

This example shows the configuration in the case of a mixed speed bundle:

```
RP/0/RP0/CPU0:Router(config)# config

RP/0/RP0/CPU0:Router(config-if)# interface bundle-ether 50
RP/0/RP0/CPU0:Router(config-if)# root
RP/0/RP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/11
RP/0/RP0/CPU0:Router(config-if)# bundle id 50 mode active
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/16
RP/0/RP0/CPU0:Router(config-if)# bundle id 50 mode active
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/27
RP/0/RP0/CPU0:Router(config-if)# bundleid 50 mode active
```

```
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# interface HundredGigE 0/6/0/1
RP/0/RP0/CPU0:Router(config-if)# bundleid 50 mode active
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# root
RP/0/RP0/CPU0:Router(config)# commit
RP/0/RP0/CPU0:Router(config)# end
```

The following output is shown for the **show bundle bundle-ether** command:

```
show bundle bundle-ether50
```

```
Bundle-Ether50
 Status:                  Up
 Local links <active/standby/configured>:  4 / 0 / 4
 Local bandwidth <effective/available>:   130000000 (130000000) kbps
 MAC address (source):           0011.2233.4458 (Chassis pool)
 Inter-chassis link:            No
 Minimum active links / bandwidth:     1 / 1 kbps
 Maximum active links:           64
 Wait while timer:             2000 ms
 Load balancing:              Default
 LACP:                   Operational
  Flap suppression timer:         Off
  Cisco extensions:             Disabled
 mLACP:                   Not configured
 IPv4 BFD:                 Not configured

 Port         Device      State    Port ID     B/W, kbps
 -------------------- --------------- ----------- -------------- ----------
 Te0/0/0/11      Local      Active    0x8000, 0x0002  10000000
   Link is Active
 Te0/0/0/16      Local      Active    0x8000, 0x0003  10000000
   Link is Active
 Te0/0/0/27      Local      Active    0x8000, 0x0004  10000000
   Link is Active
 Hu0/6/0/1       Local      Active    0x8000, 0x0001  100000000
   Link is Active
```

In order to view the weight of a mixed speed bundle, run the **show bundle load-balancing** command. The following is the truncated output of this command.

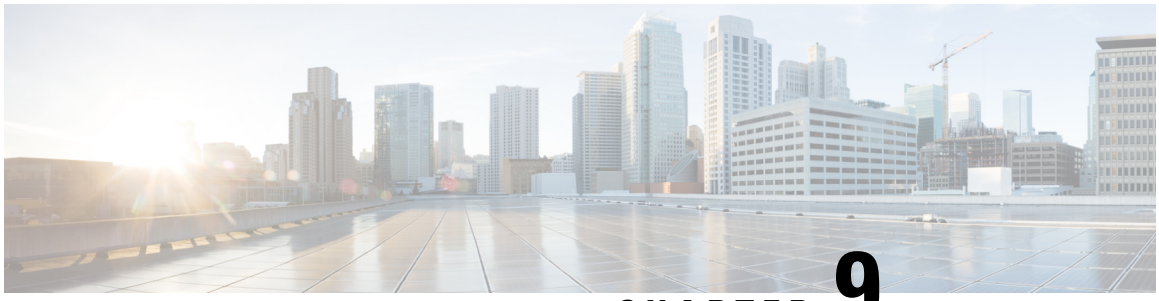
```
show bundle load-balancing bundle-ether50 location 0/0/cpu0

<snip>

Bundle-Ether50
 Type:         Ether (L3)
 Members <current/max>: 4/64
 Total Weighting:    13
 Load balance:     Default
 Locality threshold:  65
 Avoid rebalancing?   False
 Sub-interfaces:    1

 Member Information:
  Port:         LON ULID BW
  -------------------- --- ---- --
  Hu0/6/0/1        0   0 10
  Te0/0/0/11       1   1  1
  Te0/0/0/16       2   2  1
  Te0/0/0/27       3   3  1
```

```
 Platform Information:
 =====================

         * Bundle Summary Information *
          --------------------------

Interface       : Bundle-Ether50    Ifhandle      : 0x00000ce0
Lag ID          : 1           Virtual Port    : 255
Number of Members   : 4           Local to LC    : Yes
Hash Modulo Index   : 13
MGSCP Operational Mode : No

Member Information:
LON    Interface    ifhandle   SFP  port  slot remote/rack_id
-----  ---------------  ----------  ---  ----  ---- --------------
0    Hu0/6/0/1     0x100001c0  648  116   8    0/0
1    Te0/0/0/11    0x04000380   65   9   2    0/0
2    Te0/0/0/16    0x040004c0   67   8   2    0/0
3    Te0/0/0/27    0x04000780   72   4   2    0/0

</snip>
```

Example: Configuring a VLAN Link Bundle

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

```
RP/0/RP0/CPU0:Router(config-subif)# config
RP/0/RP0/CPU0:Router(config-subif)# interface Bundle-Ether 1
RP/0/RP0/CPU0:Router(config-ifsubif)# ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-ifsubif)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links
RP/0/RP0/CPU0:Router(config-ifsubif)# exit
RP/0/RP0/CPU0:Router(config-subif)# ip addr 20.2.3.4/24
RP/0/RP0/CPU0:Router(config-subif)# interface Bundle-Ether 1.1
RP/0/RP0/CPU0:Router(config-subif)# encapsulation dot1q 100
RP/0/RP0/CPU0:Router(config-subif) # ip addr 10.2.3.4/24
RP/0/RP0/CPU0:Router(config-subifif)# no shutdown
RP/0/RP0/CPU0:Router(config-subifif)# exit
RP/0/RP0/CPU0:Router(config-if)# interface Bundle-Ether 1.2
RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RP0/CPU0:Router(config-subif)Router # ip addr20.2.3.4/24

RP/0/RP0/CPU0:Router(config-subifif)# no shutdown
RP/0/RP0/CPU0:Router(config-subifif)# 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
```



CHAPTER 9

Configuring Traffic Mirroring

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

Feature History for Traffic Mirroring

Release 7.3.1	SPAN to File feature was introduced.
Release 7.2.12	Local SPAN feature was introduced.
Release 7.0.14	Support for the following features was introduced in ERSPAN: • Configuration of IP DSCP. • Tunnel IP. • Ability to source ranges of interfaces and SVIs. • Sequence bit is set in the GRE header and the value of sequence number is always 0 for ERSPAN packets. • ERSPAN and Security ACL should be separate. Support for File Mirroring was introduced.
Release 7.0.11	This feature was introduced.

• Troubleshooting Traffic Mirroring, on page 193

Introduction to Traffic Mirroring

Traffic mirroring, which is sometimes called port mirroring, or Switched Port Analyzer (SPAN) is a Cisco proprietary feature. Traffic mirroring enables you to monitor Layer 3 network traffic passing in, or out of, a set of Ethernet interfaces. You can then pass this traffic to a network analyzer for analysis.

Traffic mirroring copies traffic from one or more Layer 3 interfaces or sub-interfaces. Traffic mirroring then sends the copied traffic to one or more destinations for analysis by a network analyzer or other monitoring device. Traffic mirroring does not affect the switching of traffic on the source interfaces or sub-interfaces. It allows the system to send mirrored traffic to a destination interface or sub-interface.

Traffic mirroring is introduced on switches because of a fundamental difference between switches and hubs. When a hub receives a packet on one port, the hub sends out a copy of that packet from all ports except from the one at which the hub received the packet. In case of switches, after a switch boots, it starts to build up a Layer 2 forwarding table on the basis of the source MAC address of the different packets that the switch receives. After the system builds this forwarding table, the switch forwards traffic that is destined for a MAC address directly to the corresponding port.

Layer 2 SPAN is not supported on the router.

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

Implementing Traffic Mirroring on the Cisco 8000 Series Routers

ERSPAN

Encapsulated Remote Switched Port Analyzer (ERSPAN) is a traffic mirroring mechanism used to monitor network traffic passing in or out of a set of ports on a router. It copies or mirrors traffic from one or more source ports and sends the copied traffic through GRE tunnels to one or more destinations for analysis. The destination may be a network analyzer or other monitoring devices.

Table 24: Feature History Table

Feature Name	Release Information	Feature Description
Partial packet capture ability for ERSPAN (Rx)	Release 7.5.3	With this feature, you can perform partial packet capture in the RX direction. Earlier, the ability for entire packet capture was available, now you can choose entire or partial packet capture in the RX direction. Here, partial packet capture is also known as truncation.

Feature Name	Release Information	Feature Description
ERSPAN over MPLS traffic	Release 7.5.3	With this release, the router allows you to mirror MPLS traffic and set up the GRE tunnel with the next hop over a labeled path. This feature helps you to remote-monitor the traffic on traffic analyzers.
Higher payload analysis with eight ERSPAN sessions	Release 7.3.2	With this release, Cisco 8000 Series routers support eight ERSPAN sessions. This functionality helps you analyze higher payloads in real time across Layer 3 domains on your network.
ERSPAN over GRE IPv6	Release 7.3.2	With this release, the router allows you to mirror IPv4 or IPv6 traffic with ERSPAN over GRE IPv6 sessions to monitor traffic on remote traffic analyzers. In earlier releases, ERSPAN traffic monitoring was possible only on IPv4 networks.

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.

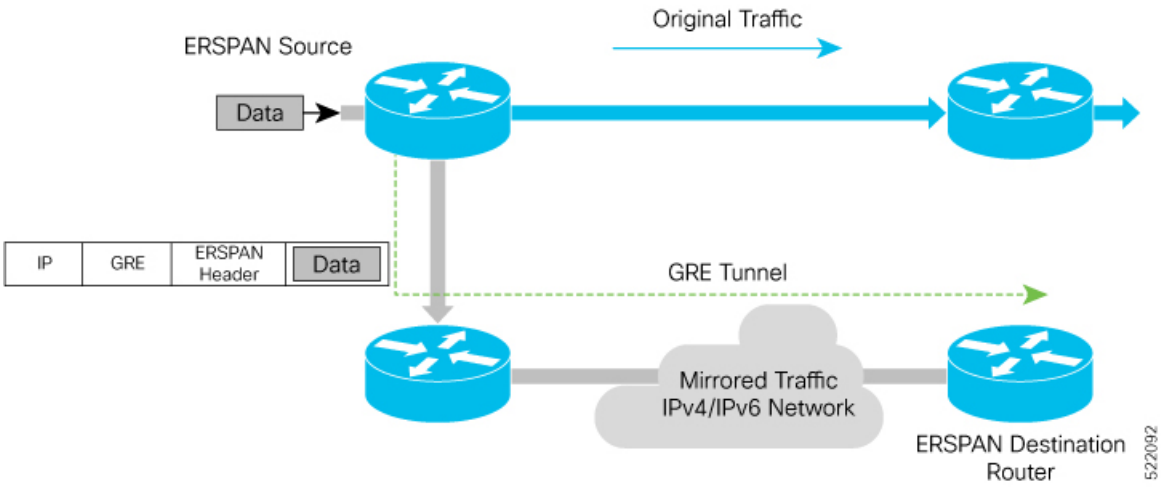
ERSPAN transports mirrored traffic over an IP network. The traffic is encapsulated at the source router and is transferred across the network.

From Cisco IOS XR Software Release 7.5.3 onwards, the packet truncation feature is supported over remote GRE tunnels. You can now get the flexibility to truncate packets and mirror the traffic.

Starting with Cisco IOS XR Software Release 7.0.14, sequence bit is set in the GRE header and the value of sequence number is always 0 for ERSPAN packets.

Starting with Cisco IOS XR Software Release 7.5.3, the sequence number bit will always be set to one and the sequence number field (4 bytes), will always be set to zero.

Figure 8: ERSPAN over GRE



Supported Capabilities

The following capabilities are supported:

- The source interfaces are layer 3 interfaces, such as physical, and bundle interfaces or subinterface.
- The routers mirror IPv4 and IPv6 traffic.
- ERSPAN with GRE IPv4 or IPv6 has tunnel destinations.
- ERSPAN over GRE IPv4 and IPv6 supports SPAN ACL.
- Supports MPLS traffic mirroring and GRE tunnel configuration with the next hop over a labeled path.
- Each monitor session allows only one destination interface.
- ACL permit or deny entries with capture action are part of mirroring features.
- The next hop interface must be a main interface. It can be a Physical or Bundle interface.
- Supports full packet capture.
- In ERSPAN over GRE IPv6, the **HopLimit** and **TrafficClass** fields in outer IPv6 header are editable under the tunnel configuration.
- The maximum SPAN sessions supported in the Cisco 8000 Router are as follows:.

SPAN Type	7.3.1 and Prior Releases	7.3.2 and Later Releases
ERSPAN (GRE IPv4, GRE IPv6, or GRE IPv4 + GRE IPv6)	4	8
Local SPAN	4	4
SPAN to File	4	4
Combined SPAN (GRE IPv4 + GRE IPv6 + Local SPAN + SPAN to File)	4	8

Supported Capabilities for ERSPAN Packet Truncation support

The following are the capabilities and requirements:

- Ability to enable the new ERSPAN GREv4 and GREv6 truncation configuration per device.
- Truncation configuration should be on the monitor sessions. Packets received from all sources will only be truncated when you configure the truncation on a monitor session.
- By default, the whole packet will be mirrored without the **mirror first <number>** (truncation size) configuration.
- If the monitor session truncation size is less than the configured-truncation size (343 bytes), then whole packet is mirrored.

  If the monitor session truncation size exceeds 343 bytes, the configuration is accepted. However, only 343 bytes truncation size is programmed.

  An `ios-msg` is displayed to warn the user.

  Example: `ERSPAN only support 343 bytes truncation size. monitor-session with session_id <id> will be set to 343 bytes only.`

Restrictions

The following are the ERSPAN and SPAN ACL restrictions:

- The ERSPAN mirror packet is received with a TTL minus 1.

  The mirror packet is not identical to the incoming packet and TTL minus 1 is the expected value in the ERSPAN packet.
- The router mirrors only unicast traffic.

  However, from Cisco IOS XR Software Release 7.5.3 onwards, the router can mirror multicast traffic.
- Remove and re-apply monitor-sessions on all interfaces after modifying the access control list (ACL).
- GRE tunnel is only dedicated to ERSPAN mirrored packets. There should be no IPv4 and IPv6 address configured under the GRE tunnel.
- Only ERSPAN TYPE II header is supported. The value of the index field is always 0. The value of the session-ID field is an internal number that is used by the data path to distinguish between sessions.
- Traffic accounting of the ERSPAN mirrored packets is not supported.

Note You can view the SPAN packet count per session, using the **show monitor-session status internal** command.

- ERSPAN decapsulation is unsupported.
- From Cisco IOS XR Software Release 7.5.3 onwards, the ERSPAN will be functional regardless of any configuration related to MPLS or LDP present on the router.
- MPLS packet mirroring is supported only from Cisco IOS XR Software Release 7.5.3 onwards.

- On Q100-based systems, due to data path limitations, only the upper 64 bits of the source IPv6 address in the outer IPv6 header of an ERSPAN packet are valid; the lower 64 bits are set to zero. The destination GREv6 IPv6 address must use the full 128-bit value.

Traffic Mirroring Terminology

- Ingress traffic—Traffic that enters the switch.
- Egress traffic—Traffic that leaves the switch.
- Source port—A port that the system monitors 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 switched or routed port that you monitor for network traffic analysis. In a single local or remote traffic mirroring session, you can monitor source port traffic, such as received (Rx) for ingress 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, sub-interface, 100-Gigabit Ethernet, or 400-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.
- Each source port can be configured with a direction (ingress) to monitor. 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 monitor session can have only one destination port.
- A single destination port can belong to only one monitor session.

Note The destination of ERSPAN monitoring session is a GRE IPv4 or IPv6 tunnel.

Supported Traffic Mirroring Types

The system supports the following traffic mirroring types:

- ACL-based traffic mirroring. The system mirrors traffic that is based on the configuration of the global interface ACL.
- Layer 3 traffic mirroring is supported. The system can mirror Layer 3 source ports.

ACL-Based Traffic Mirroring

You can mirror traffic that is based on the definition of a global interface access list (ACL). When you are mirroring Layer 3 traffic, the ACL is configured using the **ipv4 access-list** or **ipv6 access-list** command with the **capture** keyword. The **permit** and **deny** commands determine the behavior of regular traffic. The **capture** keyword designates that the packet is to be mirrored to the destination port.

Starting with Cisco IOS XR Software Release 7.0.14, configuration of ERSPAN and security ACL will be separate. Neither of these will have an impact or dependency on the other, but both can be applied simultaneously.

ERSPAN over GRE IPv6

The ERSPAN over GRE IPv6 feature enables mirroring IPv4 or IPv6 traffic in your network. The router encapsulates the traffic adding an ERSPAN header inside the GRE IPv6 packet. The GRE header of the ERSPAN encapsulated packets have the sequence number set to 0. The router sends the replicated traffic packet to be monitored to the destination through the GRE IPv6 channel to achieve traffic mirroring. The mirrored traffic is sent to remote traffic analyzer for monitoring purposes. For the traffic mirroring to work, the ERSPAN GRE IPv6 tunnel next-hop must have ARP or neighbor resolved. We recommend using the `cef proactive-arp-nd` enable command to configure missing adjacency information for the next hop.

```
Router# configure
Router(config)# cef proactive-arp-nd enable
Router(config)# commit
```

Configuring ERSPAN over GRE IPv6

1. Enable GRE IPv6 tunnel configuration.

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#interface tunnel-ip1
RP/0/RP0/CPU0:router(config-if)#tunnel mode gre ipv6
RP/0/RP0/CPU0:router(config-if)#tunnel source 2001:DB8:1::1
RP/0/RP0/CPU0:router(config-if)#tunnel destination 2001:DB8:2::1
RP/0/RP0/CPU0:router(config-if)#no shut
RP/0/RP0/CPU0:router(config)#commit
```

2. Enable ERSPAN session.

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#monitor-session mon1 ethernet
RP/0/RP0/CPU0:router(config-mon)#destination interface tunnel-ip1
RP/0/RP0/CPU0:router(config-mon)#commit
RP/0/RP0/CPU0:router(config-mon)#end
```

3. Configure ERSPAN session under port to be monitored.

```
RP/0/RP0/CPU0:router(config)#interface HundredGigE0/1/0/14
RP/0/RP0/CPU0:router(config-if)#monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)#exit
RP/0/RP0/CPU0:router(config-if)#exit
RP/0/RP0/CPU0:router(config)#interface Bundle-Ether1
RP/0/RP0/CPU0:router(config-if)#monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)#exit
RP/0/RP0/CPU0:router(config-if)#exit
RP/0/RP0/CPU0:router(config)#interface HundredGigE0/1/0/15.100
RP/0/RP0/CPU0:router(config-subif)#monitor-session mon1 ethernet direction rx-only
```

Verification

Use the `show monitor-session status` command o verify the configuration of the ERSPAN over GRE IPv6 feature.

```
P/0/RP0/CPU0:router#show monitor-session mon1 status
Monitor-session mon1
Destination interface tunnel-ip1
================================================================================
Source Interface            Dir         Status
---------------------      ----  ------------------------------------------------
Hu0/1/0/14                  Rx    Operational
Hu0/1/0/15.100              Rx    Operational
BE1                         Rx    Operational
BE1.1                       Rx    Operational


RP/0/RP0/CPU0:R1-SF-D#show monitor-session erspan3 status internal
Thu Jul 15 06:00:14.720 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session erspan3 (ID 0x00000007) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip372 (0x0f00049c)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv6
            Source IP: 77:3:1::79
            Dest IP: 95::90
            VRF:
            ToS: 100
            TTL: 200
            DFbit: Not set
0/3/CPU0: Destination interface tunnel-ip372 (0x0f00049c)
          Tunnel data:
            Mode: GREoIPv6
            Source IP: 77:3:1::79
            Dest IP: 95::90
            VRF:
            ToS: 100
            TTL: 200
            DFbit: Not set
0/RP0/CPU0: Destination interface tunnel-ip372 (0x0f00049c)
          Tunnel data:
            Mode: GREoIPv6
            Source IP: 77:3:1::79
            Dest IP: 95::90
```

```
            VRF:
            ToS: 100
            TTL: 200
            DFbit: Not set
Information from SPAN EA on all nodes:
Monitor-session 0x00000007 (Ethernet)
0/3/CPU0: Name 'erspan3', destination interface tunnel-ip372 (0x0f00049c)
Platform, 0/3/CPU0:
  Monitor Session ID: 7

  Monitor Session Packets: 2427313444
  Monitor Session Bytes: 480591627492
```

Configuring Partial Packet Capture Ability for ERSPAN (RX)

To configure partial traffic mirroring, use the **mirror first** command in monitor session configuration mode.

`Mirror first <number>`: Configures the size of truncation packets for an ERSPAN session

Use the following command to create a ERSPAN monitor session for mirroring the packets:

```
monitor-session <name> [ethernet]
destination interface tunnel-ip <number>
mirror first <number>
    traffic-class <traffic-class>
```

Configuration Example

Use the following command to create a ERSPAN monitor session for mirroring packets to Tunnel-IP 30 with truncation enabled:

```
monitor-session mon1 ethernet
 destination interface tunnel-ip 30
 mirror first  343
!
```

Attach the session to the interfaces using the following configuration:

```
interface <>
  monitor-session session-name ethernet direction rx-only|tx-only|both | acl [acl_name]
```

Running Configuration

```
interface tunnel-ip30
 tunnel mode gre ipv4
 tunnel source 2.2.2.2
 tunnel destination 200.0.0.2
!

interface HundredGigE0/0/0/12
 ipv4 address 12.0.0.2 255.255.255.0
 monitor-session mon1 ethernet direction rx-only
!
```

Verification

The **show monitor-session status internal** displays the size of the programmed truncation.

Example:

```
Router#show monitor-session mon1 status internal
Fri Apr 12 18:50:45.006 UTC
Information from SPAN Manager and MA on all nodes:
Packet truncation size: 343B
Monitor-session mon1 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface Tunnel-IP 20 (0x0f000250)
          Last error: Success
```

```
Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/RP0/CPU0: Name 'mon1', destination interface Tunnel-IP 20 (0x0f000250)
Platform, 0/RP0/CPU0:

  Monitor Session Packets: 142462

  Monitor Session Bytes: 7653237
```

ERSPAN traffic to a destination in a non-default VRF

ERSPAN traffic to a destination in a non-default VRF is an ERSPAN feature that sends mirrored traffic over GRE tunnels that belong to different VRF instances. This capability helps design a network with multiple Layer 3 partitions, enabling traffic segregation and management across different network segments.

Table 25: Feature History Table

Feature Name	Release Information	Description
ERSPAN traffic to a destination in a non-default VRF	Release 7.5.2 Release 7.3.4	Encapsulated Remote Switched Port Analyzer (ERSPAN) now transports mirrored traffic through GRE tunnels with multiple VRFs, helping you design your network with multiple Layer 3 partitions. In earlier releases, ERSPAN transported mirrored traffic through GRE tunnels that belonged to only default VRF.

Restrictions for Traffic Mirroring

The system supports the following forms of traffic mirroring:

- Mirroring traffic to a GRE IPv4 or IPv6 tunnel (also known as Encapsulated Remote Switched Port Analyzer [ER-SPAN] in Cisco IOS Software). The system allows 8 monitor sessions for ERSPAN, 4 monitor sessions for Local SPAN, and 4 monitor sessions for SPAN to File. The total number of monitor sessions for all SPAN features is 8.
- The system does not support traffic mirroring counters per interface.
- The system does not support bundle member interfaces as sources for mirroring sessions.
- The router does not support port-level mirroring for any type of SPAN.
- ERSPAN tunnel statistics is not supported.
- The dropped packets at NPU cannot be captured by regular ERSPAN session. For capturing dropped packets at NPU, use Mirroring forward-drop packets, on page 187 feature.

The following general restrictions apply to traffic mirroring using ACLs:

- Configure ACLs on the source interface to avoid default mirroring of traffic. If a Bundle interface is a source interface, configure the ACLs on the bundle interface (not bundle members).

The following restrictions apply to ERSPAN ACL:

- ERSPAN next-hop must have ARP resolved.
  - Any other traffic or protocol triggers ARP.
- ERSPAN decapsulation is not supported.
- ERSPAN does not work if the GRE next hop is reachable over subinterface. For ERSPAN to work, the next hop must be reachable over the main interface.
- However, from Cisco IOS XR Software Release 7.5.3 onwards, GRE next hop can be resolved over subinterface or the main interface.

Modifying ERSPAN monitor-session configuration

When you modify the ERSPAN monitor-session configuration, the **show configuration** and **show configuration commit changes** command outputs differ. Specifically, the **show configuration commit changes** command output displays some extraneous ACL commands deleted and added back. This modified output doesn't impact your configuration or affect performance. This issue is fixed in Cisco IOS XR Release 7.5.1.

The following example highlights the extraneous ACL commands under the **show configuration commit changes** command output.

```
Router(config)#interface HundredGigE0/1/0/0
Router(config-if)#no monitor-session ERSPANTun2005
Router(config-if)#monitor-session ERSPANTun2 ethernet direction rx-only port-level
Router(config-if-mon)#acl
Router(config-if-mon)#acl ipv4 erspan-filter
Router(config-if-mon)#acl ipv6 erspan-filter-ipv6
Router(config-if-mon)#
Router(config-if-mon)#show configuration
Building configuration...
!! interface HundredGigE0/1/0/0
  monitor-session ERSPANTun2 ethernet direction rx-only port-level
   acl
   acl ipv4 erspan-filter
   acl ipv6 erspan-filter-ipv6
 !
!
end

Router(config-if-mon)#commit
Router(config-if-mon)#end
Router#sh configuration commit changes las 1
Building configuration...
!!
interface HundredGigE0/1/0/0
 no monitor-session ERSPANTun2005 ethernet direction rx-only port-level
 monitor-session ERSPANTun2 ethernet direction rx-only port-level
  no acl
  acl
  no acl ipv4 erspan-filter
  acl ipv4 erspan-filter
  no acl ipv6 erspan-filter-ipv6
  acl ipv6 erspan-filter-ipv6
```

```
 !
!
end
```

Configuring Traffic Mirroring

These tasks describe how to configure traffic mirroring:

Configuring ACLs for Traffic Mirroring

This section describes the configuration for creating ACLs for traffic mirroring. You must configure the global interface ACLs by using one of the following commands with the **capture** keyword:

- **ipv4 access-list**
- **ipv6 access-list**

Note Starting with Cisco IOS XR Software Release 7.0.14, ACL feature will provide a support of separate ACL configuration for SPAN.

Configuration

- **Security ACL**

  Use the following configuration to configure ACLs for traffic mirroring.

```
/* Create an IPv4 ACL (TM-ACL) for traffic mirroring */
Router(config)# ipv4 access-list TM-ACL
Router(config-ipv4-acl)# 10 permit udp 10.10.10.0 0.0.0.255 eq 10 any capture
Router(config-ipv4-acl)# 20 permit udp 10.10.10.0 0.0.0.255 eq 20 any
Router(config-ipv4-acl)# exit
Router(config)# commit

/* Apply the traffic monitoring to SPAN source interface */
Router(config)# interface HundredGigE0/0/0/12
Router(config-if)# monitor-session mon1 ethernet direction rx-only port-level acl
Router(config-if)# ipv4 access-group TM-ACL ingress
!
```

Use the following configuration as an example to deny data forwarding for an ACE entry, but still mirror the traffic:

```
ipv4 access-list acl1
10 deny ipv4 any 2.1.0.0/16 capture
20 permit ipv4 any any
!
```

If `acl1` is attached to the interface as shown below:

```
RP/0/RP0/CPU0(config-if)# ipv4 access-group acl1 ingress
```

Data Traffic to 2.1.0.0/16 is dropped. Mirroring happens only if `icmp-off` keyword is added to the ACE as shown below. If this keyword is not added, mirroring does not take place. Furthermore, the `icmp-off` workaround is applicable only to security ACL.

```
ipv4 access-list acl1
10 deny ipv4 any 2.1.0.0/16 capture icmp-off
20 permit ipv4 any any
!
```

- **SPAN ACL**
  - SPAN ACL does not support User Defined Fields (UDF).
  - Deny action in SPAN ACL is ignored, and no packet drops from SPAN ACL. Deny ACEs will be internally converted to permit ACEs. Packets will also be mirrored.
  - There is no implicit deny-all entry in SPAN ACL.
  - IPV6 ACL is required for mirroring IPV6 packet, if IPV4 ACL is configured, and vice versa. This follows the same structure as Security ACL with IPv4 and IPv6 mirror options.

Use the following configuration to enable traffic mirroring with ACLs.

```
/* Create a SPAN IPv4 ACL (v4-monitor-acl) for traffic mirroring */
Router(config)# ipv4 access-list v4-monitor-acl
Router(config-ipv4-acl)# 10 permit udp 20.1.1.0 0.0.0.255 eq 10 any
Router(config-ipv4-acl)# 20 permit udp 30.1.1.0 0.0.0.255 eq 20 any
Router(config-ipv4-acl)# exit
Router(config)# commit

/*Create a SPAN IPv6 ACL (v6-monitor-acl) for traffic mirroring */
Router(config)# ipv6 access-list v6-monitor-acl
Router(config-ipv6-acl)# 10 permit ipv6 host 120:1:1::1 host 130:1:1::1
Router(config-ipv6-acl)# exit

/* Apply the traffic monitoring to SPAN source interface */
Router(config)# interface HundredGigE0/0/0/12
Router(config-if)# monitor-session mon1 ethernet direction rx-only
Router(config-if)# acl ipv4 v4-monitor-acl
Router(config-if)# acl ipv4 v6-monitor-acl!
```

Note For SPAN to work, the `capture` keyword is required for Security ACL.

Use the `show access-lists [ipv4 | ipv6] acl-name hardware ingress span [detail | interface | location | sequence | verify] location x` command to display ACL information:

```
Router# show access-lists ipv4 v4span1 hardware ingress span interface bundle-Ether 100
location 0/3/cpu0
ipv4 access-list v4span1
10 permit ipv4 host 51.0.0.0 host 101.0.0.0
20 permit ipv4 host 51.0.0.1 host 101.0.0.1
30 permit ipv4 host 51.0.0.2 any
40 permit ipv4 any host 101.0.0.3
50 permit ipv4 51.0.1.0 0.0.0.255 101.0.1.0 0.0.0.255
60 permit ipv4 51.0.2.0 0.0.0.255 101.0.2.0 0.0.0.255 precedence critical
```

Troubleshooting ACL-Based Traffic Mirroring

Take note of these configuration issues:

- Even when the system configures the **acl** command on the source mirroring port, if the ACL configuration command does not use the **capture** keyword, the system does not mirror traffic.
- If the ACL configuration uses the **capture** keyword, but you have not configured the **acl** command on the source port, the system mirrors the traffic, but does not apply access list configuration.

This example shows both the **capture** keyword in the ACL definition and the **acl** command that is configured on the interface:

```
/* Create an IPv4 ACL (TM-ACL) for traffic mirroring */
Router(config)# ipv4 access-list TM-ACL
Router(config-ipv4-acl)# 10 permit udp 10.1.1.0 0.0.0.255 eq 10 any capture
Router(config-ipv4-acl)# 20 permit udp 10.1.1.0 0.0.0.255 eq 20 any

Apply the traffic monitoring to interface
Router(config)#interface HundredGigE0/0/0/12
Router(config-if)# monitor-session mon1 ethernet direction rx-only port-only acl
Router(config-if)# ipv4 access-group TM-ACL ingress
```

Flexible CLI for ERSPAN

Starting with Cisco IOS XR Software Release 7.0.14, ERSPAN can be configured using flexible CLI. This CLI is a single configuration object containing all the properties of an ERSPAN session, tunnel properties, and the list of source interfaces, which can be easily removed and re-added. Flexible CLI minimises risk of user error and promotes operational simplicity.

Configure a flexible CLI group in ERSPAN containing:

- Global ERSPAN session configuration
- Tunnel interface configuration
- ERSPAN source attachment configuration, applied to a regexp of interface names

Note The flexible CLI group contains only the session and interface properties. The session and interface objects themselves must be created in the configuration as usual.

The following example shows a global flexible CLI configuration:

```
group erspan-group-foo
  monitor-session 'foo' ethernet   /* Global configuration */
    destination interface tunnel-ip0
  !
  interface 'tunnel-ip0'   /* Tunnel interface configuration */
    tunnel tos 10
    tunnel mode gre ipv4
    tunnel source 10.10.10.1
    tunnel destination 20.20.20.2
  !
  interface 'GigabitEthernet0/0/0/[0-3]'   /* Interface configuration */
    monitor-session foo ethernet
  !
end-group
```

To enable all ERSPAN configurations, execute `apply-group erspan-group-foo` command. To disable ERSPAN configuration, delete this command.

Note The following three keywords are regular expressions and must be quoted:

- Definition of session name (example: `foo`)
- Definition of tunnel name (example: `tunnel-ip0`)
- Set of source interface names (example: `GigabitEthernet0/0/0/[0-3]`)

Use the `show running-config inheritance` command to view the final configuration after the group is expanded, and the `show monitor-session status` to check the operational state of ERSPAN session.

Note Starting from Release 7.3.3, when a combination of IP-in-IP decap and GRE ERSPAN tunnels are in use, resource utilization of IP-in-IP decap tunnels is accounted. However, resource utilization of ERSPAN GRE tunnels is not accounted in the *Total In Use* counter of **show controllers npu resources sipidxtbl location all** command output, but the *OOR State* would display *RED* if the total number of IP-in-IP decap and ERSPAN GRE tunnels reach 15.

Attaching the Configurable Source Interface

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **interface** *type number*

Example:

```
RP/0/RP0/CPU0(config)# interface HundredGigE 0/1/0/10/0/1/0
```

Enters interface configuration mode for the specified source interface. The interface number is entered in *rack*/*slot*/*module*/*port* notation. For more information about the syntax for the router, use the question mark (?) online help function.

Step 3 **ipv4 access-group** *acl-name* {**ingress** | **egress**}

Example:

```
RP/0/RP0/CPU0(config-if)# ipv4 access-group acl1 ingress
```

Controls access to an interface.

Step 4 **monitor-session** *session-name* **ethernet direction rx-onlyport-level**

Example:

```
RP/0/RP0/CPU0(config-if)# monitor-session mon1 ethernet direction rx-only port-level acl
RP/0/RP0/CPU0(config-if-mon)#
```

Attaches a monitor session to the source interface and enters monitor session configuration mode.

Note
rx-only specifies that only ingress traffic is replicated.

Step 5 **acl**

Example:

```
RP/0/RP0/CPU0(config-if-mon)# acl
```

Specifies that the traffic mirrored is according to the defined ACL.

Note
If an ACL is configured by name then this overrides any ACL that may be configured on the interface.

Step 6 **exit**

Example:

```
RP/0/RP0/CPU0(config-if-mon)# exit
RP/0/RP0/CPU0(config-if)#
```

Exits monitor session configuration mode and returns to interface configuration mode.

Step 7 **end** or **commit**

Example:

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

or

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

Saves configuration changes.

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

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

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

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

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

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

Step 8 **show monitor-session [session-name] status** [**detail**] [**error**]

Example:

```
RP/0/RP0/CPU0# show monitor-session status
```

Displays information about the monitor session.

Introduction to ERSPAN rate limit

ERSPAN rate limit is an ERSPAN feature used to control the amount of mirrored traffic being sent over the network to an ERSPAN destination. By setting a specific rate limit, you can prevent network congestion and ensure that the ERSPAN traffic does not overload the network infrastructure.

With rate limiting, you can limit the amount of traffic to a specific rate, which prevents the network and remote ERSPAN destination traffic overloading. If the rate-limit exceeds, then the system may cap or drop the monitored traffic.

This feature enables you monitor traffic flow through any IP network. This includes third-party switches and routers.

ERSPAN operates in the following modes:

- ERSPAN Source Session – box where the traffic originates (is SPANned).
- ERSPAN Termination Session or Destination Session – box where the traffic is analyzed.

You can configure the QoS parameters on the traffic monitor session.

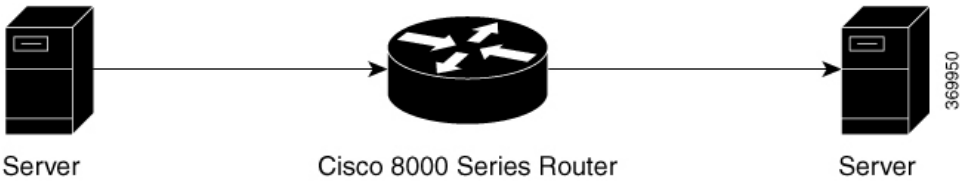
- Traffic Class (0 through 7)
  - Traffic class 0 has the lowest priority and 7 the highest.
  - The default traffic class is the same as that of the original traffic class.

Benefits

With ERSPAN rate limit feature, you can limit the mirrored traffic and use the mirrored traffic for data analysis.

Topology

Figure 9: Topology for ERSPAN Rate Limit



The encapsulated packet for ERSPAN is in ARPA/IP format with GRE encapsulation. The system sends the GRE tunneled packet to the destination box identified by an IP address. At the destination box, SPAN-ASIC decodes this packet and sends out the packets through a port. ERSPAN rate limit feature is applied on the router interface to rate limit the monitored traffic.

The intermediate switches carrying ERSPAN traffic from source session to termination session can belong to any L3 network.

Configure ERSPAN Rate Limit

Use the following steps to configure ERSPAN rate limit:

```
monitor-session ERSPAN ethernet
destination interface tunnel-ip1
!

RP/0/RP0/CPU0:pyke-008#sh run int tunnel-ip 1

interface tunnel-ip1
ipv4 address 4.4.4.1 255.255.255.0
tunnel mode gre ipv4
tunnel source 20.1.1.1
tunnel destination 20.1.1.2
!

RP/0/RP0/CPU0:pyke-008#sh run int hundredGigE 0/0/0/16

interface HundredGigE0/0/0/16
ipv4 address 215.1.1.1 255.255.255.0
ipv6 address 3001::2/64
monitor-session ERSPAN ethernet direction rx-only port-level
  acl
!
ipv4 access-group ACL6 ingress
```

Running Configuration

```
!!A traffic class needs to be configured under the monitor session.
monitor-session mon2 ethernet
destination interface tunnel-ip30
traffic class 5

A shaper needs to be configured for this traffic class:

policy-map m8
class TC1
  bandwidth percent 11
 !
 class TC2
```

```
 bandwidth percent 12
!
class TC3
 bandwidth percent 13
!
class TC4
 bandwidth percent 14
!
class TC5
 shape average percent 15
!
class TC6
 bandwidth percent 16
!
class TC7
 bandwidth percent 17
```

```
This policy-map has to be installed on the interface over which the mirrored traffic is
sent in the egress direction:
interface TenGigE0/6/0/9/0
service-policy output m8
```

Verification

```
RP/0/RP0/CPU0:ios#show monitor-session FOO status detail
Wed May 2 15:14:05.762 UTC
Monitor-session FOO
Destination interface tunnel-ip100
Source Interfaces
-----------------
TenGigE0/6/0/4/0
Direction: Both
Port level: True
ACL match: Disabled
```

Introduction to Local SPAN

Local SPAN overview

Local SPAN is the most basic form of traffic mirroring. In Local SPAN, both mirror source and mirror destination interfaces are present on the same router.

Local SPAN Supported Capabilities

The following capabilities are supported for Local SPAN:

- Only ingress traffic.
- The destination interface can only be an L2 or L3 physical main interface.
- The following interfaces are configured as sources for a Local SPAN session:
  - L3 physical main and sub-interface and bundle main and sub-interface.
  - L2 ethernet interfaces: Ethernet Flow Point (EFP) and trunk
  - BVI interface

- The following types of traffic are mirrored Local SPAN:
  - IPv4, IPv6, and MPLS
  - IP-in-IP
- Extended ACL to reduce mirrored traffic throughput
- Traffic shaping on the destination interface
- Session statistics. There's one counter for all types of traffic, that is, IPv4, IPv6, and MPLS.
- Up to four Local SPAN sessions. This session number is shared between ERSPAN, Local SPAN, and SPAN to File features.
- Up to 1000 source interfaces

Local SPAN Restrictions

The following are the restrictions for Local SPAN:

- Egress mirroring isn't supported.
- The physical interface used as destination can't be a bundle member link.
- GRE tunnels are not supported as source or destination interfaces.
- Per-source interface mirroring statistics isn't supported. However, SPAN session statistics are supported. The session statistics would contain total number of packets mirrored by the session.
- A destination interface can't be a mirrored source interface and vice versa.
- ACLs for Local SPAN are applied only in ingress direction.
- If the ACL keyword is present in monitor-session configuration for an interface but no ACL is applied to that interface, traffic packets are not mirrored.
- ACL for MPLS traffic isn't supported.
- NetFlow or sFlow configuration is not supported on interfaces that already have a Local SPAN session configured.
- The dropped packets at NPU cannot be captured by regular SPAN session. For capturing dropped packets at NPU, use Mirroring forward-drop packets, on page 187 feature.

Configuring Local SPAN

Configuring Local SPAN consists of 2 parts:

1. Creating a local SPAN session

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#monitor-session mon1 ethernet
RP/0/RP0/CPU0:router(config-mon)#destination interface HundredGigE0/1/0/0
RP/0/RP0/CPU0:router(config-mon)#commit
RP/0/RP0/CPU0:router(config-mon)#end
RP/0/RP0/CPU0:router#
```

2. Attaching the SPAN session to an interface

```
RP/0/RP0/CPU0:router(config-mon)#interface HundredGigE0/1/0/2
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)# no shut
RP/0/RP0/CPU0:router(config-if)#!
RP/0/RP0/CPU0:router(config-if)#
RP/0/RP0/CPU0:router(config-if)#interface Bundle-Ether1
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)# no shutdown
RP/0/RP0/CPU0:router(config-if)#!
RP/0/RP0/CPU0:router(config-if)#
```

```
RP/0/RP0/CPU0:monitor(config-if)#
RP/0/RP0/CPU0:monitor(config-if)#interface HundredGigE0/1/0/14.100
RP/0/RP0/CPU0:monitor(config-subif)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:monitor(config-if-mon)# no shut
RP/0/RP0/CPU0:monitor(config-subif)#!
RP/0/RP0/CPU0:monitor(config-subif)#
RP/0/RP0/CPU0:monitor(config-subif)#interface Bundle-Ether1.1
RP/0/RP0/CPU0:monitor(config-subif)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:monitor(config-if-mon)# no shut
RP/0/RP0/CPU0:monitor(config-subif)#!
RP/0/RP0/CPU0:monitor(config-subif)#commit
```

Verification

```
RP/0/RP0/CPU0:router#show monitor-session status
Monitor-session mon1
Destination interface HundredGigE0/1/0/0
=========================================
Source Interface      Dir   Status
--------------------- ----  ----------------
Hu0/1/0/2             Rx    Operational
Hu0/1/0/14.100        Rx    Operational
BE1                   Rx    Operational
BE1.1                 Rx    Operational
```

Execute the `show monitor-session status internal` command for session statistics:

```
RP/0/RP0/CPU0:router#show monitor-session status internal
Thu Aug 13 20:05:23.478 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon1 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface HundredGigE0/1/0/0 (0x00800190)
          Last error: Success
0/1/CPU0: Destination interface HundredGigE0/1/0/0 (0x00800190)
0/RP0/CPU0: Destination interface HundredGigE0/1/0/0 (0x00800190)
Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/1/CPU0: Name 'mon1', destination interface HundredGigE0/1/0/0 (0x00800190)
Platform, 0/1/CPU0:

  Monitor Session ID: 1
  Monitor Session Packets: 32
  Monitor Session Bytes: 4024

0/2/CPU0: Name 'mon1', destination interface HundredGigE0/1/0/0 (0x00800190)
Platform, 0/2/CPU0:

  Monitor Session ID: 1
  Monitor Session Packets: 0
  Monitor Session Bytes: 0
```

Local SPAN with ACL

Local SPAN with ACL is used to filter and mirror ingress traffic. Only Access Control Entries (ACEs) with `capture` keyword are considered for mirroring. Both permit and deny packets are captured if the ACE contains `capture` keyword. Per interface, only one IPv4 ingress ACL and one IPv6 ingress ACL is allowed.

Configuring Local SPAN with ACL

Use the following configuration to enable local SPAN with IPv4 ACLs:

1. Configure ACLs for traffic mirroring.

```
Router(config)# ipv4 access-list acl1
Router(config-ipv4-acl)# 10 permit ipv4 25.0.0.0 0.0.0.255 any capture
Router(config-ipv4-acl)# 20 permit ipv4 20.0.0.0 0.0.0.255 any
Router(config-ipv4-acl)# 30 permit ipv4 131.1.1.0 0.0.0.255 any capture
Router(config-ipv4-acl)# 40 permit ipv4 191.1.1.0 0.0.0.255 any capture
```

2. Apply the traffic monitoring to an interface.

```
Router(config)# interface HundredGigE0/1/0/2
Router(config-if)# ipv4 address 131.1.1.2 255.255.255.0
Router(config-if)# monitor-session mon1 ethernet direction rx-only port-level
Router(config-if-mon)# acl
Router(config-if-mon)# ipv4 access-group acl1 ingress
```

Verification

```
RP/0/RP0/CPU0:ios#show running-config ipv4 access-list acl1
Thu Aug 13 20:22:54.388 UTC
ipv4 access-list acl1
 10 permit ipv4 22.0.0.0 0.0.0.255 any capture
 20 permit ipv4 20.0.0.0 0.0.0.255 any
 30 permit ipv4 131.1.1.0 0.0.0.255 any capture
 40 deny ipv4 181.1.1.0 0.0.0.255 any capture
!
```

Use the following configuration to enable local SPAN with IPv6 ACLs:

1. Configure ACLs for traffic mirroring.

```
Router(config)# ipv6 access-list acl2
Router(config-ipv6-acl)# 10 permit ipv6 10:1:1::2/64 any capture
Router(config-ipv6-acl)# 20 permit ipv6 10:1:1::3/64 any
Router(config-ipv6-acl)# 30 permit ipv6 10:1:1::4/64 any capture
```

2. Apply the traffic monitoring to an interface.

```
Router(config)# interface HundredGigE0/1/0/3
Router(config-if)# ipv6 address 10:1:1::5/64
Router(config-if)# monitor-session mon2 ethernet direction rx-only port-level
Router(config-if-mon)# acl
Router(config-if-mon)# ipv6 access-group acl2 ingress
```

Verification

```
RP/0/RP0/CPU0:ios#show running-config ipv6 access-list acl2
Thu Aug 14 20:22:54.388 UTC
ipv6 access-list acl2
 10 permit ipv6 10:1:1::2/64 any capture
 20 permit ipv6 10:1:1::3/64 any
 30 permit ipv6 10:1:1::4/64 any capture
 !
```

Local SPAN Rate Limit

Local SPAN rate limiting takes place at the session level and not at source interface level. For rate limiting, local SPAN session should configure a traffic class. This traffic class is used to shape traffic on an egress interface. A QoS policy is applied to the egress interface over which mirrored traffic is sent.

Example for Local SPAN Rate Limit Configuration

```
Router# monitor-session mon2 ethernet
destination interface HundredGigE0/1/0/19
 traffic-class 5

class-map match-any TC5
 match traffic-class 5
 end-class-map

policy-map shape-foo
 class TC5  /* This has to match the class that was configured on monitor session */
 shape average percent 15
 class class-default

interface HundredGigE0/1/0/19    /* This is the egress interface over which mirrored packets
 are sent */
 service-policy output shape-foo
```

Traffic Mirroring with DSCP

Differentiated Service Code Point (DSCP) value of Differentiated Services (DS) field in IP packet is used to classify the traffic in the network. DS field formerly known as Type of Service (ToS).You can set the DSCP value in the six most significant bits of the differentiated services (DS) field of the IP header, thereby giving 2^6 = 64 different values (0 to 63). These six bits affect the Per Hop Behavior (PHB) and hence affects how a packet is moved forward. The default value of DSCP is zero (0). DSCP was defined under RFC 2474.

Following the principle of traffic classification, DSCP places a particular packet into a limited number of traffic classes. Similarly, the router is also informed about the DSCP values and the router can prioritize thepacket in traffic flow.

Refer the table to know more about the service class names defined in RFC 2474.

Table 26: DSCP, DS, and ToS values

DSCPValue in Decimal	DS Binary	DSHex	DSCPName	DS/ToSValue	ServiceClass
0	000000	0x00	DF/CS0	0	Standard
-	-	-	none	2	
1	000001	0x01	None	4	
1	000001	0x01	LE	4	Lower-effort
2	000010	0x02	None	8	
4	000100	0x04	None	16	
8	001 000	0x08	CS1	32	Low-priority data
10	001 010	0x0a	AF11	40	High-throughput data

12	001 100	0x0c	AF12	48	High-throughput data
14	001 110	0x0e	AF13	56	High-throughput data
16	010 000	0x10	CS2	64	OAM
18	010 010	0x12	AF21	72	Low-latency data
20	010 100	0x14	AF22	80	Low-latency data
22	010 010	0x16	AF23	88	Low-latency data
24	011 000	0x18	CS3	96	Broadcastvideo
26	011 000	0x1a	AF31	104	Multimedia streaming
28	011 100	0x1c	AF32	112	Multimedia streaming
30	011 110	0x1e	AF33	120	Multimedia streaming
32	100 000	0x20	CS4	128	Real-timeinteractive
34	100 010	0x22	AF41	136	Multimedia conferencing
36	100 100	0x24	AF42	144	Multimedia conferencing
38	100 110	0x26	AF43	152	Multimedia conferencing
40	101 000	0x28	CS5	160	Signaling(IP telephony, etc)
44	101 100	0x2c	Voice-admit	176	
46	101 110	0x2e	EF	184	Telephony
48	110 000	0x30	CS6	192	Networkrouting control
56	111 000	0x38	CS7	224	“reserved”

DSCP marking on egress GRE tunnel in ERSPAN

DSCP marking on egress GRE tunnel in ERSPAN is a mechanism used to classify and manage network traffic by assigning different priority levels to packets. Configuring the DSCP marking on an egress GRE tunnel for ERSPAN traffic, enables you to define the Quality of Service (QoS) for those mirrored packets.

Table 27: Feature History Table

Feature Name	Release Information	Feature Description
DSCP marking on egress GRE tunnel in ERSPAN	Release 7.5.4	You can now set or modify Differentiated Service Code Point (DSCP) value on the ERSPAN GRE tunnel header. This feature allows you to control the QoS for your network's ERSPAN GRE tunnel traffic and eases the effort to control your customers' bandwidth across next-hop routers.

Starting Cisco IOS XR Software Release 7.5.4, you can set or modify the DSCP marking on the ERSPAN GRE tunnels. ERSPAN uses GRE encapsulation to route captured traffic.

Configure DSCP Marking on Egress GRE Tunnel in ERSPAN

Configuration Example

This example shows how you can configure DSCP Marking on Egress GRE tunnel in ERSPAN.

```
Router#configure terminal
Router(config)#interface tunnel-ip1
Router(config-if)#tunnel tos 96
Router(config-if)#tunnel mode gre ipv4
Router(config-if)#tunnel source 192.0.2.1
Router(config-if)#tunnel destination 192.0.2.254
Router(config-if)#commit
```

Note You can configure DSCP value on both IPv4 and IPv6 headers.

Running Configuration

```
interface tunnel-ip1
 tunnel tos 96
 tunnel mode gre ipv4
 tunnel source 192.0.2.1
 tunnel destination 192.0.2.254
!
```

Verification

You can use the following commands to verify that ToS value is configured:

```
Router#show run interface tunnel-ip 1
interface tunnel-ip1
 ipv4 address 192.0.2.0/24
 tunnel tos 96
 tunnel mode gre ipv4
 tunnel source 192.0.2.1
 tunnel vrf red
 tunnel destination 192.0.2.254
```

```
Router#show monitor-session ERSPAN-2 status internal
Information from SPAN Manager and MA on all nodes:
Monitor-session ERSPAN-2 (ID 0x00000003) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip1 (0x20008024)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 192.0.2.1
            Dest IP: 192.0.2.254
            VRF: red
            VRF TBL ID: 0
            ToS: 96
            TTL: 255
            DFbit: Not set
```

DSCP bitmask to filter ingress ERSPAN traffic

DSCP bitmask to filter ingress ERSPAN traffic is a mechanism used to filter ingress ERSPAN traffic with a specific DSCP value. The router matches the bitmask found in the ACL rule with the DSCP field in the IP packet header. The result determines whether the packet matches the desired bitmask for classifying and prioritizing traffic as it enters the network.

Table 28: Feature History Table

Feature Name	Release Information	Feature Description
DSCP bitmask to filter ingress ERSPAN traffic	Release 7.5.4	You can now mirror multiple traffic flows for matched Differentiated Service Code Point (DSCP) value of IP header on the Encapsulated remote SPAN (ERSPAN). The matched DSCP value is based on the DSCP value and the bitmask configured in Access Control List (ACL) rule. Earlier, you could monitor single traffic flow by setting the RFC 4594 defined DSCP values in the GRE tunnel header. This feature introduces the following changes: • **CLI:** deny (IPv4), deny (IPv6), permit (IPv4), and permit (IPv6) are modified to include new keyword **bitmask**. • **YANG DATA Model:** New XPaths for Cisco-IOS-XR-um-ipv4-access-list-cfg and Cisco-IOS-XR-um-ipv6-access-list-cfg (see Github, YANG Data Models Navigator).

Starting Release 7.5.4, You can configure an ACL rule with DSCP bitmask on the ERSPAN GRE tunnels to mirror specific traffic flows.

Without ACL rule, ERSPAN mirrors all the traffic on the incoming port. When ACL is configured with DSCP and DSCP mask on the ERSPAN, ERSPAN mirrors the traffic whose DSCP value lies within the combination of DSCP value and the specified mask.

A DSCP value is mapped to a single traffic class as per the defined value in RFC2474. Masking the DSCP value in ACL rule allows to mirror multiple traffic flows. DSCP value and mask operate similar to IPv4 address and mask.

Configure DSCP Bitmask to Filter Ingress ERSPAN Traffic

To configure DSCP bitmask, use the bitmask option along with the dscp option while configuring the ACL.

Configuration Example for IPv4

This example shows how you can configure DSCP bitmask on ingress ERSPAN for IPv4 traffic.

```
/*configure the ACL*/
Router# config
Router(config)# ipv4 access-list acl1
Router(config-ipv4-acl)# 10 permit ipv4 host 192.0.2.1 any dscp af22 bitmask 0x3f
Router(config-ipv4-acl)# commit
```

```
Router(config-ipv4-acl)# exit

/* Perform the following configurations to attach the created ACL to an interface*/
Router(config)# interface HundredGigE0/0/0/6
Router(config-if)# ipv4 address 192.0.2.51 255.255.255.0

/* Monitor the ingress ACL applied and DSCP masked IPv4 traffic on ERSPAN*/
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv4 acl1
Router(config-if)# commit
```

Running Configuration

```
Router(config)# show running-config ipv4 access-list
ipv4 access-list acl1
 10 permit ipv4 host 192.0.2.1 any dscp af22 bitmask 0x3f
!

interface HundredGigE0/0/0/6
 ipv4 address 192.0.2.51 255.255.255.0
 monitor-session TEST ethernet direction rx-only port-level  acl ipv4 acl1
!
!
```

Configuration Example for IPv6

This example shows how you can configure DSCP bitmask on ingress ERSPAN for IPv6 traffic.

```
/*configure the ACL*/
Router# config
Router(config)# ipv6 access-list acl1
Router(config-ipv6-acl)# 10 permit ipv6 host 2001:DB8::2/32 any dscp 33 bitmask 0x3f
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit

/* Perform the following configurations to attach the created ACL to an interface*/
Router(config)# interface HundredGigE 0/0/10/3
Router(config-if)# ipv6 address 2001:DB8::1/32

/* Monitor the ingress ACL applied and DSCP masked IPv4 traffic on ERPSAN*/
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv6 acl1
Router(config-if)# commit
```

Running Configuration

```
Router(config)# show running-config ipv6 access-list
ipv6 access-list acl1
 10 permit ipv6 acl1 host 2001:DB8::2/32 any dscp 33 bitmask 0x3f
!
interface HundredGigE0/0/10/3
 ipv6 address 2001:db8::1/32
 monitor-session TEST ethernet direction rx-only port-level acl ipv6 acl1
!
!
```

Monitor multiple ERSPAN sessions with SPAN and security ACL

Table 29: Feature History Table

Feature Name	Release Information	Feature Description
Monitor multiple ERSPAN sessions with SPAN and security ACL	Release 7.5.4	With this feature, you can use SPAN and security ACL together to monitor multiple ERSPAN sessions under the same source interface. SPAN ACL helps you to distribute the mirrored traffic over different destination interfaces and Security ACL helps you to allow selective incoming traffic.

Starting Cisco IOS XR Software Release 7.5.4 you can monitor multiple ERSPAN sessions using GREv4 and GREv6 under the same source interface. Multiple ERSPAN monitor sessions configured on an interface allow you to choose the destination interface for the mirrored traffic. For the configuration of monitor sessions, you can use SPAN and security ACLs together. The SPAN and security ACLs are applicable only in the ingress traffic.

Configure Multiple Monitor ERSPAN Sessions with SPAN and Security ACL

This example shows how to configure SPAN and Security ACL for SPAN with GREv4 and GREv6 Monitor Sessions.

Configuration example

Use the following configuration to attach SPAN and security ACLs for traffic mirroring.

```
Router# config
/*Perform the following configurations to attach the SPAN ACL to an interface*/
Router(config-if)#monitor-session always-on-v4 ethernet direction rx-only port-level
Router(config-if-mon)#acl ipv4 v4-monitor-acl1
Router(config-if-mon)#acl ipv6 v6-monitor-acl1
Router(config-if-mon)#exit
Router(config-if)#monitor-session on-demand-v4 ethernet direction rx-only port-level
Router(config-if-mon)#acl ipv4 v4-monitor-acl2
Router(config-if-mon)#acl ipv6 v6-monitor-acl2
Router(config-if-mon)#exit
/*Perform the following configurations to attach the security ACL to an interface*/
Router(config-if)#ipv4 access-group sec_aclv4 ingress
Router(config-if)#ipv6 access-group sec_aclv6 ingress
Router(config-if)#commit
```

Running configuration

```
Router(config)#show running-config interface
monitor-session always-on-v4 ethernet direction rx-only port-level
```

```
  acl ipv4 v4-monitor-acl2
  acl ipv6 v6-monitor-acl2
!
monitor-session on-demand-v4 ethernet direction rx-only port-level
  acl ipv4 v4-monitor-acl2
  acl ipv6 v6-monitor-acl2
!
ipv4 access-group sec_aclv4 ingress
ipv6 access-group sec_aclv6 ingress
!
!
```

SPAN to file

SPAN to file is a network monitoring feature that allows the captured traffic from a SPAN session to be written directly to a file for later analysis.

Table 30: Feature History Table

Feature name	Release information	Feature description
SPAN-to-file support in Tx and Rx direction	Release 7.5.3	With this feature, the ability to capture the packet in Tx direction along with the ability to store the capture on the file is supported. You can now capture the packet in the Tx direction and store the capture on the file. Earlier, you could only capture or mirror the traffic in the Rx direction. You now have the flexibility to choose Tx, Rx, or both directions. You can now capture and analyze the outgoing (Tx) packets.
Partial packet capture ability for SPAN-to-file (Rx)	Release 7.5.3	With this feature, you can perform partial packet capture in the Rx direction. Earlier, the ability for entire packet capture was available in the Tx direction only, now you can choose entire or partial packet capture in the Rx direction also. Here, partial packet capture is also known as truncation.

Feature name	Release information	Feature description
SPAN-to-file PCAPng file format	Release 7.3.1	PCAPng is the next generation of packet capture format that contains a dump of data packets captured over a network and stored in a standard format. The PCAPng file contains different types of information blocks, such as the section header, interface description, enhanced packet, simple packet, name resolution, and interface statistics. These blocks can be used to rebuild the captured packets into recognizable data. The PCAPng file format: • Provides the capability to enhance and extend the existing capabilities of data storage over time • Allows you to merge or append data to an existing file. • Enables to read data independently from network, hardware, and operating system of the machine that made the capture.

SPAN to File is an extension of the pre-existing SPAN feature that allows network packets to be mirrored to a file instead of an interface. This helps in the analysis of the packets at a later stage. The file format is PCAP, which helps that data to be used by tools, such as tcpdump or Wireshark.

Note A maximum of 100 source ports are supported across the system. Individual platforms may support lower numbers. All the SPAN sessions are configured under the Ethernet class. At any given time, the system supports four SPAN to File sessions.

When you configure a file as a destination for a SPAN session, the system creates buffer on each node to which the network packets are logged. The buffer is for all packets on the node regardless of which interface they are from. That is, multiple interfaces can provide packets to the same buffer. The system deletes the buffer when the session configuration is removed. Each node writes a file on the active RP, which contains the node ID of the node on which the buffer was located.

The minimum buffer size is 1KB. The maximum buffer size is 1000KB and default buffer size is 2KB.

If multiple interfaces are attached to a session, then interfaces on the same node are expected to have their packets sent to the same file. Bundle interfaces can be attached to a session with a file destination, which is similar to attaching individual interfaces.

From Cisco IOS XR Software Release 7.5.3 onwards, the capture of all the outgoing packets from the router is supported.

Earlier to Cisco IOS XR Software Release 7.5.3, there was no functionality which enables to capture the payload of packets coming from your customers for security reasons.

Limitations and restrictions for SPAN to File

- Only incoming packet mirroring on the source interface is supported. Outgoing mirrored packets cannot be dumped to the file.

However, from Cisco IOS XR Software Release 7.5.3 onwards, there are no restrictions.

- SPAN ACLs can only be applied in ingress direction only. Hence, ACLs for SPAN to File can only be applied in ingress direction only.
- ACL on MPLS traffic is not supported.
- MPLS over GRE traffic is supported, however, GRE interfaces cannot be configured as source interfaces.
- Packet truncation applies for SPAN to File and ERSPAN interfaces only. If you change the destination to Local SPAN, then an **ios_msg** is displayed as a warning. The entire packet is mirrored after this message is displayed.

  Example: `The Partial Packet Capture feature is not supported by Local SPAN. The entire Packet will be mirrored.`
- Packet truncation is per monitor session.
- Currently, truncation per interface is not supported.
- For outgoing (TX) SPAN to File, Security ACL is not supported.
- For outgoing (TX) SPAN to File, only transit traffic is mirrored.

  Self-originating traffic cannot be mirrored.

Supported capabilities for SPAN to File

- The ability to mirror outgoing traffic and punt it to the CPU across all NPU versions.
- Ability to mirror outgoing IPv4, IPv6, and MPLS traffic to file.
- Ability to mirror outgoing traffic across all types of L3 interfaces, including physical, sub, bundle, and bundle sub interfaces
- Ability to mirror outgoing traffic across L2 or BVI interfaces.
- Ability to enable the new SPAN to File truncation configuration for both RX and TX direction. You can specify the `both` keyword to enable RX and TX mirroring on a single source interface.

  See Configuring SPAN to File for Truncation and Direction, on page 186
- Ability to configure a different truncation size on each monitor session.
- Ability to configure SPAN to File mirroring packet truncation size from 1 to 10000. If you try to configure a value out of the range, the configuration will not accept it and displays an error message.
- Ability to change the truncation size, when packet collecting has stopped. Removing or re-adding the monitor session is not required.
- Ability to change the truncation size during packet collecting ON. Not required to stop the monitor session.
- The entire packet is mirrored by default, without the mirror first (truncation size) configuration.

  Also, if the packet size is less than the configured truncation size, the entire packet is mirrored.

Action commands for SPAN to File

Action commands allows you to start and stop network packet collection. You can run the action commands on sessions where the destination is a file. The action command autocompletes names of the globally configured SPAN to File sessions. The following table provides more information on action commands.

Table 31: Action commands for SPAN to File

Action	Command	Description
Start	`monitor-session <name> packet-collection start`	Use this command to start writing packets for the specified session to the configured buffer.
Stop	`monitor-session <name> packet-collection stop [ discard-data | write directory <dir> filename <filename> ]`	Use this command to stop writing packets to the configured buffer. If you specify the `discard-data` option, the system clears the buffer. If you specify the `write` option, the system writes the buffer to disk before clearing it. When writing the buffer to disk, save the file in .pcap format at the following location: `/<directory>/<node_id>/<filename>`. If you include a .pcap extension when specifying the filename, the system will remove it to prevent the extension from being added twice.

Configuring SPAN to File

Use the following command to configure SPAN to File:

```
monitor-session <name> [ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]
    destination file [size <kbytes>] [buffer-type linear]
```

The `monitor-session <name> [ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]` part of the command creates a monitor-session with the specified name and class and is a pre-existing chain point from the current SPAN feature. The `destination file [size <kbytes>] [buffer-type linear]` part of the command adds a new “file” option to the existing “destination”.

`destination file` has the following configuration options:

- Buffer size.
- Two types of buffer:
  - Circular: Once the buffer is full, the start is overwritten.
  - Linear: Once the buffer is full, no further packets are logged.

Note The default buffer-type is circular. Only linear buffer is explicitly configurable. Changing any of the parameters (buffer size or type) recreates the session, and clears any buffers of packets.

All configuration options which are applied to an attachment currently supported for other SPAN types should also be supported by SPAN to file. This may include:

- ACLs
- Write only first X bytes of packet.
- In Cisco IOS XR Release 7.5.3, truncation per global session is supported and not per interface.

Note These options are implemented by the platform when punting the packet.

Once a session has been created, then interfaces may be attached to it using the following configuration:

```
interface GigabitEthernet 0/0/0/0
    monitor-session <name> [ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]
```

The attachment configuration is unchanged by SPAN to File feature.

Configuration Examples

To configure a `mon1` monitor session, use the following commands:

```
monitor-session mon1 ethernet
          destination file size 230000
         !
```

In the above example, omitting the `buffer-type` option results in default circular buffer.

To configure a `mon2` monitor session, use the following commands:

```
monitor-session mon2 ethernet
          destination file size 1000 buffer-type linear
         !
```

To attach monitor session to a physical or bundle interface, use the following commands:

```
RP/0/RSP0/CPU0:router#show run interface Bundle-Ether 1
Fri Apr 24 12:12:59.348 EDT
interface Bundle-Ether1
monitor-session ms7 ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]
[direction {rx-only|tx-only|both[SW(1] }] [port-level]
acl [<acl_name>]!
```

Running Configuration

```
!! IOS XR Configuration 7.1.1.124I
!! Last configuration change at Tue Nov 26 19:29:05 2019 by root
!
hostname OC
logging console informational
!
monitor-session mon1 ethernet
 destination file size 230000 buffer-type circular
!
monitor-session mon2 ethernet
 destination file size 1000 buffer-type linear
```

```
!
interface Bundle-Ether1
monitor-session ms7 ethernet
          direction rx-only
end
```

Verification

To verify packet collection status:

```
RP/0/RP0/CPU0:router#show monitor-session status
Monitor-session mon1
Destination File - Packet collecting
=============================================
Source Interface        Dir.      Status
--------------------- ---- ------------------
Hu0/9/0/2                    Rx       Operational

Monitor-session mon2
Destination File - Packet collecting
=========================================
Source Interface         Dir    Status
--------------------- ---- --------------
BE2.1.                           Rx     Operational
```

If packet collection is not active, the following line is displayed:

```
Monitor-session mon2
Destination File - Not collecting
```

Configuring SPAN to File for Truncation and Direction

Configuring SPAN to File for Truncation

Use the **mirror first** command in monitor session configuration mode to create a SPAN to File monitor session for mirroring the packets with truncation enabled:

```
monitor-session <name> [ethernet]
destination file [size <kbytes>] [buffer-type linear|circular]
mirror first <number>
```

Once a session has been created, then interfaces may be attached to it using the following configuration:

```
interface <>
  monitor-session session-name ethernet direction rx-only|tx-only|both | acl [acl_name]
```

Configuration Examples

To configure a `mon1` monitor session, use the following commands:

```
monitor-session mon1 ethernet
 destination file
 mirror first 128
!
```

Configuring SPAN to File for Direction

Use the following command to create a SPAN to File monitor session for mirroring the packets:

```
monitor-session mon2 ethernet
 destination file
!
```

Attach the session which has been created to the interfaces using the following configuration:

```
interface <>
  monitor-session session-name ethernet direction rx-only|tx-only|
acl [acl_name]
```

Running Configuration for all

```
monitor-session mon3 ethernet
destination file
!

interface Hu0/9/0/2
monitor-session mon1 ethernet
direction rx-only
!

interface bundle-ether1
monitor-session mon2 ethernet
direction tx-only
!

interface bundle-ether2.1
monitor-session mon3 ethernet
direction both
end
```

Verification

The **show monitor-session status** displays the direction.

```
Router#show monitor-session status
                  Monitor-session mon1
                  Destination File - Packet collecting
                  ============================================
                  Source Interface      Dir   Status
                  --------------------- ----  ------------------
                  Hu0/9/0/2             Rx    Operational
                  Monitor-session mon2
                  Destination File - Packet collecting
                  =======================================
                  Source Interface      Dir   Status
                  --------------------- ----  --------------
                  BE1                  Tx    Operational
                  Monitor-session mon3
                  Destination File - Packet collecting
                  =======================================
                  Source Interface      Dir   Status
                  --------------------- ----  --------------
                  BE2.1                Both    Operational
```

Mirroring forward-drop packets

Mirroring forward-drop packets is a network monitoring feature that captures and analyzes packets that a router drops while forwarding them.

Table 32: Feature History Table

Feature Name	Release Information	Description
Mirroring forward-drop packets	Release 7.5.4	Mirroring forward-drop packets feature copies or mirrors the packets that are dropped during the forwarding process at the router ingress to a configured destination. These mirrored packets can be captured and analyzed using network monitoring tools. The analysis of dropped packets helps you understand the types of traffic that are blocked, analyze potential security threats, troubleshoot, and optimize network performance. This feature introduces the following changes: • **CLI: forward-drop rx** • **YANG Data Model:** New XPath for Cisco-IOS-XR-um-monitor-session-cfg.yang (see GitHub, YANG Data Models Navigator)

In a network, packets are forwarded from one device to another until they reach their destination. However, in some cases, routers may drop packets during this forwarding process. These packets are known as forward-drop packets.

The packet drop can happen for several reasons, such as congestion on the network, errors in the packet header or payload, blocking by firewall or access control lists (ACL), and so on. These forward-drop packets are typically discarded before they can reach their intended destination, and may have to be re-transmitted by the source device. This feature supports mirroring of these forward-drop packets at the ingress (Rx direction) to another destination. When a global forward-drop session is configured for the router, the forward-drop packets at the ingress are mirrored or copied to the configured destination. You can configure the mirror destination as a file (for SPAN-to-file sessions) or an IPv4 GRE tunnel ID (for ERSPAN) or sFlow.

Mirroring forward-drop packets to a suitable destination for analysis can help in the following:

- Network visibility: By mirroring and analyzing forward-drop packets, network administrators gain better visibility into the types of traffic that are blocked by the firewalls and access control lists (ACL).
- Threat detection: As the original dropped packet is forwarded without any change, it helps in identifying the source of potential security threats.
- Troubleshooting: Analyzing forward-drop packets helps in troubleshooting network issues that may be causing the packet drop. This helps in taking proactive measures to avoid escalation of the issue.

Guidelines and restrictions for mirroring forward-drop packets

- Only one global forward-drop session can be configured on a router.
- In-band traffic destined to router management interface cannot be captured using this functionality.
- For ERSPAN sessions that monitor forward-drop packets, a default value of 0 is used for the encapsulation traffic class, irrespective of the DSCP value assigned for the tunnel.
- ERSPAN counters are not updated for forward-drop packets.
- Not all packets that are dropped by NPU will be mirrored.

Configuring Forward-Drop

Perform the following tasks on the router to configure a global session for mirroring forward-drop packets:

1. Configure the tunnel mode.
2. Configure the tunnel source.
3. Configure the tunnel destination.
4. Configure a traffic mirroring session.
5. Associate a destination interface with the traffic mirroring session.
6. Run **forward-drop rx** command to start mirroring forward-drop packets.

Note Forward-drop can be configured using either ERSPAN or sFlow. However, it is recommended not to enable both features simultaneously, as this may lead to router instability.

This example shows how to configure a global traffic mirroring session for forward-drop packets.

```
Router(config)# interface tunnel-ip 2
Router(config-if)# tunnel mode gre ipv4
Router(config-if)# tunnel source 20.20.20.20
Router(config-if)# tunnel destination 192.1.1.3
Router(config-if)# exit
Router(config)# monitor-session mon2 ethernet
Router(config)# destination interface tunnel-ip2
Router(config)# forward-drop rx
Router(config)# commit
```

Running Configuration

This section shows forward-drop running configuration.

```
RP/0/RSP0/CPU0:router# show running-config
interface tunnel-ip 2
tunnel mode gre ipv4
tunnel source 20.20.20.20
tunnel destination 192.1.1.3
!
monitor-session mon2 ethernet
destination interface tunnel-ip2
forward-drop rx
!
```

Verification

Verify the forward-drop packets are mirrored using the **show monitor-session** command.

```
Router# show monitor-session mon2 status detail
Mon Aug 15 19:14:31.975 UTC
Monitor-session mon2
   Destination interface tunnel-ip2
   All forwarding drops:
      Direction: Rx
Source Interfaces
-----------------
```

Introduction to file mirroring

Table 33: Feature History Table

Feature Name	Release Information	Description
File mirroring	Release 7.0.14	This feature enables the router to copy files and directories automatically from an active RP to a standby RP thus eliminating the manual intervention or the use of EEM scripts.

Prior to Cisco IOS XR Software Release 7.2.1 7.0.14, the router did not support file mirroring from active RP to standby RP. Administrators had to manually perform the task or use EEM scripts to sync files across active RP and standby RP. Starting with Cisco IOS XR Software Release 7.0.14, file mirroring feature enables the router to copy files or directories automatically from `/harddisk:/mirror` location in active RP to `/harddisk:/mirror` location in standby RP or RSP without user intervention or EEM scripts.

Two new CLIs have been introduced for the file mirroring feature:

- **`mirror enable`**

  The `/harddisk:/mirror` directory is created by default, but file mirroring functionality is only enabled by executing the `mirror enable` command from configuration terminal. Status of the mirrored files can be viewed with `show mirror status` command.

- **`mirror enable checksum`**

  The `mirror enable checksum` command enables MD5 checksum across active to standby RP to check integrity of the files. This command is optional.

Limitations

The following limitations apply to file mirroring:

- Supported only on Dual RP systems.
- Supports syncing only from active to standby RP. If files are copied into standby `/harddisk:/mirror` location, it won't be synced to active RP.
- A slight delay is observed in `show mirror` command output when mirror checksum configuration is enabled.
- Not supported on multichassis systems.

Configure File Mirroring

File mirroring has to be enabled explicitly on the router. It is not enabled by default.

```
RP/0/RSP0/CPU0:router#show run mirror

Thu Jun 25 10:12:17.303 UTC
mirror enable
mirror checksum
```

Following is an example of copying running configuration to `harddisk:/mirror` location:

```
RP/0/RSP0/CPU0:router#copy running-config harddisk:/mirror/run_config
Wed Jul  8 10:25:51.064 PDT
Destination file name (control-c to abort): [/mirror/run_config]?
Building configuration..
32691 lines built in 2 seconds (16345)lines/sec
[OK]
```

Verification

To verify the syncing of file copied to mirror directory, use the `show mirror` command.

```
RP/0/RSP0/CPU0:router#show mirror
Wed Jul  8 10:31:21.644 PDT
% Mirror rsync is using checksum, this show command may take several minutes if you have
many files. Use Ctrl+C to abort
MIRROR DIR: /harddisk:/mirror/
% Last sync of this dir ended at Wed Jul  8 10:31:11 2020
Location   |Mirrored |MD5 Checksum                      |Modification Time
-------------------------------------------------------------------------
run_config |yes      |76fc1b906bec4fe08ecda0c93f6c7815 |Wed Jul  8 10:25:56 2020
```

If checksum is disabled, `show mirror` command displays the following output:

```
RP/0/RSP0/CPU0:router#show mirror
Wed Jul 8 10:39:09.646 PDT
MIRROR DIR: /harddisk:/mirror/
% Last sync of this dir ended at Wed Jul  8 10:31:11 2020
Location   |Mirrored |Modification Time
-----------------------------------------
run_config |yes         |Wed Jul  8 10:25:56 2020
```

If there is a mismatch during the syncing process, use `show mirror mismatch` command to verify.

```
RP/0/RP0/CPU0:router# show mirror mismatch
Wed Jul  8 10:31:21.644 PDT
 MIRROR DIR: /harddisk:/mirror/
 % Last sync of this dir ended at Wed Jul  8 10:31:11 2020
 Location  |Mismatch Reason     |Action Needed
-----------------------------------------------
 test.txt  |newly created item. |send to standby
```

Traffic Mirroring Configuration Examples

This section contains examples of how to configure traffic mirroring:

Viewing Monitor Session Status: Example

This example shows sample output of the **show monitor-session** command with the **status** keyword:

```
RP/0/RP0/CPU0:router# show monitor-session status

Monitor-session cisco-rtp1
Destination interface HundredGigE0/5/0/38
================================================================================
Source Interface   Dir  Status
---------------------- ---- -------------------------------------------------
Gi0/5/0/4         Rx Operational
Gi0/5/0/17        Rx Operational

RP/0/RP0/CPU0:router# show monitor-session status detail
```

```
Monitor-session sess1
 Destination interface is not configured
 Source Interfaces
 -----------------
 HundredGigE0/0/0/0
  Direction: Rx
  ACL match: Enabled
  Portion:  Full packet
  Status:  Not operational (destination interface not known).
 HundredGigE0/0/0/2
  Direction: Rx
  ACL match: Disabled
  Portion:  First 100 bytes

RP/0/RP0/CPU0:router# show monitor-session status error

Monitor-session ms1
Destination interface HundredGigE0/2/0/15 is not configured
================================================================================
Source Interface   Dir  Status
-------------------- ---- --------------------------------------------------

Monitor-session ms2
Destination interface is not configured
================================================================================
Source Interface   Dir  Status
-------------------- ---- --------------------------------------------------
```

Monitor Session Statistics: Example

The monitor session statistics is provided in the form of packets and bytes. Use the following command to get the status:

```
RP/0/RP0/CPU0:router# show monitor-session <session_name> status internal
RP/0/RP0/CPU0:Router1#show monitor-session mon2 status internal
Wed Oct  9 19:39:30.402 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.1.1.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set
0/1/CPU0: Destination interface tunnel-ip2 (0x0f000034)
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.1.1.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
```

```
0/1/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/1/CPU0:

  Monitor Session ID: 1

  Monitor Session Packets: 11
  Monitor Session Bytes: 1764

0/2/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/2/CPU0:

  Monitor Session ID: 1

  Monitor Session Packets: 0
  Monitor Session Bytes: 0
```

Note

- Currently, the system does not allow you to clear these counters.
- The counters are present on the line-card that contains the interface over which the mirrored packets are sent to the ERSPAN session destination.

  If required, to clear the counters, delete and recreate the monitor session. Also, clear the counters by performing a Shut/No Shut of the tunnel interface, which triggers a Delete+Create action.

Layer 3 ACL-Based Traffic Mirroring: Example

This example shows how to configure Layer 3 ACL-based traffic mirroring:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# monitor-session ms1
RP/0/RP0/CPU0:router(config-mon)# destination tunnel-ip 1

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

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE/2/0/11
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group span ingress
RP/0/RP0/CPU0:router(config-if)# monitor-session ms1 ethernet direction rx-only acl
RP/0/RP0/CPU0:router(config-if-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 access-list span
RP/0/RP0/CPU0:router(config-ipv4-acl)# 5 permit ipv4 any any dscp 5 capture
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any
RP/0/RP0/CPU0:router(config-ipv4-acl)# commit
```

Troubleshooting Traffic Mirroring

When you encounter any issue with traffic mirroring, begin troubleshooting by checking the output of the **show monitor-session status** command. This command displays the recorded state of all sessions and source interfaces:

```
Monitor-session sess1
```

```
<Session status>
================================================================================
Source Interface   Dir  Status
-------------------- ---- ------------------------------------------------------
Gi0/0/0/0       Both <Source interface status>
Gi0/0/0/2       Both <Source interface status>
```

In the preceding example, the line marked as `<Session status>` can indicate one of these configuration errors:

Session Status	Explanation
Session is not configured globally	The session does not exist in global configuration. Check **show run** command output to ensure that a session with a correct name has been configured.
Destination interface <intf> is not configured	The interface that has been configured as the destination does not exist. For example, the destination interface may be configured to be a VLAN subinterface, but the VLAN subinterface may not have been yet created.
Destination interface <intf> (<down-state>)	The destination interface is not in Up state in the Interface Manager. You can verify the state using the **show interfaces** command. Check the configuration to see what might be keeping the interface from coming up (for example, a sub-interface needs to have an appropriate encapsulation configured).

The <Source interface status> can report these messages:

Source Interface Status	Explanation
Operational	Everything appears to be working correctly in traffic mirroring PI. Please follow up with the platform teams in the first instance, if mirroring is not operating as expected.
Not operational (Session is not configured globally)	The session does not exist in global configuration. Check the **show run** command output to ensure that a session with the right name has been configured.
Not operational (destination interface not known)	The session exists, but it either does not have a destination interface specified, or the destination interface named for the session does not exist (for example, if the destination is a sub-interface that has not been created).
Not operational (source same as destination)	The session exists, but the destination and source are the same interface, so traffic mirroring does not work.
Not operational (destination not active)	The destination interface or pseudowire is not in the Up state. See the corresponding *Session status* error messages for suggested resolution.

Source Interface Status	Explanation
Not operational (source state <down-state>)	The source interface is not in the Up state. You can verify the state using the **show interfaces** command. Check the configuration to see what might be keeping the interface from coming up (for example, a sub-interface needs to have an appropriate encapsulation configured).
Error: see detailed output for explanation	Traffic mirroring has encountered an error. Run the **show monitor-session status detail** command to display more information.

The **show monitor-session status detail** command displays full details of the configuration parameters, and of any errors encountered. For example:

```
RP/0/RP0/CPU: router#show monitor-session status detail


Monitor-session sess1
 Destination interface is not configured
 Source Interfaces
 -----------------
 HundredGigE0/0/0/0
  Direction: Both
  ACL match: Enabled
  Portion:  Full packet
  Status:  Not operational (destination interface not known)
 HundredGigE0/0/0/2
  Direction: Both
  ACL match: Disabled
  Portion:  First 100 bytes
  Status: Not operational (destination interface not known). Error: 'Viking SPAN PD' detected
 the 'warning' condition 'PRM connection creation failure'.
Monitor-session foo
 Destination next-hop HundredGigE 0/0/0/0
 Source Interfaces
 -----------------
 HundredGigE 0/1/0/0.100:
  Direction: Both
  Status:  Operating
 HundredGigE 0/2/0/0.200:
  Direction: Tx
  Status:  Error: <blah>

Monitor session bar
 No destination configured
 Source Interfaces
 -----------------
 HundredGigE 0/3/0/0.100:
  Direction: Rx
  Status:  Not operational(no destination)
```

Additional Debugging Commands

Here are additional trace and debug commands:

```
RP/0/RP0/CPU0:router# show monitor-session platform trace ?
```

```
 all   Turn on all the trace
 errors Display errors
 events Display interesting events

RP/0/RP0/CPU0:router# show monitor-session trace ?

 process Filter debug by process

RP/0/RP0/CPU0:router# debug monitor-session platform ?

 all   Turn on all the debugs
 errors VKG SPAN EA errors
 event  VKG SPAN EA event
 info  VKG SPAN EA info

RP/0/RP0/CPU0:router# debug monitor-session platform all

RP/0/RP0/CPU0:router# debug monitor-session platform event

RP/0/RP0/CPU0:router# debug monitor-session platform info

RP/0/RP0/CPU0:router# show monitor-session status ?

 detail  Display detailed output
 errors  Display only attachments which have errors
 internal Display internal monitor-session information
 |     Output Modifiers

RP/0/RP0/CPU0:router# show monitor-session status

RP/0/RP0/CPU0:router# show monitor-session status errors

RP/0/RP0/CPU0:router# show monitor-session status internal
```

If there is no route to the destination IPv4 address, the status displayed for the monitor session looks like this:

```
RP/0/RP0/CPU0:Router1#show monitor-session mon2 status internal
Wed Oct  9 19:24:06.084 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034) (down)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.10.10.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set
0/1/CPU0: Destination interface  is not configured
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.10.10.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set
```

To verify if there is a route to the destination IPv4 address, use the following command:

```
RP/0/RP0/CPU0:Router1#show cef ipv4 130.10.10.2
Wed Oct  9 19:25:12.282 UTC
0.0.0.0/0, version 0, proxy default, default route handler, drop adjacency, internal 0x1001011
```

```
 0x0 (ptr 0x8e88d2b8) [1], 0x0 (0x8ea4d0a8), 0x0 (0x0)
Updated Oct  9 19:03:36.068
Prefix Len 0, traffic index 0, precedence n/a, priority 15
   via 0.0.0.0/32, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8e2db240 0x0]
    next hop 0.0.0.0/32
     drop adjacency
```

When a route is present, the command used in the previous example displays the following:

```
RP/0/RP0/CPU0:Router1#show cef ipv4 130.10.10.2
Wed Oct  9 19:26:06.141 UTC
130.1.1.0/24, version 20, internal 0x1000001 0x0 (ptr 0x8e88aa18) [1], 0x0 (0x8ea4dc68),
0x0 (0x0)
Updated Oct  9 19:26:02.139
Prefix Len 24, traffic index 0, precedence n/a, priority 3
   via 131.1.1.1/32, HundredGigE0/1/0/2, 2 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8f8e2260 0x0]
    next hop 131.10.10.1/32
    local adjacency
```

The show monitor command displays the following:

```
show monitor-session mon2 status internal
Wed Oct  9 19:26:12.405 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.10.10.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set
0/1/CPU0: Destination interface tunnel-ip2 (0x0f000034)
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.10.10.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/1/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/1/CPU0:

  Monitor Session ID: 1

  Monitor Session Packets: 0
  Monitor Session Bytes: 0

0/2/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/2/CPU0:

  Monitor Session ID: 1

  Monitor Session Packets: 0
  Monitor Session Bytes: 0
```

```
Missing ARP to the next hop to the destination
This condition is detected via this show command:
show monitor-session mon2 status internal
```

After resolving ARP for the next hop, which is done by invoking a ping command to the destination, the show command output displays the following:

```
RP/0/RP0/CPU0:Router1#show monitor-session mon2 status internal
Wed Oct  9 19:32:24.856 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.10.10.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set
0/1/CPU0: Destination interface tunnel-ip2 (0x0f000034)
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.10.10.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/1/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/1/CPU0:

  Monitor Session ID: 1
    Monitor Session Packets: 0
  Monitor Session Bytes: 0

0/2/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/2/CPU0:

  Monitor Session ID: 1
    Monitor Session Packets: 0
  Monitor Session Bytes: 0
```

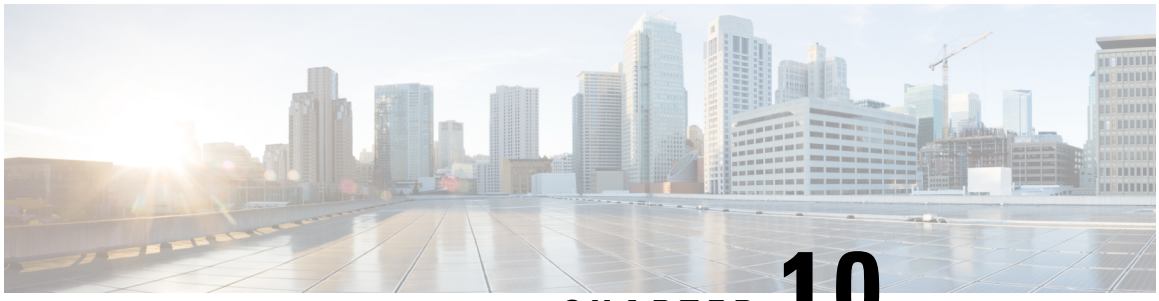
Where to Go Next

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

For information about modifying Ethernet management interfaces for the shelf controller (SC), route processor (RP), and distributed RP, see the Advanced Configuration and Modification of the Management Ethernet Interface later in this document.

For information about IPv6 see the Implementing Access Lists and Prefix Lists on

Cisco IOS XR Software module in the Cisco IOS XR IP Addresses and Services Configuration Guide.



CHAPTER 10

Configuring Virtual Loopback and Null Interfaces

This module describes the configuration of loopback and null interfaces. Loopback and null interfaces are considered virtual interfaces.

A virtual interface represents a logical packet switching entity within the router. Virtual interfaces have a global scope and do not have an associated location. Virtual interfaces have instead a globally unique numerical ID after their names. Examples are Loopback 0, 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 switch processor (RP). The configuration and control plane are mirrored onto the standby RP and, in the event of a failover, the virtual interfaces move to the ex-standby, which then becomes the newly active RP.

Feature History for Configuring Loopback and Null Interfaces on Cisco IOS XR Software

Release	Modification
Release 7.0.11	This feature was introduced.

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 that you need for each command. If you suspect a 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 or active. Any packet that the system transmits over a virtual loopback interface is immediately received by the same interface. Loopback interfaces emulate a physical interface.

In Cisco IOS XR Software, virtual loopback interfaces perform these 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 to 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 that is involved with using access lists by directing undesired network traffic to the null interface.

The only interface configuration command that you can specify for the null interface is the **ipv4 unreachables** command. With the **ipv4 unreachables** command, if the software receives a nonbroadcast packet destined for itself that uses a protocol it does not recognize, it sends an Internet Control Message Protocol (ICMP) protocol unreachable message to the source. If the software receives a datagram that it cannot deliver to its ultimate destination because it knows of no route to the destination address, it replies to the originator of that datagram with an ICMP host unreachable message. By default, the system enables the **ipv4 unreachables** command. If we do not want ICMP to send protocol unreachable, then you need to configure using the **ipv4 icmp unreachable disable** command.

By default, the system creates the Null 0 interface during boot process and you cannot remove it. You can configure the **ipv4 unreachables** command for this interface, but most configuration is unnecessary because this interface just discards all the packets thats the system sends.

Use the **show interfaces null0** command to display the Null 0 interface.

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 switch processor (RP) failover situations. For this to happen, the virtual IPv4 address must share a common IPv4 subnet with a management Ethernet interface on both the RPs.

On a 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 RPs 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 failover, 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 failover. The routers use nonstop forwarding (NSF) over bundles and tunnels through the failover of the host RP.

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

Protocol configuration such as tacacs source-interface, snmp-server trap-source, ntp source, logging source-interface do not use the virtual management IP address as their source by default. Use the **ipv4 virtual address use-as-src-addr** command to ensure that the protocol uses the virtual IPv4 address as its source address. Alternatively, you can also configure a loopback address with the designated or desired IPv4 address and set that as the source for protocols such as TACACS+ using the **tacacs source-interface** command.

How to Configure Virtual Interfaces

This section contains the following procedures:

Configuring Virtual Loopback Interfaces

This task explains how to configure a basic loopback interface.

Restrictions

- The IP address of a loopback interface must be unique across all routers on the network.
- That IP address must not be used by another interface on the router.
- The IP address must not be used by an interface on any other router on the network.

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

/* Enters interface configuration mode and names the new loopback interface */

```
RP/0/RP0/CPU0:router#(config)# interface Loopback 3
```

/* Assigns an IP address and subnet mask to the virtual loopback interface */

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38/32

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

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

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

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

/* Display the configuration of the loopback interface */

```
RP/0/RP0/CPU0:router# show interfaces Loopback 3
```

Configuring Null Interfaces

This task explains how to configure a basic null interface.

/* Enters global configuration mode. */

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

/* Enter the null 0 interface configuration mode. */

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

/* Save configuration changes. */

```
RP/0/RP0/CPU0:router(config-null0)# end
```

/* Verif the configuration of the null interface. */

```
RP/0/RP0/CPU0:router# show interfaces null 0
```

Configuring Virtual IPv4 Interfaces

This task explains how to configure an IPv4 virtual interface.

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

/* Define an IPv4 virtual address for the management Ethernet interface. */

```
RP/0/RSP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8
RP/0/RSP0/CPU0:router(config-null0)# end
or
RP/0/RSP0/CPU0:router(config-null0)# commit
```

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

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

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

This is an example for configuring a virtual IPv4 interface:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8
RP/0/RSP0/CPU0:router(config-null0)# commit
```

Configuration Examples for Virtual Interfaces

This section provides the following configuration examples:

Configuring a Loopback Interface: Example

The following example indicates how to configure a loopback interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Loopback 3
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38/32
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Loopback 3

Loopback3 is up, line protocol is up
 Hardware is Loopback interface(s)
 Internet address is 172.18.189.38/32
 MTU 1514 bytes, BW Unknown
   reliability 0/255, txload Unknown, rxload Unknown
 Encapsulation Loopback, loopback not set
 Last clearing of "show interface" counters never
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 total input drops
```

```
   0 drops for unrecognized upper-level protocol
   Received 0 broadcast packets, 0 multicast packets
   0 packets output, 0 bytes, 0 total output drops
   Output 0 broadcast packets, 0 multicast packets
```

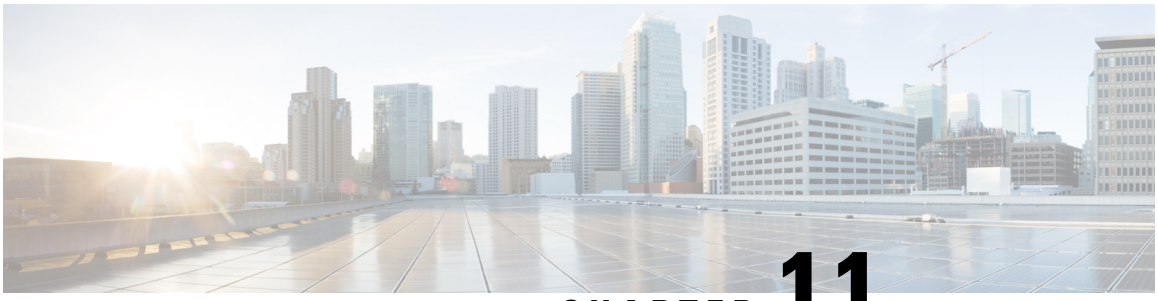
Configuring a Null Interface: Example

The following example indicates how to configure a null interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Null 0
RP/0/RP0/CPU0:router(config-null0)# ipv4 unreachables
RP/0/RP0/CPU0:router(config-null0)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Null 0

Null0 is up, line protocol is up
 Hardware is Null interface
 Internet address is Unknown
 MTU 1500 bytes, BW Unknown
   reliability 0/255, txload Unknown, rxload Unknown
 Encapsulation Null, loopback not set
 Last clearing of "show interface" counters never
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 total input drops
   0 drops for unrecognized upper-level protocol
   Received 0 broadcast packets, 0 multicast packets
   0 packets output, 0 bytes, 0 total output drops
   Output 0 broadcast packets, 0 multicast packets
```

Configuring a Virtual IPv4 Interface: Example

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



CHAPTER 11

Configure GRE Tunnels

Tunneling provides a mechanism to transport packets of one protocol within another protocol. This chapter describes GRE tunneling protocol.

Release	Feature(s) Added
Release 7.3.1	GRE Tunnel feature was introduced.

GRE tunnels

Table 34: Feature History Table

Feature Name	Release Information	Description
Disabling time-to-live (TTL) decrement at GRE encapsulation	Release 7.3.2	This feature allows you to disable the time-to-live (TTL) decrement of the incoming packets. The result is that encapsulation of the original incoming packet takes place without any change in the TTL value. This feature avoids dropping incoming packets with a TTL value equal to one after GRE encapsulation. Before this release, the TTL value of incoming packets was decremented by one before GRE decapsulation. This feature introduces the **tunnel ttl disable** command.

Feature Name	Release Information	Description
GRE tunnel	Release 7.3.1	Generic Routing Encapsulation (GRE) provides a simple approach to transporting packets of one protocol over another protocol using encapsulation. This capability is now extended to the Cisco 8000 Series Routers. This feature supports: • Unidirectional GRE encapsulation • Unidirectional GRE decapsulation And introduces the following commands: • **show interface tunnel-ip <> accounting (encap)** • **show interface tunnel-ip <> accounting (decap)**
Outer-header hashing support for MPLSoGRE and IPoGRE traffic	Release 7.3.1	This feature allows load-balancing of GRE traffic in transit routers. A transit node distributes incoming GRE traffic evenly across all available ECMP links in a GRE tunnel topology. A hashing function uses GRE outer and inner header tuples such as source IP, destination IP, protocol, and router ID to determine traffic entropy. This capability is now extended to the Cisco 8000 Series Routers.

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 should be delivered to a destination network inside an outer IP packet. The GRE tunnel behaves as virtual point-to-point link that has 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 toward the GRE tunnel endpoint. Upon reaching the tunnel endpoint, GRE encapsulation is removed and the payload is forwarded to the packet's ultimate destination.

A tunnel configured using encapsulation mode performs encapsulation of IPv4/IPv6 payload inside the GRE header. A tunnel configured using decapsulation mode performs the opposite. Here, outer GRE header is decapsulated and the inner IPv4/IPv6/MPLS payload is forwarded to the next hop router. Both encapsulation and decapsulation tunnel interfaces collect statistics periodically. The statistics can be displayed on demand using the CLI commands `show interface tunnel-ip1 accounting` and `show policy-map type pbr address-family ipv4 statistics`. For more information, see Unidirectional GRE Encapsulation (GREv4), on page 210 and Unidirectional GRE Decapsulation (GREv4), on page 210.

To perform load-balancing of GRE traffic in transit routers, a transit node distributes incoming GRE traffic evenly across all available ECMP links in a GRE tunnel topology. Furthermore, to determine traffic entropy, a hashing function uses GRE outer and inner header tuples such as source IP, destination IP, protocol, and router ID.

GRE encapsulation and decapsulation over BVI

Table 35: Feature History Table

Feature Name	Release Information	Description
GRE encapsulation and decapsulation over BVI	Release 7.5.4	You can now transport packets using the GRE protocol over Bridge-Group Virtual Interfaces (BVI). This feature uses GRE to encapsulate packets between two endpoints and transmit the encapsulated packets over a BVI interface. At the destination, the GRE packet is decapsulated. GRE encapsulation and decapsulation over BVI allows transmitting packets securely using network layer protocols while maintaining Layer 2 connectivity between the physical interfaces.

From Cisco IOS XR Release 7.5.4, GRE packets are supported over a BVI interface. This support provides GRE encapsulation and decapsulation over the BVI interfaces.

The BVI is a virtual interface within the router that acts like a normal routed interface. The BVI does not support bridging itself, but acts as a gateway for the corresponding bridge-domain to a routed interface within the router. A BVI is associated with a single bridge domain and represents the link between the bridging and the routing domains on the router.

When using GRE over BVI, the GRE header is added to the original IP packet before it is sent to the BVI. The BVI then bridges the encapsulated packet to the destination interface, which is a BVI, physical interface, or a remote network.

When the encapsulated packet reaches its destination, the receiving interface performs GRE decapsulation, which involves removing the GRE header from the original IP packet. The resulting IP packet is then forwarded to its final destination.

For information on BVI, see the *Integrated Routing and Bridging* section in the *L2VPN Configuration Guide for Cisco 8000 Series Routers*.

Supported Features on a GRE Tunnel

GRE tunnel supports the following features:

- GRE or IP-in-IP tunnels support 16 unique source addresses. These 16 unique source addresses are repeated multiple times to configure 1000 encapsulation tunnels or 64 decapsulation tunnels.
- GRE encapsulation supports the following features:
  - IPv4/IPv6 over GRE IPv4 transport
  - MPLS PoP over GRE IPv4 transport
  - ABF (Access List Based Forwarding) v4/v6 over GRE
  - VRF (Virtual Routing and Forwarding) support over GRE
- GRE decapsulation supports the following features:
  - PBR-based GRE decapsulation configuration

- CLI-based GRE decapsulation configuration
- IPv4/IPv6 over GRE decapsulation
- MPLS/SRTE over GRE decapsulation
- A GRE tunnel in decapsulation mode has only tunnel source configured, without any tunnel destination address. This decapsulated GRE tunnel behaves like a P2MP (Point-to-multipoint) tunnel, which means that an incoming GRE packet can have any source IP address and matching destination IP address to the tunnel source configured. However, once a source IP address is used for decapsulated P2MP tunnel, it cannot be re-used with other decapsulation tunnels.

- The command `tunnel ttl disable` is supported. This command controls TTL decrement of a packet being encapsulated. After configuring this command fo a tunnel interface, TTL value of incoming packet is not decremented by one, and original incoming packet is encapsulated without changing the TTL. By default, `tunnel ttl disable` isn't configured. This means that the TTL of incoming packets is decremented by one before GRE encapsulation.

  For example, consider an incoming packet that had the TTL value equal to one. On GRE encapsulation, the TTL value is decremented by one and becomes zero. Therefore the router will discard the packet and send an ICMP message back to the originating host. Using this feature, you can disable TTL decrement and avoid the packet discard.

  Configuration Example

```
Router#configure
Router(config)#interface tunnel-ip30016
Router(config-if)#tunnel ttl disable
Router(config-if)#commit
```

Limitations for Configuring GRE Tunnels

This list describes the limitations for configuring GRE tunnels:

- GRE tunnels configured without any decapsulation or encapsulation mode support only ERPSAN feature.
- Don't create multiple GRE/IP-in-IP tunnels with the same pair of source and destination IP address or interface name. Configure all tunnels with unique source-destination pairs. In an encapsulation or decapsulation tunnel where only either source or destination is mentioned, the source-destination pair should also be unique when compared to other encapsulation or decapsulation tunnels.
- Bi-directional GRE tunnel isn't supported.
- Routing protocols over GRE tunnels aren't supported.
- Multicast over GRE isn't supported.
- GRE KA (Keep Alive) isn't supported.
- GRE parameters such as MTU (Maximum Transmission Unit) and key functionalities aren't supported.

Configure GRE Tunnels

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 source and destination. The router supports only uni-directional GRE with either encapsulation or decapsulation mode.

```
Router# configure
Router(config)# interface tunnel-ip1
Router(config-if)# ipv4 address 101.0.1.2 255.255.255.0
Router(config-if)# ipv6 address 101:0:1::2/64
Router(config-if)# tunnel mode gre ipv4 [encap | decap]
Router(config-if)# tunnel source 2.2.1.1
Router(config-if)# tunnel destination 2.2.2.1/32
Router(config-if)# commit
Router(config-if)# exit
```

To configure ABFv4/v6 over GRE:

```
router static
  address-family ipv4 unicast
  201.0.1.0/24 tunnel-ip1
  address-family ipv6 unicast
  201:0:1::0/64 tunnel-ip1

ipv4 access-list abf-gre
  1 permit ipv4 any any nexthop1 ipv4 201.0.1.2
ipv6 access-list abf6-gre
  1 permit ipv6 any any nexthop1 ipv6 201:0:1::2

interface HundredGigE0/0/0/24
  ipv4 address 24.0.1.1/24
  ipv6 address 24:0:1::1/64
  ipv4 access-group abf-gre ingress
  ipv6 access-group abf6-gre ingress
!
```

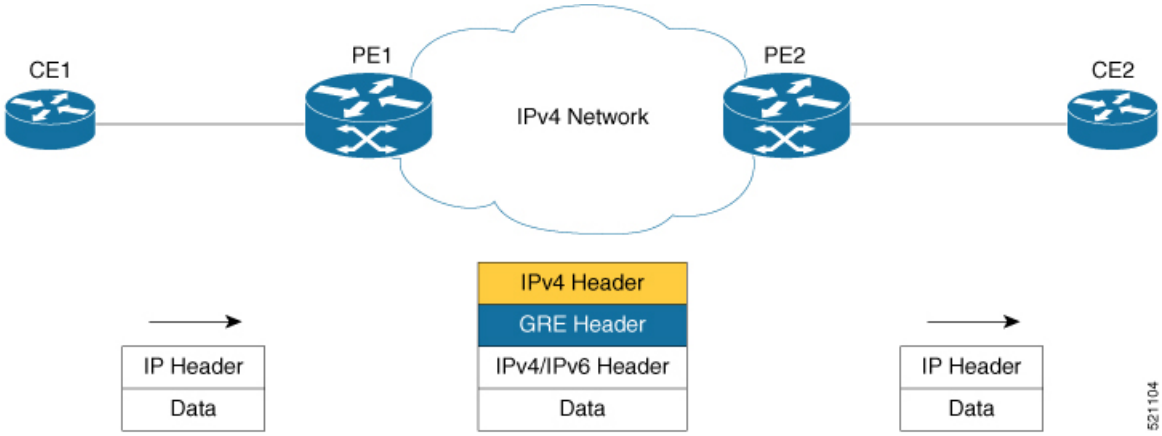
To configure MPLS PoP label over GRE:

```
router static
  address-family ipv4 unicast
  201.0.1.0/24 tunnel-ip1
  address-family ipv6 unicast
  201:0:1::0/64 tunnel-ip1

mpls static
  interface HundredGigE0/0/0/24
  lsp gre
    in-label 30501 allocate
    forward path 1 resolve-nexthop 201.0.1.2 out-label pop
!
```

Note Bi-directional GRE tunnel supports only ERSPAN.

Unidirectional GRE Encapsulation (GREv4)

A tunnel configured using encapsulation mode performs encapsulation of IPv4/IPv6 payload inside the GRE header. The following figure shows GRE encapsulation. Routers in the IP cloud have no knowledge of encapsulated IP source address or destination address.



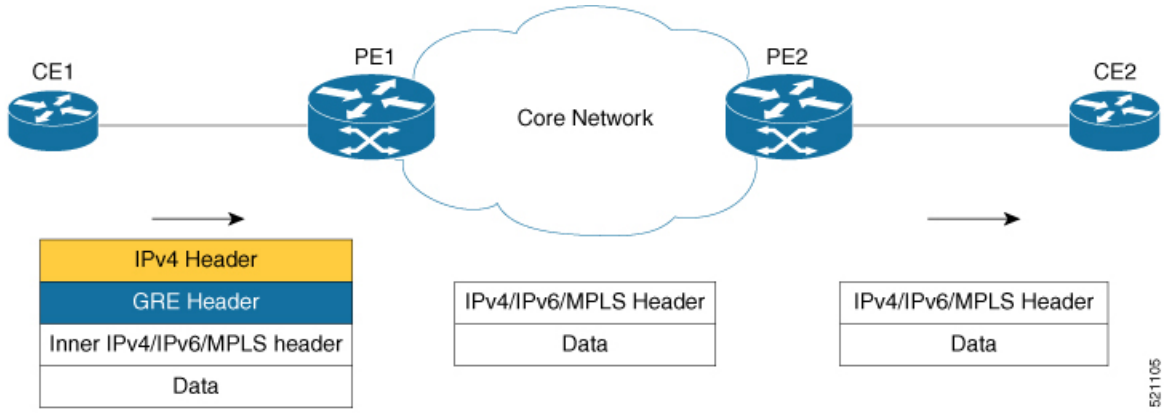
Configuration

The following example shows how to configure GRE tunnel encapsulation:

```
interface tunnel-ip1
  ipv4 address 101.0.1.1/24
  ipv6 address 101:0:1::1/64
  tunnel mode gre ipv4 encap
  tunnel source [ loopback1 | <any-ipaddres> | any-interface]
  tunnel destination [ 20.0.1.1/32 | 20.0.1.0/24 | 20.0.1.0/28]

router static
  address-family ipv4 unicast
  201.0.1.0/24 tunnel-1

router static
  address-family ipv6 unicast
  201:0:1::0/64 tunnel-1
```

Unidirectional GRE Decapsulation (GREv4)

In unidirectional GRE decapsulation, the outer GRE header is decapsulated and the inner IPv4/IPv6/MPLS payload is forwarded to the next hop router. The following figure shows GRE decapsulation. In the figure, PE1 strips off outer GRE header and inner payload is forwarded as regular IPv4/IPv6/MPLS forwarding.



Configuration

There are two methods to configure GRE tunnel decapsulation:

1. CLI-based tunnel decapsulation configuration

```
interface tunnel-ip1
  ipv4 address 101.0.1.1/24
  ipv6 address 101:0:1::1/64
  tunnel mode gre ipv4 decap
  tunnel source [ loopback1 | <any-ipaddres> | any-interface]
  tunnel destination [ 20.0.1.1/32 | 20.0.1.0/24 | 20.0.1.0/28]
```

2. PBR-based tunnel decapsulation configuration

```
class-map type traffic match-all test_gre1
                match protocol gre
                match destination-address ipv4 10.0.1.2 255.255.255.255
                match source-address ipv4 10.10.10.1 255.255.255.255
                end-class-map
policy-map type pbr P1-test
                class type traffic test_gre1  decapsulate gre
vrf-policy vrf default address-family ipv4 policy type pbr input P1-test
```

ECMP and LAG Hashing for NVGRE Flows

Table 36: Feature History Table

Feature Name	Release Information	Feature Description
ECMP and LAG Hashing for NVGRE Flows	Release 7.5.2	This feature allows transit routers to load balance the GRE traffic, based on GRE payload. A transit node distributes incoming GRE traffic across ECMP and LAG paths in a GRE tunnel topology. A hashing function uses GRE payload that consists of inner Ethernet frame with destination MAC and source MAC addresses, to derive the traffic entropy. ECMP and LAG hashing is enabled on Cisco 8000 series routers by default.

Network Virtualization using Generic Routing Encapsulation (NVGRE) endpoints are network devices that act as interfaces between physical and virtual networks. NVGRE endpoint encapsulates Ethernet data frames to and from GRE tunnel. The encapsulated GRE packet is bridged and routed to the destination. On the destination, the NVGRE endpoint decapsulates the GRE packet to recover the original Ethernet frame. NVGRE is described in RFC 7637.

NVGRE uses the following header information for encapsulation:

Header	Parameters
Outer Ethernet Header	Destination MAC address, Source MAC add
Outer IP Header	IPv4 and IPv6 addresses as delivery protocol
GRE Header	GRE protocol type 0x6558 (transparent Ethe
GRE Payload	Inner Ethernet frame with Destination MAC address

For load balancing the GRE traffic, the transit router uses GRE payload that consists of inner Ethernet frame with destination MAC and source MAC addresses. The transit router derives the traffic entropy information from the GRE payload.

The hashing function considers the following parameters of GRE packets, along with Router ID, for load balancing the GRE traffic:

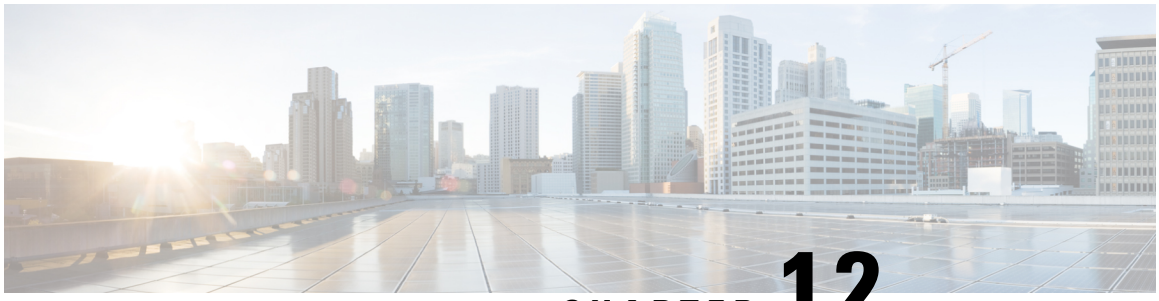
Header	Parameters
Outer IPv4 Header	Source IP address, Destination IP address, IP
Outer IPv6 Header	Source IP address, Destination IP address, Fl (GRE)

Inner Header	Destination MAC address, Source MAC

Restrictions for ECMP and LAG Hashing for NVGRE Flows

ECMP and LAG hashing does not support:

- Outer IPv4 header with Options field.
- Outer IPv6 header with extension headers.



CHAPTER 12

Configuring 802.1Q VLAN Interfaces

This module describes the configuration and management of 802.1Q VLAN interfaces.

The IEEE 802.1Q specification establishes a standard method for tagging Ethernet frames with VLAN membership information. It 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 7.0.11	This feature was introduced.
Release 7.2.12	Support for Layer 2 interfaces was introduced.

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, ensure that you meet the following conditions:

- You must have configured a HundredGigE interface, a FourHundredGigE interface, or an Ethernet bundle interface.

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 you can configure so that the devices can communicate as if they were attached to the same wire. When in fact, they are located on several different LAN segments. Because VLANs are based on logical instead of physical connections, they are 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 40Gigabit, HundredGig, FourHundredGig, and bundle 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 that you can create on a hardware interface. These software-defined interfaces allow the segregation of traffic into separate logical channels on a single hardware interface. It also allows for the better utilization of the available bandwidth on the physical interface.

You can distinguish subinterfaces from each other by adding an extension at the end of the interface name and designation. For instance, the system indicates Ethernet subinterface 23 on the physical interface designated TenGigE 0/1/0/0, by TenGigE 0/1/0/0.23.

Before the system allows a subinterface 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, you must explicitly define the VLAN identifier.

Supported Encapsulation

Table 37: 802.1ad Encapsulation Support for Layer 3 Interfaces and subinterfaces

Interface Type	Encapsulation	Standard	Support Status
Layer 3 interface Layer 3 subinterface Layer 3 bundle subinterface	Single-Tag Encapsulation	dot1ad	Supported (From Cisco IOS XR Software Release 24.4.1 onwards)
		dot1q	Supported.
	Double-Tag Encapsulation	dot1ad <> dot1q<>	Supported.
		dot1q <> dot1q<>	[1]Supported.

[1] The **encapsulation dot1q <x> second-dot1q <y>** encapsulation type is supported on Q200-based line cards from Cisco IOS XR Software Release 24.1.1 onwards and supported for all hardware platforms in the Cisco 8000 Series Routers from Cisco IOS XR Software Release 24.4.1 onwards.

For information about supported encapsulation for Layer 2 Interfaces and subinterfaces, see Virtual LANs in Layer 2 VPNs.

Subinterface MTU

The system inherits the subinterface maximum transmission unit (MTU)from the physical interface with an additional four bytes allowed for the 802.1Q VLAN tag.

Native VLAN

The router does not support a native VLAN. However, the equivalent functionality is accomplished using an **encapsulation** command as follows:

```
encapsulation dot1q TAG-ID
```

Layer 2 VPN on VLANs

The Layer 2 Virtual Private Network (L2VPN) feature enables Service Providers (SPs) to provide Layer 2 services to geographically disparate customer sites.

The configuration model for configuring VLAN attachment circuits (ACs) is similar to the model used for configuring basic VLANs, where the user first creates a VLAN subinterface, and then configures that VLAN in subinterface configuration mode. To create an AC, you need to include the **l2transport** keyword in the **interface** command string to specify that the interface is a Layer 2 interface.

VLAN ACs support three modes of L2VPN operation:

- Basic Dot1Q AC—The AC covers all frames that are received and sent with a specific VLAN tag.
- QinQ AC— Only outer tag (s-tag) of 0x88a8 and inner tag (c-tag) of 0x8100 is supported.

Keep the following in mind when configuring L2VPN on a VLAN:

- Cisco IOS XR software supports 255 ACs per LC.

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

How to Configure 802.1Q VLAN Interfaces

This section contains the following procedures:

Configuring 802.1Q VLAN Subinterfaces

This task explains how to configure 802.1Q VLAN subinterfaces. To remove these subinterfaces, see the "Removing an 802.1Q VLAN Subinterface" section.

Tip You can programmatically configure and retrieve the VLAN interfaces and subinterfaces parameters using `openconfig-vlan.yang` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

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

/* Enter subinterface configuration mode and specifies the interface type, location, and subinterface number. */

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10
```

- 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.

/* Set the Layer 2 encapsulation of an interface. */

```
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
```

Note
- The **dot1q vlan** command is replaced by the **encapsulation dot1q** command on the Cisco 8000 Series Router. It is still available for backward-compatibility, but only for Layer 3 interfaces.

/* Assign an IP address and subnet mask to the subinterface. */

```
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24
```

- 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.

/* The **exit** command is not explicitly required. */

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

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

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

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

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

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

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

Example

```
"RP/0/RP0/CPU0:S3(config)#interface fourHundredGigE 0/5/0/1.100
RP/0/RP0/CPU0:S3(config-subif)#ipv4 address 100.100.100.100/31
RP/0/RP0/CPU0:S3(config-subif)#encapsulation dot1q 100
RP/0/RP0/CPU0:S3(config-subif)#no shutdown
RP/0/RP0/CPU0:S3(config-subif)#commit
Mon Jul  8 23:05:01.979 PDT
RP/0/RP0/CPU0:S3(config-subif)#end
RP/0/RP0/CPU0:S3#show interfaces fourHundredGigE 0/5/0/1.100 brief
Mon Jul  8 23:05:08.784 PDT

               Intf       Intf        LineP              Encap  MTU        BW
               Name       State       State               Type (byte)    (Kbps)
--------------------------------------------------------------------------------
      FH0/5/0/1.100        up        up             802.1Q  1518  400000000

RP/0/RP0/CPU0:S3#show interfaces brief location 0/5/CPU0 | include 802.1Q
Mon Jul  8 23:07:43.929 PDT
      FH0/5/0/1.100        up        up             802.1Q  1518  400000000
RP/0/RP0/CPU0:S3#

RP/0/RP0/CPU0:S3#"
```

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 [HundredGigE | TenGigE | Bundle-Ether | TenGigE]** *interface-path*] *id.subinterface* **l2transport**
3. **encapsulation dot1q** *vlan-id*
4. **end** or **commit**
5. **show interfaces** [**HundredGigE** | **TenGigE**] *interface-path-id.subinterface*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 2	**interface [HundredGigE | TenGigE | Bundle-Ether | TenGigE]** *interface-path*] *id.subinterface* **l2transport** **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0.1 l2transport`	Enters subinterface configuration and specifies the interface type, location, and subinterface number. • Replace the 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. • You must include the **l2transport** keyword in the command string; otherwise, the configuration creates a Layer 3 subinterface rather that an AC.
Step 3	**encapsulation dot1q** *vlan-id* **Example:** `RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100`	Sets the Layer 2 encapsulation of an interface.

	Command or Action	Purpose

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if-l2)# end`
	or
	`RP/0/RP0/CPU0:router(config-if-l2)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		- 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** [**HundredGigE** | **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.

Removing an 802.1Q VLAN Subinterface

This task explains how to remove 802.1Q VLAN subinterfaces that have been previously configured using the Configuring 802.1Q VLAN subinterfaces section in this module.

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

/* Remove the subinterface, which also automatically deletes all the configuration applied to the subinterface. */

```
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.

Note Repeat to remove other VLAN subinterfaces.

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

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

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

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

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

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

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

Configuration Examples for VLAN Interfaces

This section contains the following example:

VLAN Subinterfaces: Example

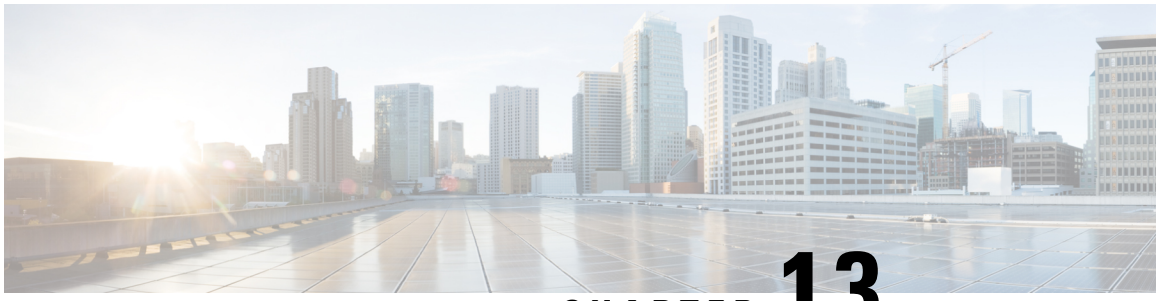
The following example shows how to create three VLAN subinterfaces at one time:

```
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)# encapsulation dot1q 100
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 10.0.10.1/24
RP/0/RP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.2

RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 101
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 10.0.20.1/24
RP/0/RP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.3

RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 102
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 ethernet trunk bundle-Ether 1
Trunk                               Sub types        Sub states
VLAN trunks: 1,
  1 are 802.1Q (Ether)
Sub-interfaces: 3,
  3 are up.
802.1Q VLANs: 3,
  3 have VLAN Ids,

RP/0/RP0/CPU0:router# show vlan interfaceInterface     St Ly    MTU    Subs    L3
   Up    Down  Ad-Down
Te0/2/0/4.1         802.1Q              10  up
Te0/2/0/4.2         802.1Q              20  up
Te0/2/0/4.3         802.1Q              30  up
RP/0/RP0/CPU0:router# show vlan trunks briefBE1           Up L3   1514    1000
  0    1000     1000       0       0

Summary                       1000        0    1000     1000       0       0

Te0/2/0/4           802.1Q (Ether)        up
```

The following example shows how to create two VLAN subinterfaces on an Ethernet bundle:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface bundle-ether 2
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.2.1/24
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# interface bundle-ether 2.1

RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 192.168.100.1/24
RP/0/RP0/CPU0:router(config-subif)# exit
RP/0/RP0/CPU0:router(config)# interface bundle-ether 2.2

RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 200
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
```



CHAPTER 13

Configure IP-in-IP Tunnels

This chapter provides conceptual and configuration information for IP-in-IP tunnels.

Table 38: Feature History for Configure Tunnels

Release 7.0.11	This feature was introduced.
Release 7.0.14	Support for the following feature was introduced in Configure Tunnels: • Extended ACL must match on the outer header for IP-in-IP Decapsulation.

Table 39: Feature History Table

Feature Name	Release Information	Feature Description
IPv4 packets with IPv6 outer header	Release 7.5.3	With this release, decapsulation of IPv4 and IPv6 tunnels with IPv6 outer headers are supported. This feature helps the administrators to take advantage of the benefits of IPv6, such as improved routing and security, without having to upgrade their entire network to IPv6.

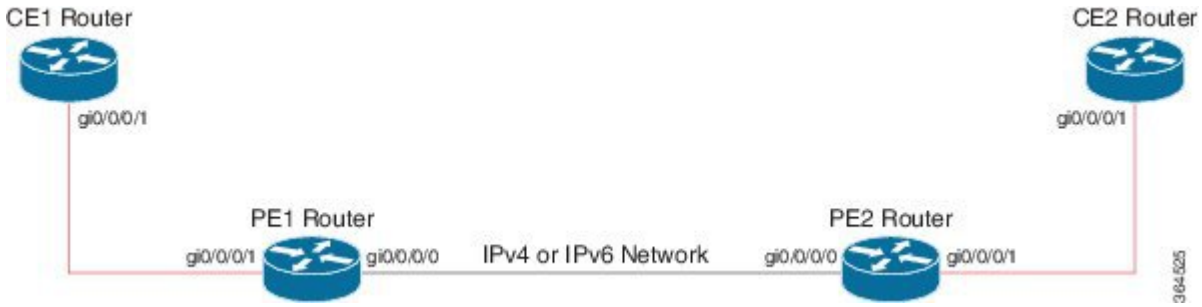
Overview of IP-in-IP Tunnels

Tunneling provides a mechanism to transport packets of one protocol within another protocol. IP-in-IP tunneling refers to the encapsulation and decapsulation of an IP packet as a payload in another IP packet. Cisco 8000 Series Routers support IP-in-IP decapsulation with all possible combinations of IPv4 and IPv6; that is, IPv4 over IPv4, IPv6 over IPv4, IPv4 over IPv6, and IPv6 over IPv6. For example, an IPv4 over IPv6 refers to an IPv4 packet as a payload encapsulated within an IPv6 packet and routed across an IPv6 network to reach the destination IPv4 network, where it is decapsulated.

IP-in-IP tunneling can be used to connect remote networks securely or provide virtual private network (VPN) services.

The following example provides configurations for an IPv4 or IPv6 tunnel, with the transport VRF as the default VRF for the following simplified network topology.

Figure 10: IP-in-IP Tunnel Network Topology



Guidelines and Restrictions for Configure IP-in-IP Tunnels

- The feature does not support decapsulation tunnels on subinterfaces.
- Only the default Virtual Routing and Forwarding (VRF) instance is supported.
- IPv6 link local addresses are not supported.
- Regular tunnels cannot use a configured IP address as the tunnel source; only a non-existent IP address can be used.
- Configuring multiple interfaces with the same IP address is not supported.
- Each line card can have different number of Network Processor (NP) slices.
- The maximum IPv4 and IPv6 IP-in-IP decapsulation tunnels supported is 64 per slice.

Configuration Example for IPv4 Tunnel

PE1 Router Configuration	PE2 Router Configuration

```
interface GigabitEthernet0/0/0/0
 !! Link between PE1-PE2
 ipv4 address 100.1.1.1/24
!
interface GigabitEthernet0/0/0/1
 !! Link between CE1-PE1
ipv4 address 20.1.1.1/24
 ipv6 address 20::1/64
!
interface tunnel-ip 1
ipv4 address 10.1.1.1/24
 ipv6 address 10::1/64
 tunnel mode ipv4
 tunnel source GigabitEthernet0/0/0/0
 tunnel destination 100.1.1.2
 !

router static
address-family ipv4 unicast
   30.1.1.0/24 tunnel-ip1
  address-family ipv6 unicast
   30::0/64 tunnel-ip1
  !
 !
!
```

```
interface GigabitEthernet0/0/0/0
!! Link between PE1-PE2
 ipv4 address 100.1.1.2/24
!
interface GigabitEthernet0/0/0/1
 !! Link between PE2-CE2
ipv4 address 30.1.1.1/24
 ipv6 address 30::1/64
!
interface tunnel-ip 1
ipv4 address 10.1.1.2/24
 ipv6 address 10::2/64
 tunnel mode ipv4
 tunnel source GigabitEthernet0/0/0/0
 tunnel destination 100.1.1.1
 !

router static
address-family ipv4 unicast
   20.1.1.0/24 tunnel-ip1
  address-family ipv6 unicast
   20::0/64 tunnel-ip1
  !
 !
!
```

CE1 Router Configuration	CE2 Router Configuration

```
interface GigabitEthernet0/0/0/1
!! Link between CE1-PE1
 ipv4 address 20.1.1.2 255.255.255.0
 ipv6 address 20::2/64
!
router static
 address-family ipv4 unicast
  30.1.1.0/24 20.1.1.1
 address-family ipv6 unicast
  30::0/64 20::1
 !
!
```

```
interface GigabitEthernet0/0/0/1
!! Link between CE2-PE2
 ipv4 address 30.1.1.2 255.255.255.0
 ipv6 address 30::2/64
!
router static
 address-family ipv4 unicast
  20.1.1.0/24 30.1.1.1
address-family ipv6 unicast
  20::0/64 30::1
 !
!
```

Configuration Example for IPv6 Tunnel

PE1 Router Configuration	PE2 Router Configuration

```
interface GigabitEthernet0/0/0/0
 !! Link between PE1-PE2
 ipv6 address 100::1/64
!
interface GigabitEthernet0/0/0/1
 !! Link between CE1-PE1
 vrf RED
 ipv4 address 20.1.1.1/24
 ipv6 address 20::1/64
!
interface tunnel-ip 1
 vrf RED
 ipv4 address 10.1.1.1/24
 ipv6 address 10::1/64
 tunnel mode ipv6
 tunnel source GigabitEthernet0/0/0/0
 tunnel destination 100::2
 !
vrf RED
 address-family ipv6 unicast
  import route-target
   2:1
  !
  export route-target
   2:1
  !
 address-family ipv4 unicast
  import route-target
   2:1
  !
  export route-target
   2:1
  !
router static
vrf RED
  address-family ipv4 unicast
   30.1.1.0/24 tunnel-ip1
  address-family ipv6 unicast
   30::0/64 tunnel-ip1
  !
 !
!
```

```
interface GigabitEthernet0/0/0/0
!! Link between PE1-PE2
 ipv6 address 100::2/64
!
interface GigabitEthernet0/0/0/1
 !! Link between PE2-CE2
 vrf RED
 ipv4 address 30.1.1.1/24
 ipv6 address 30::1/64
!
interface tunnel-ip 1
 vrf RED
 ipv4 address 10.1.1.2/24
 ipv6 address 10::2/64
 tunnel mode ipv6
 tunnel source GigabitEthernet0/0/0/0
 tunnel destination 100::1
 !
vrf RED
 address-family ipv6 unicast
  import route-target
   2:1
  !
  export route-target
   2:1
  !
 address-family ipv4 unicast
  import route-target
   2:1
  !
  export route-target
   2:1
  !
router static
vrf RED
  address-family ipv4 unicast
   20.1.1.0/24 tunnel-ip1
  address-family ipv6 unicast
   20::0/64 tunnel-ip1
  !
 !
!
```

CE1 Router Configuration

```
interface GigabitEthernet0/0/0/1
!! Link between CE1-PE1
 ipv4 address 20.1.1.2 255.255.255.0
 ipv6 address 20::2/64
!
router static
 address-family ipv4 unicast
  30.1.1.0/24 20.1.1.1
 address-family ipv6 unicast
  30::0/64 20::1
 !
!
```

CE2 Router Configuration

```
interface GigabitEthernet0/0/0/1
!! Link between CE2-PE2
 ipv4 address 30.1.1.2 255.255.255.0
 ipv6 address 30::2/64
!
router static
 address-family ipv4 unicast
  20.1.1.0/24 30.1.1.1
address-family ipv6 unicast
  20::0/64 30::1
 !
!
```


- IP-in-IP Decapsulation, on page 229
- ECMP Hashing Support for Load Balancing, on page 237

IP-in-IP Decapsulation

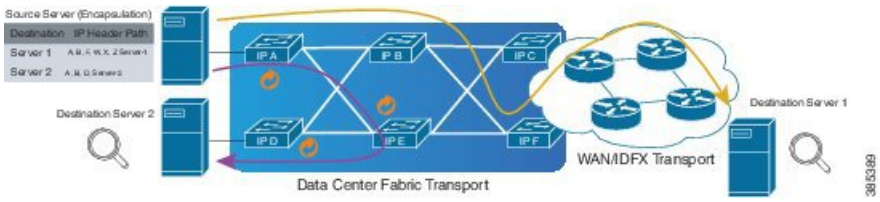
IP-in-IP encapsulation involves the insertion of an outer IP header over the existing IP header. The source and destination address in the outer IP header point to the endpoints of the IP-in-IP tunnel. The stack of IP headers is used to direct the packet over a predetermined path to the destination, provided the network administrator 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.

In IP-in-IP encapsulation and decapsulation has two types of packets. The original IP packets that are encapsulated are called Inner packets and the IP header stack added while encasulation are called the Outer packets.

Note The router only supports decapsulation and no encapsulation. Encapsulation is done by remote routers.

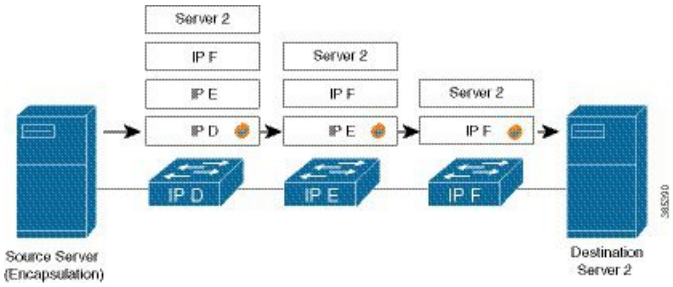
The following topology describes a use case where IP-in-IP encapsulation and decapsulation are used for different segments of the network from source to destination. The IP-in-IP tunnel consists of multiple routers that are used to decapsulate and direct the packet through the data center fabric network.

Figure 11: IP-in-IP Decapsulation Through a Data Center Network



The following illustration shows how the stacked IPv4 headers are decapsulated as they traverse through the decapsulating routers.

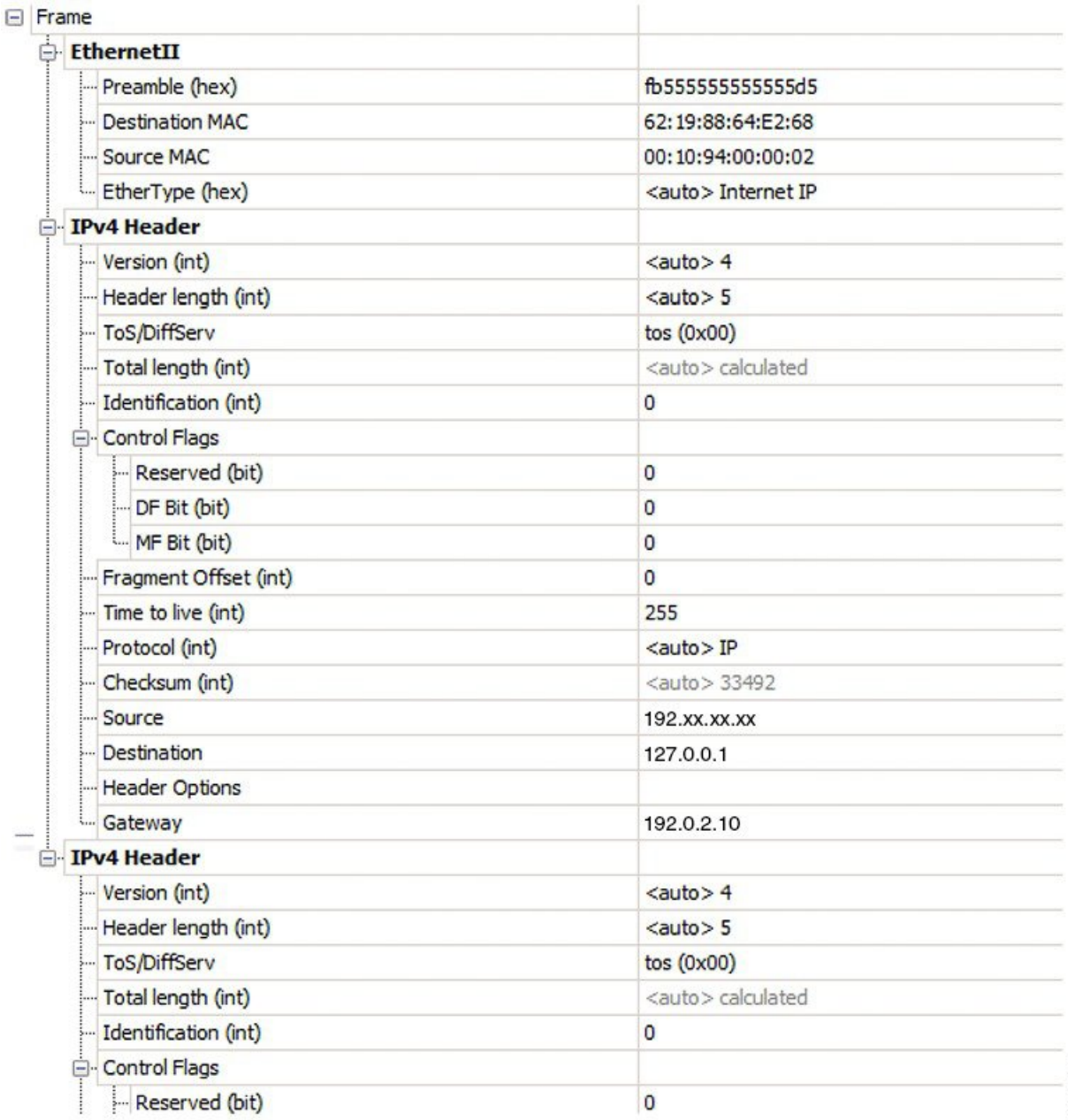
Figure 12: IP Header Decapsulation



Stacked IP Header in an Encapsulated Packet

The encapsulated packet has an outer IPv4 header that is stacked over the original IPv4 header, as shown in the following illustration.

Figure 13: Encapsulated Packet



Configuration

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

```
Router(config)# interface loopback 0
Router(config-if)# ipv4 address 127.0.0.1/32
Router(config-if)# no shutdown
Router(config-if)# interface tunnel-ip 10
```

```
Router(config-if)# ipv4 unnumbered loopback 1
Router(config-if)# tunnel mode ipv4 decap
Router(config-if)# tunnel source loopback 0
```

- **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 decapsulation.
- **tunnel source**: indicates the source address for the IP-in-IP decap tunnel with respect to the router interface.

Note You can configure the tunnel destination only if you want to decapsulate packets from a particular destination. If no tunnel destination is configured, then all the ip-in-ip ingress packets on the configured interface are decapsulated.

Running Configuration

```
Router# show running-config interface tunnel-ip 10
...
interface tunnel-ip 10
ipv4 unnumbered loopback 1
tunnel mode ipv4 decap
```

Extended ACL to Match the Outer Header for IP-in-IP Decapsulation

Starting with Cisco IOS XR Software Release 7.0.14, extended ACL has to match on the outer header for IP-in-IP Decapsulation. Extended ACL support reduces mirrored traffic throughput. This match is based only on the IPv4 protocol, and extended ACL is applied to the received outermost IP header, even if the outer header is locally terminated.

Sample configuration:

```
Router#show running-config interface bundle-Ether 50.5
Tue May 26 12:11:49.017 UTC
interface Bundle-Ether50.5
ipv4 address 101.1.5.1 255.255.255.0
encapsulation dot1q 5
ipv4 access-group ExtACL_IPinIP ingress
ipv4 access-group any_dscpegg egress
!


Router#show access-lists ipv4 ExtACL_IPinIP hardware ingress location$
Tue May 26 12:11:55.940 UTC
ipv4 access-list ExtACL_IPinIP
10 permit ipv4 192.168.0.0 0.0.255.255 any ttl gt 150
11 deny ipv4 172.16.0.0 0.0.255.255 any fragments
12 permit ipv4 any any
```

Decapsulation using tunnel source direct

Table 40: Feature History Table

Feature Name	Release Information	Feature Description
Decapsulation using tunnel source direct	Release 7.5.3	Tunnel source direct allows you to decapsulate the tunnels on any L3 interface on the router. You can use the **tunnel source direct** configuration command to choose the specific IP Equal-Cost Multipath (ECMP) links for troubleshooting, when there are multiple IP links between two devices.

To debug faults in various large networks, you may have to capture and analyze the network traffic at a packet level. In datacenter networks, administrators face problems with the volume of traffic and diversity of faults. To troubleshoot faults in a timely manner, DCN administrators must identify affected packets inside large volumes of traffic. They must track them across multiple network components, analyze traffic traces for fault patterns, and test or confirm potential causes.

In some networks, IP-in-IP decapsulation is currently used in network management, to verify ECMP availability and to measure the latency of each path within a datacenter.

The Network Management System (NMS) sends IP-in-IP (IPv4 or IPv6) packets with a stack (multiple) of predefined IPv4 or IPv6 headers (device IP addresses). The destination device at each hop removes the outside header, performs a lookup on the next header, and forwards the packets if a route exists.

Using the **tunnel source direct** command, you can choose the specific IP Equal-Cost Multipath (ECMP) links for troubleshooting, when there are multiple IP links between two devices.

Tip You can programmatically configure and manage the Ethernet interfaces using `openconfig-ethernet-if.yang` and `openconfig-interfaces.yang` OpenConfig data models. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

Guidelines and Limitations

The following guidelines are applicable to this feature.

- The **tunnel source direct** command is only compatible with 'tunnel mode decap' for IP-in-IP decapsulation.
- The source-direct tunnel is always operationally `up` unless it is administratively shut down. The directly connected interfaces are identified using the **show ip route direct** command.
- All Layer 3 interfaces that are configured on the device are supported.
- Platform can accept and program only certain number of IP addresses. The number of IP addresses depends on the make of the platform linecard (LC). Each LC can have different number of Network Processor (NP) slices and interfaces.

- Only one source-direct tunnel per address-family is supported for configuration.
- Regular decapsulation tunnels which have specific source address, are supported. However, the tunnel's specific source address must not be part of any interface.

The following functionalities are not supported for the **tunnel source direct** option.

- GRE tunneling mode.
- VRF (only default VRF is supported).
- ACL and QoS on the tunnels.
- Tunnel encapsulation.
- Tunnel NetIO DLL: Decapsulation is not supported if the packet is punted to slow path.

Configure Decapsulation Using Tunnel Source Direct

Configuration

The **tunnel source direct** configures IP-in-IP tunnel decapsulation on any directly connected IP addresses. This option is now supported only when the IP-in-IP decapsulation is used to source route the packets through the network.

This example shows how to configure IP-in-IP tunnel decapsulation on directly connected IP addresses:

```
Router# configure terminal
Router(config)#interface Tunnel4
  Router(config)#tunnel mode ipv4 decap
  Router(config)#tunnel source direct
  Router(config)#no shutdown
```

This example shows how to configure IP-in-IP tunnel decapsulation on IPv6 enabled networks:

```
Router# configure terminal
Router(config)#interface Tunnel6
  Router(config)#tunnel mode ipv6 decap
  Router(config)#tunnel source direct
  Router(config)#no shutdown
```

Verifying the Configuration

The following example shows how to verify IP-in-IP tunnel decapsulation with **tunnel source direct** option:

```
Router#show running-config interface tunnel 1
interface Tunnel1
  tunnel mode ipv6ipv6 decapsulate-any
  tunnel source direct
  no shutdown

Router#show interface tunnel 1
Tunnel1 is up    Admin State: up
MTU 1460 bytes, BW 9 Kbit
Tunnel protocol/transport IPv6/DECAPANY/IPv6
Tunnel source - direct
Tx    0 packets output, 0 bytes    Rx    0 packets input, 0 bytes
```

Configure Tunnel Destination with an Object Group

Table 41: Feature History Table

Feature Name	Release Information	Description
Configure Tunnel Destination with an Object Group	Release 7.5.4	You can now assign an object group as the destination for an IP-in-IP decapsulation tunnel. With this functionality, you could configure an IPv4 or IPv6 object group consisting of multiple IPv4 or IPv6 addresses as the destination for the tunnel instead of a single IPv4 or IPv6 address. Using an object group instead of a singular IP address. This helps reduce the configuration complexity in the router by replacing the multiple tunnels with one destination with a single decapsulation tunnel that supports a diverse range of destinations The feature introduces these changes: • **CLI**: New **tunnel destination** command. • **YANG Data Model**: New **object-group** option supported in Cisco-IOS-XR-um-if-tunnel-cfg.yang Cisco native model (see GitHub).

In IP-in-IP Decapsulation, the router accepts a packet on a tunneled interface only when the tunnel IP address matches the source IP address of the incoming packets. With this implementation, the user needs to configure separate interface tunnels for each IP address that the router needs to receive the traffic packets. This limitation often leads to configuration overload on the router.

You can eliminate the configuration overload on the router by assigning an object group as the tunnel destination for IPv4 and IPv6 traffic types. That is, the router matches the source IP address of the incoming packet against the object group available as the tunnel destination. The decapsulation tunnel accepts the incoming traffic packets when there's a match between the packet source and the object group. Otherwise, the router drops the packets.

Restrictions

The following restrictions are applicable to the tunnel destination with an object group feature:

- GRE tunnels don't support configuring object groups as the tunnel destination.

- The router supports configuring tunnel destination with an object group only when the tunnel source is tunnel source direct.
- You can configure the object group as tunnel destination only on default VRF.
- Configuring object groups as the tunnel destination isn't applicable to tunnel encapsulation.
- Subinterfaces don't support configuring object groups as the tunnel destination.
- Configuring object groups as the tunnel destination feature is mutually exclusive with ACL and QoS features.
- The tunnel destination feature supports only IPv4 and IPv6 object groups.
- The router does not support changing tunnel configuration after its creation. Configure the tunnel source direct and tunnel destination with an object group while creating the tunnel only.

Prerequisites

- Define an object group including the network elements for the tunnel destination.
- Enable the tunnel source direct feature. For more information, see Decapsulation using tunnel source direct, on page 232.

Configuration example

This section provides an example for configuring the tunnel destination with an object group.

IPv4 configuration

```
Router# configure
/* Configure the IPv4 object group */
Router(config)# object-group network ipv4 Test_IPv4
Router(config-object-group-ipv4)# 192.0.2.0/24
Router(config-object-group-ipv4)# 198.51.100.0/24
Router(config-object-group-ipv4)# 203.0.113.0/24
Router(config-object-group-ipv4)# commit
Router(config-object-group-ipv4)# exit

/* Enters the tunnel configuration mode */
Router(config)# interface tunnel-ip 1

/* Configures the tunnel mode */
Router(config-if)# tunnel mode ipv4 decap

/* Configures the tunnel to accept all packets with destination address matching the IP
addresses on the router */
Router(config-if)# tunnel source direct

/* Configures the destination of the tunnel as the defined object-group */
Router(config-if)# tunnel destination object-group ipv4 Test_IPv4

Router(config-if)# no shutdown
Router(config-if)# commit
Router(config-if)# exit
```

IPv6 configuration

```
Router# configure
/* Configure the IPv6 object group */
Router(config)# object-group network ipv6 Test_IPv6
```

```
Router(config-object-group-ipv6)# 2001:DB8::/32
Router(config-object-group-ipv6)# 2001:DB8::/48
Router(config-object-group-ipv6)# commit
Router(config-object-group-ipv6)# exit

/* Enters the tunnel configuration mode */
Router(config)# interface tunnel-ip 2

/* Configures the tunnel mode */
Router(config-if)# tunnel mode ipv6 decap

/* Configures the tunnel to accept all packets with destination address matching the IP
addresses on the router */
Router(config-if)# tunnel source direct

/* Configures the destination of the tunnel as the defined object-group */
Router(config-if)# tunnel destination object-group ipv6 Test_IPv6

Router(config-if)# no shutdown
Router(config-if)# commit
Router(config-if)# exit
```

Running Configuration

```
Router# show running-config object-group
object-group network ipv4 Test_IPv4
 192.0.2.0/24
 198.51.100.0/24
 203.0.113.0/24
!
object-group network ipv6 Test_IPv6
 2001:DB8::/32
 2001:DB8::/48
!

Router#show running-config interface tunnel-ip 1
interface tunnel-ip1
 tunnel mode ipv4 decap
 tunnel source direct
 tunnel destination object-group ipv4 Test_IPv4
!

Router#show running-config interface tunnel-ip 2
Fri Nov 29 11:26:54.716 UTC
interface tunnel-ip2
 tunnel mode ipv6 decap
 tunnel source direct
 tunnel destination object-group ipv6 Test_IPv6
!
```

Verification

```
Router# show tunnel ip ea database

----- node0_0_CPU0 -----
tunnel ifhandle 0x80022cc
tunnel source 161.115.1.2
tunnel destination address group Test_IPv4
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
```

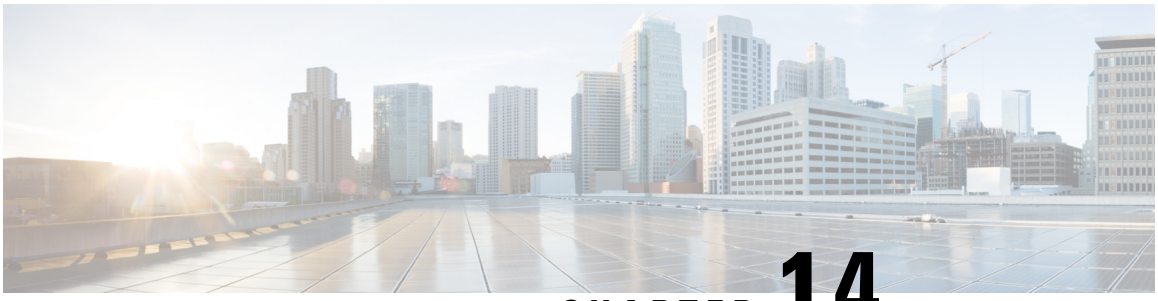
```
Encap
tunnel mtu 1500
tunnel tos 0
tunnel ttl 255
tunnel adjacency flags 0x1
tunnel o/p interface handle 0x0
tunnel key 0x0, entropy length 0 (mask 0xffffffff)
tunnel QT next 0x0
tunnel platform data (nil)
Platform:
Handle: (nil)
Decap ID: 0
Decap RIF: 0
Decap Recycle Encap ID: 0x00000000
Encap RIF: 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
```

ECMP Hashing Support for Load Balancing

The system inherently supports the n-tuple hash algorithm. The first inner header in the n-tuple hashing includes the source port and the destination port of UDP / TCP protocol headers.

The load balancing performs these functions:

- Incoming data traffic is distributed over multiple equal-cost connections.
- Incoming data traffic is distributed over multiple equal-cost connections member links within a bundle interface.
- Layer 2 bundle and Layer 3 (network layer) load-balancing decisions are taken on IPv4, and IPv6. If it is an IPv4 or an IPv6 payload, then an n-tuple hashing is done.
- An n-tuple hash algorithm provides more granular load balancing and used for load balancing over multiple equal-cost Layer 3 (network layer) paths. The Layer 3 (network layer) path is on a physical interface or on a bundle interface.
- The n-tuple load-balance hash calculation contains:
    - Source IP address
    - Destination IP address
    - IP Protocol type
    - Router ID
    - Source port
    - Destination port
    - Input interface
    - Flow-label (for IPv6 only)



CHAPTER 14

Configuring Generic UDP Encapsulation

Read this section to get an overview and know how to configure the Generic UDP Encapsulation.

Table 42: Feature History Table

Feature Name	Release Information	Feature Description
Outer IP Header-Driven Hash Computation for Incoming GUE Packets	Release 7.11.1	We now offer you the flexibility of using only the outer IP header to calculate the hashing for incoming Generic UDP Encapsulation (GUE) packets. On enabling this feature, only the outer IP source and destination addresses are used for hashing calculations. The inner IP addresses are not considered, providing a simpler method of distribution. Previously, both inner IP and outer IP headers were used for ECMP hashing the incoming GUE packets. The feature introduces these changes: **CLI:** • **hw-module profile gue underlay-hash** **YANG Data Models:** • New XPaths for `Cisco-IOS-XR-npu-hw-profile-cfg.yang` (see GitHub, YANG Data Models Navigator) The command is supported on Q200-based ASICs.

Feature Name	Release Information	Feature Description
Generic UDP Encapsulation	Release 7.3.1	This feature enables you to add an additional header to packets to identify or authenticate the data using UDP. Encapsulating packets in UDP leverages the use of the UDP source port to provide entropy to Equal Cost Multi-Path (ECMP) hashing. It provides significant performance benefits for load-balancing. This command is introduced for this feature: **decapsulate gue**

• Understand Generic UDP Encapsulation, on page 240
• Flexible Assignment of UDP Port Numbers for Decapsulation, on page 247

Understand Generic UDP Encapsulation

UDP encapsulation is a technique of adding network headers to packets and then encapsulating the packets within the User Datagram Protocol (UDP).

Encapsulating packets using UDP facilitates efficient transport across networks. By leveraging Receive Side Scaling (RSS) and Equal Cost Multipath (ECMP) routing, UDP provides significant performance benefits for load-balancing. The use of the UDP source port provides entropy to ECMP hashing and provides the ability to use the IP source or destination, and the L4 Port for load-balancing entropy.

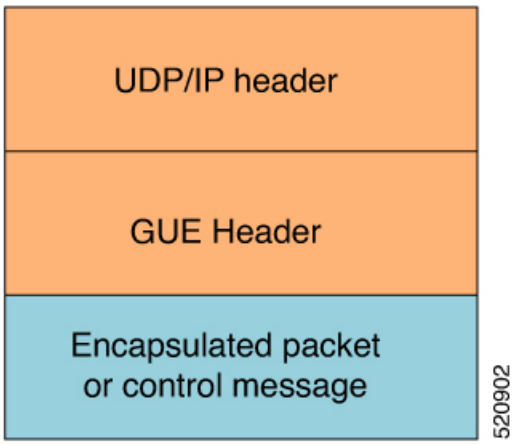
Traditional mechanisms like Generic Routing Encapsulation (GRE) can handle only the outer Source IP address and parts of the destination address. They may not provide sufficient load balancing entropy.

Generic UDP Encapsulation (GUE) is a UDP-based network encapsulation protocol that encapsulates IPv4 and IPv6 packets. GUE provides native UDP encapsulation and defines an additional header, which helps to determine the payload carried by the IP packet. The additional header can include items, such as a virtual networking identifier, security data for validating or authenticating the GUE header, congestion control data, and so on.

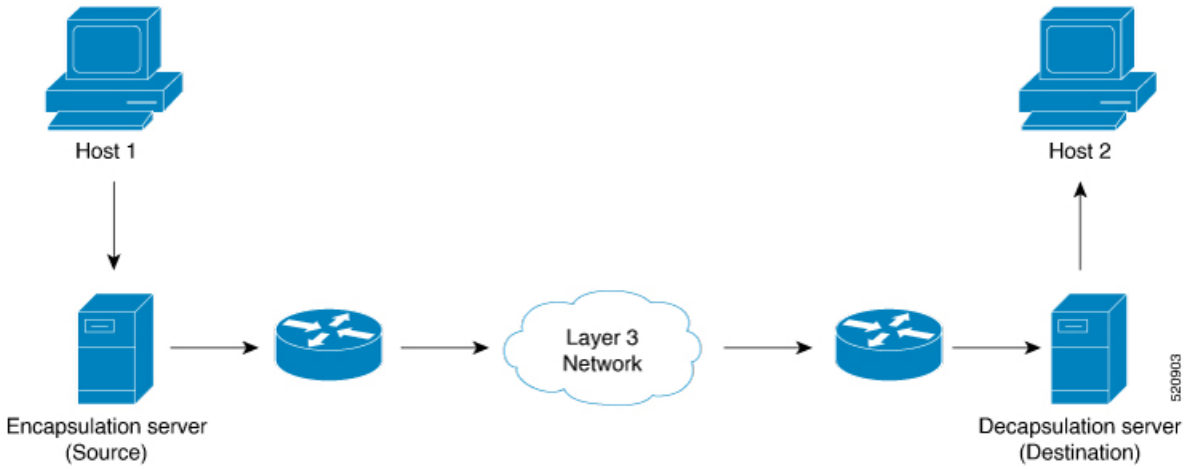
In GUE, the payload is encapsulated in an IP packet that can be IPv4 or IPv6 Carrier. The UDP header is added to provide extra hashing parameters, and optional payload demultiplexing. At the decapsulation node, the Carrier IP and UDP headers are removed, and the packet is forwarded based on the inner payload.

A GUE packet has the general format:

Figure 14: GUE Packet Format



For example, in this scenario, if the data stream is sent from Host 1 to Host 2. The server acts as a GUE encapsulator that sends the packets from Host 1. The server, on the other end receiving the data, validates the data for the valid carrier IP and UDP header and decapsulates the data.



GUE has various variants, but variant 1 of GUE allows direct encapsulation of IPv4 and IPv6 in UDP. This technique saves encapsulation overhead on links for the use of IP encapsulation, and also need not allocate a separate UDP port number for IP-over-UDP encapsulation.

Variant 1 has no GUE header, but a UDP packet carries an IP packet. The first two bits of the UDP payload is the GUE variant field and match with the first two bits of the version number in the IP header.

Benefits of using GUE

- Allows direct encapsulation of payloads, such as IPv4 and IPv6 in the UDP packet.
  - You can use UDP port for demultiplexing payloads.
  - You can use a single UDP port, allowing systems to employ parsing models to identify payloads.
- Leverages the UDP header for entropy labels by encoding a tuple-based source port.

- Leverages source IP addresses for load-balance encoding. The destination too could be terminated based on a subnet providing additional bits for entropy.
- Avoids special handling for transit nodes because they only see an IP-UDP packet with some payload..
- Eases implementation of UDP tunneling with GUE. This is because of the direct encapsulation method of the payloads into UDP.

Restrictions

- Supports Generic UDP Decapsulation for only variant 1.
- Receives IPv4 packets with the defined GUE port of 6080.
- Decapsulates IPv6 packets with the defined GUE port of 6080.
- Receives MPLS packets with the UDPoMPLS port of 6635.
- Range of source or destination ports is not supported.
- Range, Source, or Destination addresses are not supported, but subnet mask enteries are allowed.
- To perform decapsulation, a destination Port is mandatory.
- Terminating GRE after GUE or GUE after GRE is not supported.
- Terminating a label such as a VPN Deaggregation after GUE termination is not supported.
- Slow path support is not supported. To resolve the inner IP Adjacency, use the **cef proactive-arp-nd enable** command.
- Running the **clear all** command doesn't clear the interface of all its existing configurations.

Note To use only outer IP header (L3 and L4) for calculating the hashing for incoming GUE packets, use the **hw-module profile gue underlay-hash enable** command. Otherwise, by default, both outer IP header (L3 and L4) and inner IP header (L3 and L4) are considered for calculating the hashing for incoming GUE packets.

The **hw-module profile gue underlay-hash enable** command is currently not supported on the P100-based and Q100-based ASICs.

Configure GUE

Configuring GUE

Use the following configuration workflow to configure GUE:

1. Configure separate GUE decap tunnel UDP destination port numbers for IPv4, IPv6, and MPLS using **hw-module profile gue udp-dest-port** command.
2. Configure a traffic class: Create a traffic class and specify various criteria for classifying packets using the match commands, and an instruction on how to evaluate these match commands.
3. Configure a policy map: Define a policy map and associate the traffic class with the traffic policy.

4. Apply the policy for each VRF, and apply this policy on all the interfaces that are part of the VRF.

Configuration Example for GUE IPv4

1. Configure separate UDP port numbers for IPv4, IPv6, and MPLS using **hw-module profile gue udp-dest-port**command.

```
Router# configure
Router# hw-module profile gue udp-dest-port ipv4 6080 ipv6 6080 mpls 6635

Router# commit
```

Note While adding or removing the **hw-module profile gue udp-dest-port** command, you must reload the router.

2. Configure a traffic class:

```
Router# configure
Router(config)# class-map type traffic match-all udp-v4
Router(config-cmap)# match destination-address ipv4 220.100.20.0 255.255.255.255
Router(config-cmap)# match source-address ipv4 210.100.20.0 255.255.255.255
Router(config-cmap)# match protocol udp
Router(config-cmap)# match destination-port 6080
Router(config-cmap)# end-class-map
Router(config)# commit

Router(config)# class-map type traffic match-all udp-mpls1
Router(config-cmap)# match destination-address ipv4 220.100.20.0 255.255.255.255
Router(config-cmap)# match source-address ipv4 210.100.20.0 255.255.255.255
Router(config-cmap)# match protocol udp
Router(config-cmap)# match destination-port 6635
Router(config-cmap)# end-class-map
Router(config)# commit

Router(config)# class-map type traffic match-all udp-v6
Router(config-cmap)# match destination-address ipv4 220.100.20.0 255.255.255.255
Router(config-cmap)# match source-address ipv4 210.100.20.0 255.255.255.255
Router(config-cmap)# match protocol udp
Router(config-cmap)# match destination-port 6080
Router(config-cmap)# end-class-map
Router(config)# commit
```

3. Define a policy map, and associate the traffic class with the traffic policy:

```
Router(config)# policy-map type pbr magic-decap

Router(config-pmap)# class type traffic udp-v4
Router(config-pmap-c)# decapsulate gue variant 1
Router(config-pmap-c)# exit

Router(config-pmap)# class type traffic udp-v6
Router(config-pmap-c)# decapsulate gue variant 1
Router(config-pmap-c)# exit

Router(config-pmap)# class type traffic udp-mpls1
Router(config-pmap-c)# decapsulate gue variant 1
Router(config-pmap-c)# exit
```

```
Router(config-pmap)# class type traffic class-default
Router(config-pmap-c)# exit
Router(config-pmap)# end-policy-map
Router(config)# commit
Router(config)# exit
```

4. Apply the policy for each VRF:

```
Router# configure
Router(config)# vrf-policy
Router(config-vrf-policy)# vrf default address-family ipv4 policy type pbr input magic-decap
Router(config-vrf-policy)# commit
```

Running Configuration:

```
class-map type traffic match-all udp-v4
 match destination-address ipv4 220.100.20.0 255.255.255.255
 match source-address ipv4 210.100.20.0 255.255.255.255
 match protocol udp
 match destination-port 6080
 end-class-map
!
class-map type traffic match-all udp-v6
 match destination-address ipv4 220.100.20.0 255.255.255.255
 match source-address ipv4 210.100.20.0 255.255.255.255
 match protocol udp
 match destination-port 6080
 end-class-map
!
class-map type traffic match-all udp-mpls1
 match destination-address ipv4 220.100.20.0 255.255.255.255
 match source-address ipv4 210.100.20.0 255.255.255.255
 match protocol udp
 match destination-port 6635
 end-class-map
!
policy-map type pbr magic-decap
 class type traffic udp-v4
  decapsulate gue variant 1
 !
 class type traffic udp-v6
  decapsulate gue variant 1
 !
 class type traffic udp-mpls1
  decapsulate gue variant 1
 !
 class type traffic class-default
 !
 end-policy-map
!


vrf-policy
 vrf default address-family ipv4 policy type pbr input magic-decap
!
```

Verification

To view the set of counter values accumulated for the packets that match the class-map:

```
Router# show policy-map type pbr addr-family ipv4 statistics

VRF Name:       default
Policy-Name:    pmap
```

```
Policy Type:    pbr
Addr Family:    IPv4

Class:     cmap-loop1
     Classification statistics      (packets/bytes)
       Matched             :              0/0
     Transmitted statistics         (packets/bytes)
       Total Transmitted   :              0/0

Class:     cmap-loop6
     Classification statistics      (packets/bytes)
       Matched             :              0/0
     Transmitted statistics         (packets/bytes)
       Total Transmitted   :              0/0

Class:     cmap-loop2
     Classification statistics      (packets/bytes)
       Matched             :              0/0
     Transmitted statistics         (packets/bytes)
       Total Transmitted   :              0/0

Class:     cmap-loop3
     Classification statistics      (packets/bytes)
       Matched             :       198325306/17849277540
     Transmitted statistics         (packets/bytes)
       Total Transmitted   :       198325306/17849277540

Class:     cmap-loop4
    Classification statistics      (packets/bytes)
       Matched             :              0/0
     Transmitted statistics         (packets/bytes)
       Total Transmitted   :              0/0
```

To clear the policy-map counters for each class-map rule, use the **clear vrf** command:

```
Router# clear vrf default address-family ipv4 statistics
```

Outer IP Header-Driven Hash Computation for Incoming GUE Packets

When multiple paths with the same cost are available for forwarding traffic, ECMP hashing is used to determine the path to select for each packet. Each packet that needs to be forwarded is processed using a hashing algorithm. The hashing algorithm considers specific packet fields such as source IP, destination IP, source port, and destination port, and generates a hash value. The generated hash value is then mapped to one of the available paths. The selected path is used to forward the packet to its destination. The goal is to distribute the traffic evenly across the available paths to prevent congestion and utilize the network resources efficiently.

Now you can use only the outer IP header (L3 and L4) for calculating the hash value for incoming GUE packets and completely ignore the usage of the inner IP header. This functionality is configurable using the CLI command **hw-module profile gue underlay-hash**. This is supported for both GUE termination (decapsulation) and GUE transit (pass-through) nodes. By default, the feature is disabled; that is, both outer IP header (L3 and L4) and inner IP header (L3 and L4) are used for calculating the hashing for GUE packets.

Benefits

- **Load Balancing Efficiency:** By hashing only on the outer IP and L4 information, the packets with the same source and destination IP addresses and L4 ports consistently follow the same path in a load-balanced environment. This helps maintain session affinity or stickiness, as the inner IP addresses or L4 port numbers may change dynamically within the encapsulated packets.

- **Network Security:** Ignoring the inner IP helps preserve privacy and confidentiality within the encapsulated packets. By focusing on the outer IP and L4 headers, the network device does not have visibility into the inner IP addressing scheme or the specific content encapsulated within the packet, which enhances security.
- **Network Scalability:** Ignoring the inner IP reduces the complexity and overhead of packet processing, improving overall network performance and scalability, especially in high-throughput environments.

Configure Outer IP Header-Driven Hash Computation for Incoming GUE Packets

This section describes how to configure hashing with only outer IP for GUE packets.

Configuration Example

Use the following configuration to enable hashing with only outer IP for GUE packets:

```
Router# configure
Router# hw-module profile gue underlay-hash enable
Router# commit
```

Running Configuration

```
RP/0/RP0/CPU0:ios(config)#show running-config
hw-module profile gue underlay-hash enable
end
```

Verification

Following is the show command output before enabling hashing with only outer IP for GUE packets.

```
RP/0/RP0/CPU0:ios#show dpa objects sys location 0/RP0/CPU0  | include gue
  uint32_t gue_ipv4_port => 0
  uint32_t gue_ipv6_port => 0
  uint32_t gue_mpls_port => 0
  ofa_bool_t gue_underlay_hash => FALSE
```

Following is the show command output after enabling hashing with only outer IP for GUE packets.

```
RP/0/RP0/CPU0:ios#show dpa objects sys location 0/RP0/CPU0 | include gue
  uint32_t gue_ipv4_port => 0
  uint32_t gue_ipv6_port => 0
  uint32_t gue_mpls_port => 0
  ofa_bool_t gue_underlay_hash => TRUE
```

Flexible Assignment of UDP Port Numbers for Decapsulation

Table 43: Feature History Table

Feature Name	Release Information	Feature Description
Flexible Assignment of UDP Port Numbers for Decapsulation	Release 7.3.3	This feature gives you the flexibility to assign UDP port numbers from 1000 through 6400, through which IPv4, IPv6, and MPLS packets can be decapsulated. Such flexibility allows you to segregate the ingress traffic based on a QoS policy. In earlier releases, you could assign only default ports for decapsulation. The following command is introduced for this feature: `hw-module profile gue udp-dest-port ipv4 <port number> ipv6 <port number> mpls <port number>`

This feature provides decapsulation support for GUE packets. In GUE, the payload is encapsulated in an IP packet—IPv4 or IPv6 carrier. The UDP header is added to provide extra hashing parameters and optional payload demultiplexing. At the decapsulation node, the carrier IP and UDP headers are removed, and the packet is forwarded based on the inner payload. Prior to Release 7.3.3, packets were decapsulated using UDP port numbers 6080, 6615, and 6635 for IPv4, IPv6, and MPLS payloads respectively. Starting from Release 7.3.3, you can assign UDP port numbers from 1000 through 64000 to decapsulate IPv4, IPv6, and MPLS packets. Define different port numbers for IPv4, IPv6, and MPLS.

Guidelines for Setting up Decapsulation Using Flexible Port Numbers

Apply these guidelines while assigning flexible port numbers for decapsulation:

Packet	IPv4	IPv6	MPLS
UDP Outer Header	Configure IPv4 port on the hardware module.	Configure IPv6 port on the hardware module.	Configure MPLS port on the hardware module.
Encapsulation Outer Header	Configure an IPv4 encapsulation outer header that matches with the class map source.		
Inner Payload	Note that packets are forwarded based on the inner IPv4 payload.	Note that packets are forwarded based on the inner IPv6 payload.	Note that packets are forwarded based on the inner MPLS payload.

Note

- During the decapsulation of the IPv4, IPv6, and MPLS packets, the following headers are removed:
  - The UDP outer header
  - The IPv4 encapsulation outer header
- Select different values for each of these protocols. Valid port numbers are from 1000 through 64000.

Restrictions

The following restrictions are applicable while configuring unique GUE destination port numbers to decapsulate IPv4, IPv6, and MPLS packets using UDP:

- While configuring the tunnel, select one of the following:
  - Match only 16 unique source IP addresses as shown in the example:

    ```
    Router(config-cmap)#match source-address ipv4 210.100.20.0 255.255.255.255
    ```

  - Match a combination of 64 unique source and destination IP addresses as shown in the example:

    ```
    Router(config-cmap)# match destination-address ipv4 220.100.20.0 255.255.255.255
    Router(config-cmap)# match source-address ipv4 210.100.20.0 255.255.255.255
    ```

- The Classless Inter-Domain Routing (CIDR) value in the source IP address subnet mask must be only /32.
- The destination address subnet mask supports all CIDR values. However, the destination address along with the subnet mask must be unique for all the three UDP payload types—IPv4, IPv6, and MPLS. The configuration fails when the destination IP address and the subnet mask are the same for all three payloads as seen in this example:

  ```
  Router(config)#class-map type traffic match-all SRTE-GUE-DECAP-IPv4
  Router(config-cmap)#match destination-address ipv4 10.216.101.0 255.255.255.255
  ..
  Router(config)#class-map type traffic match-all SRTE-GUE-DECAP-IPv6
  Router(config-cmap)#match destination-address ipv4 10.216.101.0 255.255.255.255
  ..
  Router(config)#class-map type traffic match-all SRTE-GUE-DECAP-MPLS
  Router(config-cmap)#match destination-address ipv4 10.216.101.0 255.255.255.255
  ..
  ```

Configuring Port Numbers for Decapsulation

By configuring different port numbers on the destination router, you can match and direct traffic to different paths. For example, traffic for a specific video service can be decapsulated and sent through different ports. The steps that are involved in configuring port numbers for decapsulation are:

1. Configure the UDP destination ports for decapsulation of the required payloads.
2. Configure the traffic class to match the ports.
3. Define a policy map, and associate the traffic class with the traffic policy.

4. Apply the policy for each VRF.

Note For the hardware module flexible port configuration to take effect you must reload the line card.

Configuration Example

```
Hw-module configuration:
====================
Router# configure
Router# hw-module profile gue udp-dest-port ipv4 1001 ipv6 1002 mpls 1003

Class-map configuration:
====================
Router# configure
Router(config)# class-map type traffic match-all udp-v4
Router(config-cmap)# match protocol udp
Router(config-cmap)# match source-address ipv4 210.100.20.0 255.255.255.255
Router(config-cmap)# match destination-address ipv4 220.100.20.0 255.255.255.255
Router(config-cmap)# match destination-port 1001
Router(config-cmap)# end-class-map
Router(config)# commit

Router(config)# class-map type traffic match-all udp-v6
Router(config-cmap)# match protocol udp
Router(config-cmap)# match destination-address ipv4 220.100.20.0 255.255.255.255
Router(config-cmap)# match source-address ipv4 210.100.20.0 255.255.255.255
Router(config-cmap)# match destination-port 1002
Router(config-cmap)# end-class-map
Router(config)# commit

Router(config)# class-map type traffic match-all udp-mpls1
Router(config-cmap)# match protocol udp
Router(config-cmap)# match destination-address ipv4 220.100.20.0 255.255.255.255
Router(config-cmap)# match source-address ipv4 210.100.20.0 255.255.255.255
Router(config-cmap)# match destination-port 1003
Router(config-cmap)# end-class-map
Router(config)# commit

Ingress Policy-map configuration:
=========================
Router(config)# policy-map type pbr magic-decap
Router(config-pmap)# class type traffic udp-v4
Router(config-pmap-c)# decapsulate gue variant 1
Router(config-pmap-c)# exit

Router(config-pmap)# class type traffic udp-v6
Router(config-pmap-c)# decapsulate gue variant 1
Router(config-pmap-c)# exit

Router(config-pmap)# class type traffic udp-mpls1
Router(config-pmap-c)# decapsulate gue variant 1
Router(config-pmap-c)# exit

Router(config-pmap)# class type traffic class-default
Router(config-pmap-c)# exit
Router(config-pmap)# end-policy-map
Router(config)# commit
Router(config)# exit
```

```
Applying policy per VRF:
=======================================
Router# configure
Router(config)# vrf-policy
Router(config-vrf-policy)# vrf default address-family ipv4 policy type pbr input magic-decap
Router(config-vrf-policy)# commit
```

Running Configuration

```
!! File saved at 16:01:32 UTC Mon Feb 07 2022 by cisco
!! IOS XR Configuration 7.3.3.10I
!! Last configuration change at Mon Feb  7 15:35:11 2022 by cisco
!
logging console disable
username cisco
 group root-lr
 group cisco-support
 secret 10
$6$gHKmE1YZAo71BE1.$3KYogrvOdJxTRPZgYPGXUXkO4PqQMr2E6oYvJO4ngBmuaGsF2nAB/m1NP5Il3zh9HTzBI/k4r8PwWSbsARsmp.
!
vrf vrf-gre
 address-family ipv4 unicast
 !
 address-family ipv6 unicast
 !
!
line console
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
line default
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
!arp vrf default 29.0.1.2 0000.1122.2929 ARPA
call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method http
 !
!
ipv6 access-list abf6-gre
 1 permit ipv6 any any nexthop1 ipv6 201:0:1::2
!
ipv4 access-list abf-gre
 1 permit ipv4 any any nexthop1 ipv4 201.0.1.2
!
class-map type traffic match-all udp-v4
 match destination-address ipv4 220.100.20.0 255.255.255.255
 match source-address ipv4 210.100.20.0 255.255.255.255
 match protocol udp
 match destination-port 1001
 end-class-map
!
class-map type traffic match-all udp-v6
 match destination-address ipv4 220.100.20.0 255.255.255.255
 match source-address ipv4 210.100.20.0 255.255.255.255
 match protocol udp
```

```
 match destination-port 1002
 end-class-map
!
class-map type traffic match-all udp-mpls1
 match destination-address ipv4 220.100.20.0 255.255.255.255
 match source-address ipv4 210.100.20.0 255.255.255.255
 match protocol udp
 match destination-port 1003
 end-class-map
!
policy-map type pbr pbr-gre
 class type traffic class-default
  redirect ipv4 nexthop 202.0.1.2
 !
 end-policy-map
!
policy-map type pbr magic-decap
 class type traffic udp-v4
  decapsulate gue variant 1
 !
 class type traffic udp-v6
  decapsulate gue variant 1
 !
 class type traffic udp-mpls1
  decapsulate gue variant 1
 !
 class type traffic class-default
 !
 end-policy-map
!
interface Bundle-Ether25
 ipv4 address 25.0.1.1 255.255.255.0
 ipv6 address 25:0:1::1/64
 ipv6 enable
 shutdown
!
interface Bundle-Ether28
 ipv4 address 28.0.1.1 255.255.255.0
!
interface Loopback0
 ipv4 address 10.10.10.1 255.255.255.255
!
<output truncated>
interface MgmtEth0/RP0/CPU0/0
 ipv4 address dhcp
!
interface MgmtEth0/RP1/CPU0/0
 ipv4 address dhcp
!
interface BVI23
 ipv4 address 23.0.1.1 255.255.255.0
 ipv6 address 23:0:1::1/64
 ipv6 enable
 shutdown
!
interface BVI29
 ipv4 address 29.0.1.1 255.255.255.0
 ipv6 enable
 shutdown
!
interface HundredGigE0/0/0/0
 shutdown
!
```

```
<output truncated>
 l2transport
 !
!
interface HundredGigE0/0/0/24
 service-policy type pbr input pbr-gre
 ipv4 address 24.0.1.1 255.255.255.0
 ipv6 address 24:0:1::1/64
 ipv6 enable
!
interface HundredGigE0/0/0/24.24
 ipv4 address 24.0.24.1 255.255.255.0
 ipv6 enable
 encapsulation dot1q 24
!
interface HundredGigE0/0/0/25
 bundle id 25 mode on
!
interface HundredGigE0/0/0/26
 ipv4 address 26.0.1.1 255.255.255.0
 ipv6 address 26:0:1::1/64
 ipv6 enable
!
interface HundredGigE0/0/0/27
 ipv4 address 27.0.1.1 255.255.255.0
 ipv6 enable
!
interface HundredGigE0/0/0/27.27
 ipv4 address 27.0.27.1 255.255.255.0
 ipv6 address 27:0:27::1/64
 ipv6 enable
 shutdown
 encapsulation dot1q 27
!
interface HundredGigE0/0/0/28
 bundle id 28 mode active
!
interface HundredGigE0/0/0/29
 ipv4 address 29.0.1.1 255.255.255.0
 ipv6 enable
!
<output truncated>
interface HundredGigE0/1/0/24
 ipv4 address 124.0.1.1 255.255.255.0
 ipv6 address 124:0:1::1/64
 ipv6 enable
!
<output truncated>
!
interface HundredGigE0/1/0/30
 bundle id 28 mode active
!
interface HundredGigE0/1/0/31
 ipv4 address 31.0.1.1 255.255.255.0
 ipv6 address 31:0:1::1/64
 shutdown
!
<output truncated>
!
route-policy pass
  pass
end-policy
!
router static
```

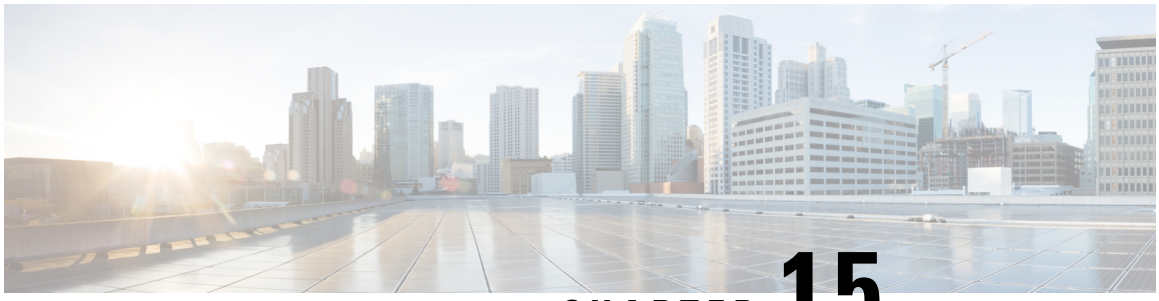
```
 address-family ipv4 unicast
  201.0.1.0/24 tunnel-ip1
  201.0.1.0/24 tunnel-ip2
  201.0.1.0/24 tunnel-ip3
  201.0.1.0/24 tunnel-ip4
 !
 address-family ipv6 unicast
  201:0:1::/64 tunnel-ip1
  201:0:1::/64 tunnel-ip2
  201:0:1::/64 tunnel-ip3
  201:0:1::/64 tunnel-ip4
 !
!
router ospf 10
 router-id 1.1.1.1
 area 0
  !  interface Bundle-Ether28
  interface Loopback0
  !
  interface HundredGigE0/0/0/26
  !
 !
!
!  interface HundredGigE0/0/0/27
!  interface HundredGigE0/0/0/27.27
router bgp 200
 bgp router-id 1.1.1.1
 address-family ipv4 unicast
  maximum-paths ibgp 64
 !
 ! redistribute connected
 ! neighbor 26.0.1.2
 !  remote-as 200
 !  address-family ipv4 unicast
 !  multipath
 !  route-policy pass in
 !  route-policy pass out
 !  next-hop-self
 neighbor 27.0.1.2
  remote-as 200
  address-family ipv4 unicast
   multipath
   route-policy pass in
   route-policy pass out
   next-hop-self
  !
 !
 neighbor 28.0.1.2
  remote-as 200
  address-family ipv4 unicast
   multipath
   route-policy pass in
   route-policy pass out
   next-hop-self
  !
 !
 neighbor 29.0.1.2
  remote-as 200
  address-family ipv4 unicast
   multipath
   route-policy pass in
   route-policy pass out
   next-hop-self
  !
```

```
 !
!
vrf-policy
 vrf default address-family ipv4 policy type pbr input magic-decap
!
l2vpn
 bridge group bg
  bridge-domain bd
 !   interface HundredGigE0/0/0/29
 !    static-mac-address 0000.1122.2929
 !   routed interface BVI29
 bridge group bg1
  bridge-domain bd1
   interface HundredGigE0/0/0/23
    static-mac-address 0000.1122.2323
   !
   routed interface BVI23
   !
  !
 !
!
mpls static
 interface HundredGigE0/0/0/24
 lsp gre
  in-label 35001 allocate per-prefix 202.0.1.2/32
  forward
   path 1 nexthop tunnel-ip1 out-label 35002
   path 2 nexthop tunnel-ip2 out-label 35002
  !
 !
!
ssh server vrf default
hw-module profile gue udp-dest-port ipv4 1001 ipv6 1002 mpls 1003
end
```

Verification

Run the **show ofa objects sys location 0/0/CPU0 | inc gue** command in the XR Config mode to verify that the unique GUE port numbers have been configured to decapsulate IPv4, IPv6, and MPLS payloads.

```
Router#show ofa objects sys location 0/0/CPU0 | inc gue
uint32_t gue_ipv4_port => 1001
uint32_t gue_ipv6_port => 1002
uint32_t gue_mpls_port => 1003
```



CHAPTER 15

Controlling the TTL Value of Inner Payload Header

Cisco 8000 Routers allow you to control the TTL value of inner payload header of IP-in-IP tunnel packets before it gets forwarded to the next-hop router. This feature enables a router to forward custom formed IP-in-IP stacked packets even if the inner packet TTL is 1. Therefore, this feature enables you to measure the link-state and path reachability from end to end in a network.

Note After you enable or disable the decrement of the TTL value of the inner payload header of a packet, you do not need to reload the line card.

Configuration

To disable the decrement of the TTL value of inner payload header of an IP-in-IP packet, use the following steps:

1. Enter the global configuration mode.
2. Disable the decrement of TTL value of inner payload header of an IP-in-IP packet.

Configuration Example

```
/* Enter the Global Configuration mode. */
Router# configure

/* Disable the decrement of TTL value of inner payload header of an IP-in-IP packet. */
Router(config)# hw-module profile cef ttl tunnel-ip decrement disable
Router(config)# commit
```

Note Starting from Release 7.3.3, Cisco IOS XR 8000 router supports a maximum of 16 IP-in-IP decap tunnels with unique source addresses. If 15 unique tunnel sources are configured that is rounded to 95% of the tunnel hardware resource OOR threshold level. As a result, the OOR State displays *Red* in **show controllers npu resources sipidxtbl location all** command output.

Associated Commands

- hw-module profile cef ttl tunnel-ip decrement disable

- IP-in-IP Decapsulation, on page 256
- ECMP Hashing Support for Load Balancing, on page 264

IP-in-IP Decapsulation

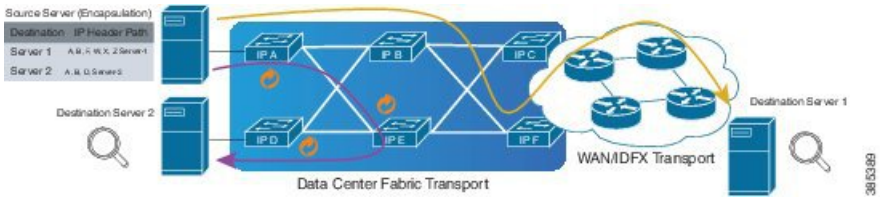
IP-in-IP encapsulation involves the insertion of an outer IP header over the existing IP header. The source and destination address in the outer IP header point to the endpoints of the IP-in-IP tunnel. The stack of IP headers is used to direct the packet over a predetermined path to the destination, provided the network administrator 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.

In IP-in-IP encapsulation and decapsulation has two types of packets. The original IP packets that are encapsulated are called Inner packets and the IP header stack added while encasulation are called the Outer packets.

Note The router only supports decapsulation and no encapsulation. Encapsulation is done by remote routers.

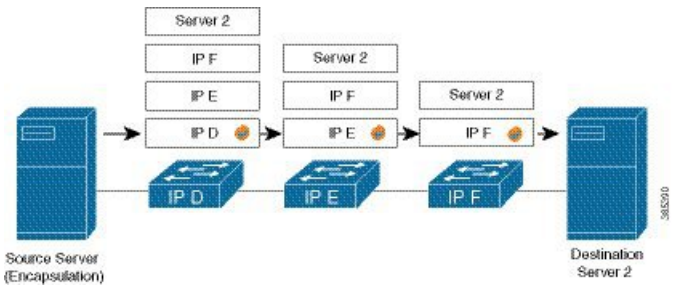
The following topology describes a use case where IP-in-IP encapsulation and decapsulation are used for different segments of the network from source to destination. The IP-in-IP tunnel consists of multiple routers that are used to decapsulate and direct the packet through the data center fabric network.

Figure 15: IP-in-IP Decapsulation Through a Data Center Network



The following illustration shows how the stacked IPv4 headers are decapsulated as they traverse through the decapsulating routers.

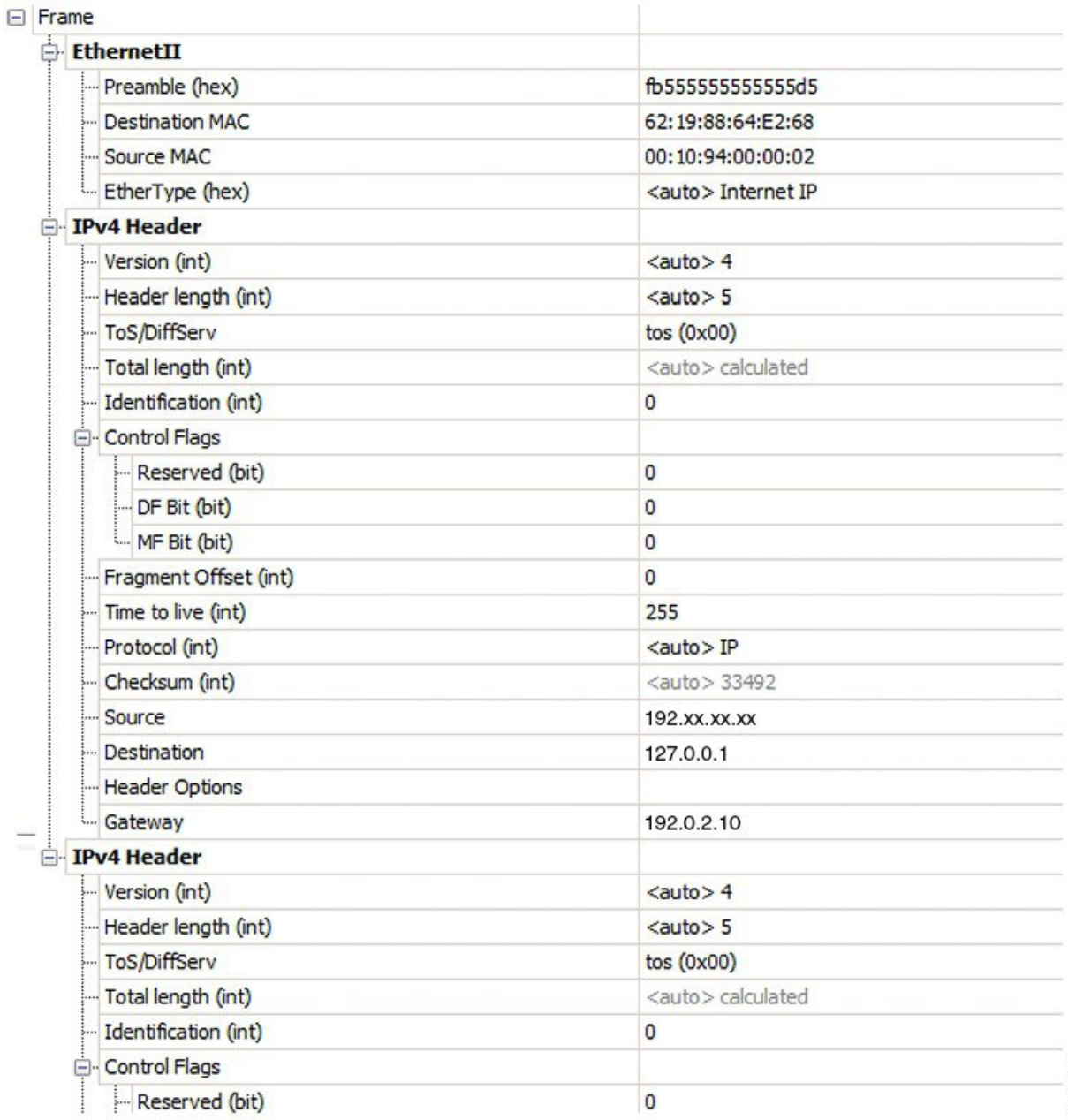
Figure 16: IP Header Decapsulation



Stacked IP Header in an Encapsulated Packet

The encapsulated packet has an outer IPv4 header that is stacked over the original IPv4 header, as shown in the following illustration.

Figure 17: Encapsulated Packet



Configuration

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

```
Router(config)# interface loopback 0
Router(config-if)# ipv4 address 127.0.0.1/32
Router(config-if)# no shutdown
Router(config-if)# interface tunnel-ip 10
Router(config-if)# ipv4 unnumbered loopback 1
Router(config-if)# tunnel mode ipv4 decap
Router(config-if)# tunnel source loopback 0
```

- **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 decapsulation.
- **tunnel source**: indicates the source address for the IP-in-IP decap tunnel with respect to the router interface.

Note You can configure the tunnel destination only if you want to decapsulate packets from a particular destination. If no tunnel destination is configured, then all the ip-in-ip ingress packets on the configured interface are decapsulated.

Running Configuration

```
Router# show running-config interface tunnel-ip 10
...
interface tunnel-ip 10
ipv4 unnumbered loopback 1
tunnel mode ipv4 decap
```

Extended ACL to Match the Outer Header for IP-in-IP Decapsulation

Starting with Cisco IOS XR Software Release 7.0.14, extended ACL has to match on the outer header for IP-in-IP Decapsulation. Extended ACL support reduces mirrored traffic throughput. This match is based only on the IPv4 protocol, and extended ACL is applied to the received outermost IP header, even if the outer header is locally terminated.

Sample configuration:

```
Router#show running-config interface bundle-Ether 50.5
Tue May 26 12:11:49.017 UTC
interface Bundle-Ether50.5
ipv4 address 101.1.5.1 255.255.255.0
encapsulation dot1q 5
ipv4 access-group ExtACL_IPinIP ingress
ipv4 access-group any_dscpegg egress
!


Router#show access-lists ipv4 ExtACL_IPinIP hardware ingress location$
Tue May 26 12:11:55.940 UTC
ipv4 access-list ExtACL_IPinIP
10 permit ipv4 192.168.0.0 0.0.255.255 any ttl gt 150
11 deny ipv4 172.16.0.0 0.0.255.255 any fragments
12 permit ipv4 any any
```

Decapsulation using tunnel source direct

Table 44: Feature History Table

Feature Name	Release Information	Feature Description
Decapsulation using tunnel source direct	Release 7.5.3	Tunnel source direct allows you to decapsulate the tunnels on any L3 interface on the router. You can use the **tunnel source direct** configuration command to choose the specific IP Equal-Cost Multipath (ECMP) links for troubleshooting, when there are multiple IP links between two devices.

To debug faults in various large networks, you may have to capture and analyze the network traffic at a packet level. In datacenter networks, administrators face problems with the volume of traffic and diversity of faults. To troubleshoot faults in a timely manner, DCN administrators must identify affected packets inside large volumes of traffic. They must track them across multiple network components, analyze traffic traces for fault patterns, and test or confirm potential causes.

In some networks, IP-in-IP decapsulation is currently used in network management, to verify ECMP availability and to measure the latency of each path within a datacenter.

The Network Management System (NMS) sends IP-in-IP (IPv4 or IPv6) packets with a stack (multiple) of predefined IPv4 or IPv6 headers (device IP addresses). The destination device at each hop removes the outside header, performs a lookup on the next header, and forwards the packets if a route exists.

Using the **tunnel source direct** command, you can choose the specific IP Equal-Cost Multipath (ECMP) links for troubleshooting, when there are multiple IP links between two devices.

Tip You can programmatically configure and manage the Ethernet interfaces using `openconfig-ethernet-if.yang` and `openconfig-interfaces.yang` OpenConfig data models. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

Guidelines and Limitations

The following guidelines are applicable to this feature.

- The **tunnel source direct** command is only compatible with 'tunnel mode decap' for IP-in-IP decapsulation.
- The source-direct tunnel is always operationally `up` unless it is administratively shut down. The directly connected interfaces are identified using the **show ip route direct** command.
- All Layer 3 interfaces that are configured on the device are supported.
- Platform can accept and program only certain number of IP addresses. The number of IP addresses depends on the make of the platform linecard (LC). Each LC can have different number of Network Processor (NP) slices and interfaces.

- Only one source-direct tunnel per address-family is supported for configuration.
- Regular decapsulation tunnels which have specific source address, are supported. However, the tunnel's specific source address must not be part of any interface.

The following functionalities are not supported for the **tunnel source direct** option.

- GRE tunneling mode.
- VRF (only default VRF is supported).
- ACL and QoS on the tunnels.
- Tunnel encapsulation.
- Tunnel NetIO DLL: Decapsulation is not supported if the packet is punted to slow path.

Configure Decapsulation Using Tunnel Source Direct

Configuration

The **tunnel source direct** configures IP-in-IP tunnel decapsulation on any directly connected IP addresses. This option is now supported only when the IP-in-IP decapsulation is used to source route the packets through the network.

This example shows how to configure IP-in-IP tunnel decapsulation on directly connected IP addresses:

```
Router# configure terminal
Router(config)#interface Tunnel4
  Router(config)#tunnel mode ipv4 decap
  Router(config)#tunnel source direct
  Router(config)#no shutdown
```

This example shows how to configure IP-in-IP tunnel decapsulation on IPv6 enabled networks:

```
Router# configure terminal
Router(config)#interface Tunnel6
  Router(config)#tunnel mode ipv6 decap
  Router(config)#tunnel source direct
  Router(config)#no shutdown
```

Verifying the Configuration

The following example shows how to verify IP-in-IP tunnel decapsulation with **tunnel source direct** option:

```
Router#show running-config interface tunnel 1
interface Tunnel1
  tunnel mode ipv6ipv6 decapsulate-any
  tunnel source direct
  no shutdown

Router#show interface tunnel 1
Tunnel1 is up    Admin State: up
MTU 1460 bytes, BW 9 Kbit
Tunnel protocol/transport IPv6/DECAPANY/IPv6
Tunnel source - direct
Tx    0 packets output, 0 bytes    Rx    0 packets input, 0 bytes
```

Configure Tunnel Destination with an Object Group

Table 45: Feature History Table

Feature Name	Release Information	Description
Configure Tunnel Destination with an Object Group	Release 7.5.4	You can now assign an object group as the destination for an IP-in-IP decapsulation tunnel. With this functionality, you could configure an IPv4 or IPv6 object group consisting of multiple IPv4 or IPv6 addresses as the destination for the tunnel instead of a single IPv4 or IPv6 address. Using an object group instead of a singular IP address. This helps reduce the configuration complexity in the router by replacing the multiple tunnels with one destination with a single decapsulation tunnel that supports a diverse range of destinations The feature introduces these changes: • **CLI**: New **tunnel destination** command. • **YANG Data Model**: New **object-group** option supported in Cisco-IOS-XR-um-if-tunnel-cfg.yang Cisco native model (see GitHub).

In IP-in-IP Decapsulation, the router accepts a packet on a tunneled interface only when the tunnel IP address matches the source IP address of the incoming packets. With this implementation, the user needs to configure separate interface tunnels for each IP address that the router needs to receive the traffic packets. This limitation often leads to configuration overload on the router.

You can eliminate the configuration overload on the router by assigning an object group as the tunnel destination for IPv4 and IPv6 traffic types. That is, the router matches the source IP address of the incoming packet against the object group available as the tunnel destination. The decapsulation tunnel accepts the incoming traffic packets when there's a match between the packet source and the object group. Otherwise, the router drops the packets.

Restrictions

The following restrictions are applicable to the tunnel destination with an object group feature:

- GRE tunnels don't support configuring object groups as the tunnel destination.

- The router supports configuring tunnel destination with an object group only when the tunnel source is tunnel source direct.
- You can configure the object group as tunnel destination only on default VRF.
- Configuring object groups as the tunnel destination isn't applicable to tunnel encapsulation.
- Subinterfaces don't support configuring object groups as the tunnel destination.
- Configuring object groups as the tunnel destination feature is mutually exclusive with ACL and QoS features.
- The tunnel destination feature supports only IPv4 and IPv6 object groups.
- The router does not support changing tunnel configuration after its creation. Configure the tunnel source direct and tunnel destination with an object group while creating the tunnel only.

Prerequisites

- Define an object group including the network elements for the tunnel destination.
- Enable the tunnel source direct feature. For more information, see Decapsulation using tunnel source direct, on page 232.

Configuration example

This section provides an example for configuring the tunnel destination with an object group.

IPv4 configuration

```
Router# configure
/* Configure the IPv4 object group */
Router(config)# object-group network ipv4 Test_IPv4
Router(config-object-group-ipv4)# 192.0.2.0/24
Router(config-object-group-ipv4)# 198.51.100.0/24
Router(config-object-group-ipv4)# 203.0.113.0/24
Router(config-object-group-ipv4)# commit
Router(config-object-group-ipv4)# exit

/* Enters the tunnel configuration mode */
Router(config)# interface tunnel-ip 1

/* Configures the tunnel mode */
Router(config-if)# tunnel mode ipv4 decap

/* Configures the tunnel to accept all packets with destination address matching the IP
addresses on the router */
Router(config-if)# tunnel source direct

/* Configures the destination of the tunnel as the defined object-group */
Router(config-if)# tunnel destination object-group ipv4 Test_IPv4

Router(config-if)# no shutdown
Router(config-if)# commit
Router(config-if)# exit
```

IPv6 configuration

```
Router# configure
/* Configure the IPv6 object group */
Router(config)# object-group network ipv6 Test_IPv6
```

```
Router(config-object-group-ipv6)# 2001:DB8::/32
Router(config-object-group-ipv6)# 2001:DB8::/48
Router(config-object-group-ipv6)# commit
Router(config-object-group-ipv6)# exit

/* Enters the tunnel configuration mode */
Router(config)# interface tunnel-ip 2

/* Configures the tunnel mode */
Router(config-if)# tunnel mode ipv6 decap

/* Configures the tunnel to accept all packets with destination address matching the IP
addresses on the router */
Router(config-if)# tunnel source direct

/* Configures the destination of the tunnel as the defined object-group */
Router(config-if)# tunnel destination object-group ipv6 Test_IPv6

Router(config-if)# no shutdown
Router(config-if)# commit
Router(config-if)# exit
```

Running Configuration

```
Router# show running-config object-group
object-group network ipv4 Test_IPv4
 192.0.2.0/24
 198.51.100.0/24
 203.0.113.0/24
!
object-group network ipv6 Test_IPv6
 2001:DB8::/32
 2001:DB8::/48
!

Router#show running-config interface tunnel-ip 1
interface tunnel-ip1
 tunnel mode ipv4 decap
 tunnel source direct
 tunnel destination object-group ipv4 Test_IPv4
!

Router#show running-config interface tunnel-ip 2
Fri Nov 29 11:26:54.716 UTC
interface tunnel-ip2
 tunnel mode ipv6 decap
 tunnel source direct
 tunnel destination object-group ipv6 Test_IPv6
!
```

Verification

```
Router# show tunnel ip ea database

----- node0_0_CPU0 -----
tunnel ifhandle 0x80022cc
tunnel source 161.115.1.2
tunnel destination address group Test_IPv4
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
```

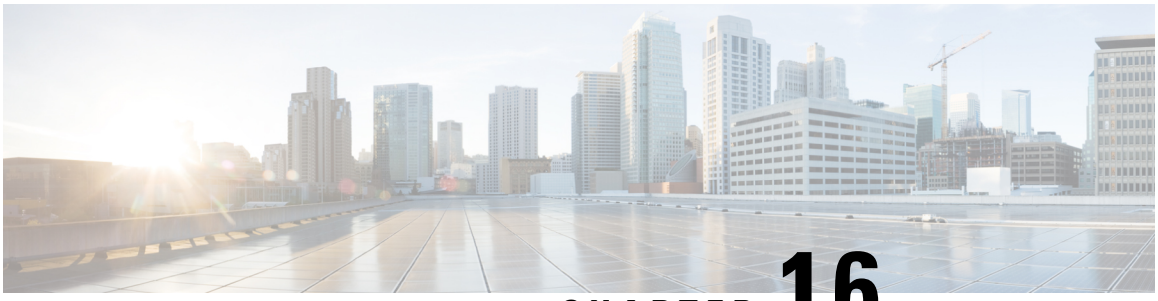
```
Encap
tunnel mtu 1500
tunnel tos 0
tunnel ttl 255
tunnel adjacency flags 0x1
tunnel o/p interface handle 0x0
tunnel key 0x0, entropy length 0 (mask 0xffffffff)
tunnel QT next 0x0
tunnel platform data (nil)
Platform:
Handle: (nil)
Decap ID: 0
Decap RIF: 0
Decap Recycle Encap ID: 0x00000000
Encap RIF: 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
```

ECMP Hashing Support for Load Balancing

The system inherently supports the n-tuple hash algorithm. The first inner header in the n-tuple hashing includes the source port and the destination port of UDP / TCP protocol headers.

The load balancing performs these functions:

- Incoming data traffic is distributed over multiple equal-cost connections.
- Incoming data traffic is distributed over multiple equal-cost connections member links within a bundle interface.
- Layer 2 bundle and Layer 3 (network layer) load-balancing decisions are taken on IPv4, and IPv6. If it is an IPv4 or an IPv6 payload, then an n-tuple hashing is done.
- An n-tuple hash algorithm provides more granular load balancing and used for load balancing over multiple equal-cost Layer 3 (network layer) paths. The Layer 3 (network layer) path is on a physical interface or on a bundle interface.
- The n-tuple load-balance hash calculation contains:
    - Source IP address
    - Destination IP address
    - IP Protocol type
    - Router ID
    - Source port
    - Destination port
    - Input interface
    - Flow-label (for IPv6 only)



CHAPTER 16

Configuring 400G Digital Coherent Optics

Table 46: Feature History Table

Feature Name	Release Information	Description
Extended Support for DP04QSDD-HE0 Optical Module	Release 7.10.2	From this release, the DP04QSDD-HE0 optical module is supported on the following router and line cards - Router: • Cisco 8202-32FH-M Line cards: • 88-LC0-34H14FH • 88-LC0-36FH
Extended Support for DP04QSDD-HE0 Optical Module	Release 7.10.1	This release introduces support for the Cisco 400G QSFP-DD High-Power (Bright) Optical Module DP04QSDD-HE0, Ethernet Variant on the Cisco 8608 router.

Feature Name	Release Information	Description
oFEC Traffic Configuration for QDD-400G-ZRP-S	Release 7.9.1	New Modulation and DAC Rate traffic configurations are supported on QDD-400G-ZRP-S optical module: • 400G-TXP-1x1-16 QAM • 4x100G-MXP-1x1-16 QAM • 3x100G-MXP-1x1-8 QAM • 2x100G-MXP-1x1-QPSK • 2x100G-MXP-1x1.25-16 QAM This increases the interoperability of the QDD-400G-ZRP-S optical module across network components supporting these formats.
Support for DP04QSDD-HE0 Optical Module	Release 7.9.1	The Cisco 400G QSFP-DD High-Power (Bright) Optical Module is an enhanced version of the currently available QSFP-DD ZR+ Optical Module. It leverages the same operational modes but provides a major enhancement by increasing the Tx Optical Power up to +1dBm. From this release, the DP04QSDD-HE0 optical module is supported on the Cisco 8201-32FH and Cisco 8201-24H8FH routers.
Support for QDD-400G-ZRP-S Optical Module	Release 7.9.1	This release introduces support for the Cisco 400G QSFP-DD-ZRP-S Ethernet Variant on the Cisco 88-LC0-34H14FH line card.

Cisco offers a range of the new 400G Digital Coherent QSFP-DD optical modules. The optical modules that are available are:

- QDD-400G-ZR-S
- QDD-400G-ZRP-S
- DP04QSDD-HE0

This chapter describes various optical modules and their supported configurations. The following fixed-port routers, line cards, from the indicated Cisco IOS XR software releases, support these optical modules.

Table 47: Fixed-Port Routers and Line Cards that Support various Optical Modules from Indicated Cisco IOS XR Software Releases

Fixed-Port Routers	Optics PID	Minimum IOS XR Software Release	
Cisco 8201	QDD-400G-ZR-S	Release 7.3.15	
	QDD-400G-ZRP-S		
	Cisco 8202	QDD-400G-ZR-S	Release 7.3.15
QDD-400G-ZRP-S			
Cisco 8101-32FH		QDD-400G-ZR-S QDD-400G-ZRP-S	Release 7.3.2
Cisco 8201-32FH	DP04QSDD-HE0	Release 7.9.1	
Cisco 8201-24H8FH	DP04QSDD-HE0	Release 7.9.1	
Cisco 8608	DP04QSDD-HE0	Release 7.10.1	
Cisco 8202-32-FH-M	DP04QSDD-HE0	Release 7.10.2	
Line Cards	**Optics PID**	**Minimum IOS XR Software Release**	
8800-LC-36FH	QDD-400G-ZR-S	Release 7.3.15	
	QDD-400G-ZRP-S		
	88-LC0-36FH-M	QDD-400G-ZR-S	Release 7.3.15
QDD-400G-ZRP-S			
88-LC0-36FH		QDD-400G-ZR-S	Release 7.3.2
		QDD-400G-ZRP-S	
		DP04QSDD-HE0	Release 7.10.2
88-LC0-34H14FH	QDD-400G-ZRP-S	Release 7.9.1	
	DP04QSDD-HE0	Release 7.10.2	

Note QDD-400G-ZR-S and QDD-400G-ZRP-S are not supported on 8102-64H fixed-port routers.

Note The Tail Trace Identifier (TTI) is not supported on QDD-400G-ZR-S and QDD-400G-ZRP-S optics.

QDD-400G-ZRP-S and DP04QSDD-HE0 are not supported on odd-numbered ports of the following routers and line cards:

- Cisco 8201
- Cisco 8202
- 8800-LC-36FH
- 88-LC0-36FH-M

The 400G Digital Coherent QSFP-DD optical modules enable wavelength-division multiplexing (WDM) functionality in the router. These optical modules are DWDM C-band (196.1 THz to 191.3 THz) tunable optical modules. They can be used in both transponder and muxponder modes.

Cisco IOS XR software creates optics and coherent DSP controllers to configure and monitor the performance of the 400G Digital Coherent QSFP-DD optical modules. Optics controllers are used to configure and monitor optical parameters, such as frequency, chromatic dispersion, transmitted output power, modulation, and so on. Coherent DSP controllers are used to monitor network performance parameters like pre- and post-forward error correction (FEC) bit-error rate (pre-FEC BER, post-FEC BER), error corrected bits (EC-BITS), and so on. Forward error correction (FEC) is configured using optical controllers and monitored using coherent DSP controllers.

The 400G Digital Coherent QSFP-DD optical modules support traffic configuration and firmware download. The Cisco IOS XR software collects performance monitoring data and alarms using versatile DOM (VDM).

Due to more power consumption by the 400G Digital Coherent QSFP-DD optical modules, the Cisco IOS XR software operates the fans at an higher speed to cool these optical modules.

The 400G Digital Coherent QSFP-DD optical module configuration is divided into the following categories:

- Traffic configuration – Comprises configuring DAC rate, muxponder mode, modulation, and FEC parameters. Applicable for optics controllers:
  - Configuring DAC Rate, on page 284
  - Configuring Muxponder Mode, on page 278
  - Configuring Modulation, on page 282
  - Configuring FEC, on page 286
- Optical configuration – Comprises configuring frequency, chromatic dispersion, and optical transmit power. Applicable for optics controllers:
  - Configuring Frequency, on page 272
  - Configuring Chromatic Dispersion, on page 274
  - Configuring Optical Transmit Power, on page 276
- Performance monitoring (PM) – Enables or disables performance monitoring in optical modules. You can also configure PM parameters that comprise signal power, chromatic dispersion, optical signal-to-noise ratio (OSNR), and differential group delay (DGD). Applicable for optics controllers and coherent DSP controllers:
  - Configuring Performance Monitoring, on page 290
  - Configuring PM Parameters, on page 290

- Loopback configuration – Configures loopback. Applicable for coherent DSP controller:
  - Configuring Loopback, on page 287
- Alarms threshold configuration – Configures thresholds for monitoring alarms that include optical signal-to-noise ratio (OSNR), differential group delay (DGD), chromatic dispersion (cd high and low), and so on. Applicable for optics controllers:
  - Configuring Alarms Threshold, on page 294

The following table contains the possible traffic configuration values for the 400G Digital Coherent QSFP-DD optical modules, in the transponder and muxponder mode:

Table 48: 400G Digital Coherent QSFP-DD Traffic Configuration Values

	QDD-400G-ZR-S	QDD-400G-ZRP-S	DP04QSDD-HE0	DP04QSDD-ER1	DP01QSDD-ZF1
Client Speed	1x400G, 4x100G	1x400G, 4x100G, 3x100G, 2x100G, 1x100G **Note** Release 7.3.15 supports only 1x400 and 4x100 client speed.	1x400G, 4x100G, 3x100G, 2x100G, 1x100G	1x400G, 2x200G, 4x100G	1x100G
Trunk Speed	400G	400G , 300G, 200G, 1x100 **Note** Release 7.3.15 supports only 400G trunk speed.	400G, 300G, 200G, 100G	400G	100G
Frequency	C-Band, 196.1 To 191.3 THz	C-Band, 196.1 To 191.3 THz	C-Band, 196.1 To 191.3 THz	193.7THz	193.7THz
FEC	cFEC	oFEC, cFEC	oFEC	cFEC, oFEC	oFEC
Modulation	16QAM	16QAM, 8QAM, QPSK Release 7.3.15 supports only 16QAM.	16QAM, 8QAM, QPSK	16QAM	QPSK

	QDD-400G-ZR-S	QDD-400G-ZRP-S	DP04QSDD-HE0	DP04QSDD-ER1	DP01QSDD-ZF1
DAC-Rate	1x1	1x1.25 (oFEC), 1x1 (cFEC)	1x1.25, 1x1	1x1	1x1
Chromatic Dispersion (CD)	-2400 to +2400	Release 7.3.15: -80000 to +80000 Release 7.3.2: -160000 to +160000	-160000 to +160000	-2400 to +2400	-2400 to +2400
Transmitted (Tx) Power	Each optical module has its own transmitting (TX) power range. You can change the transmitting (TX) power value based on the module capability.	Each optical module has its own transmitting (TX) power optimal values. You can change the transmitting (TX) power value based on the module capability.	Each optical module has its own transmitting (TX) power optimal values. You can change the transmitting (TX) power value based on the module capability.	Fixed at maximum output around -9dBm.	Fixed at maximum output around -6dBm.

QDD-400G-ZR-S Transponder and Muxponder Configuration Values

The following table contains the possible Transponder and Muxponder configuration values for the QDD-400G-ZR-S optical module:

Table 49: QDD-400G-ZR-S Transponder and Muxponder Configuration Values

TXP/MXP	Client	Trunk	Modulation	FEC	DAC Rate
400G-TXP	1 client, 400G speed	1 trunk, 400G	16 QAM	cFEC	1x1
4x100G-MXP	4 clients, 100G speed	1 trunk, 400G	16 QAM	cFEC	1x1

QDD-400G-ZRP-S Transponder and Muxponder Configuration Values

The following table contains the possible Transponder and Muxponder configuration values for the QDD-400G-ZRP-S optical module:

Table 50: QDD-400G-ZRP-S Transponder and Muxponder Configuration Values

TXP/MXP	Client	Trunk	Modulation	FEC	DAC Rate	OpenZR+ Support
400G-TXP	1 Client, 400G speed	1 trunk, 400G speed	16 QAM	oFEC	1x1.25	

TXP/MXP	Client	Trunk	Modulation	FEC	DAC Rate	OpenZR+ Support
400G-TXP	1 Client, 400G speed	1 trunk, 400G speed	16 QAM	oFEC	1x1	
400G-TXP	1 Client, 400G speed	1 trunk, 400G speed	16 QAM	cFEC	1x1	
4x100G-MXP	4 clients, 100G speed	1 trunk, 400G speed	16 QAM	oFEC	1x1.25	
4x100G-MXP	4 Client, 100G speed	1 trunk, 400G speed	16 QAM	oFEC	1x1	
4x100G-MXP	4 clients, 100G speed	1 trunk, 400G speed	16 QAM	cFEC	1x1	
3x100G-MXP	3 clients, 100G speed	1 trunk, 400G speed	8 QAM	oFEC	1x1.25	
3x100G-MXP	3 Client, 100G speed	1 trunk, 400G speed	8 QAM	oFEC	1x1	
2x100G-MXP	2 clients, 100G speed	1 trunk, 200G speed	QPSK	oFEC	1x1.50	
2x100G-MXP	2 Client, 100G speed	1 trunk, 400G speed	QPSK	oFEC	1x1	
2x100G-MXP	2 Client, 100G speed	1 trunk, 400G speed	16 QAM	oFEC	1x1.25	
1x100G-MXP	1 client, 100G speed	1 trunk, 100G speed	QPSK	oFEC	1x1.50	

The high optical performance DP04QSDD-HE0 QSFP-DD pluggable coherent optical module is developed for easy deployment in Reconfigurable Optical Add/Drop Multiplexer (ROADM) line systems.

DP04QSDD-HE0 Transponder and Muxponder Configuration Values

The following table contains the possible Transponder and Muxponder configuration values for the DP04QSDD-HE0 optical module:

Table 51: DP04QSDD-HE0 Transponder and Muxponder Configuration Values

TXP/MXP	Client	Trunk	Modulation	FEC	DAC Rate
400G-TXP	1 Client, 400G speed	1 trunk, 400G speed	16 QAM	oFEC	1x1.25
100G-TXP	1 Client, 100G speed	1 trunk, 400G speed	QPSK	oFEC	1x1.50

TXP/MXP	Client	Trunk	Modulation	FEC	DAC Rate
4x100G- MXP	4 clients, 100G speed	1 trunk, 400G speed	16 QAM	oFEC	1x1.25
3x100G-MXP	3 clients, 100G speed	1 trunk, 400G speed	8 QAM	oFEC	1x1.25
2x100-MXP	2 Client, 100G speed	2 Client, 100G speed	QPSK	oFEC	1x1.50

Configuring Frequency

You can configure frequency on optics controllers. You can select any C band frequency between the range 196.1 to 191.3 THz, in both ITU and NON-ITU channels.

Note The 100MHz-grid keyword accepts only frequency values as user input. The 50GHz-grid keyword accepts frequency, ITU-channel, or wavelength values as user input. The Cisco IOS XR software then caculates the frequency for a given wavelength or ITU-channel.

Frequency Configuration Example

The following example shows how to configure frequency on the optics controller:

```
Router#config
Router(config)#controller optics 0/2/0/16
Router(config-Optics)#dwdm-carrier 100MHz-grid frequency 1921500
Router(config-Optics)#commit
Router(config-Optics)#exit
Router(config)#exit
```

Running Configuration

This example shows the running configuration:

```
Router#show run controller optics 0/2/0/16
Fri May 28 01:42:32.488 UTC
```

```
controller Optics0/2/0/16
 dwdm-carrier 100MHz-grid frequency 1921500
 cd-low-threshold -5000
 cd-high-threshold -5000
!
```

Verification

This example shows how to verify the frequency configuration:

```
Router#show controller optics 0/2/0/16
Fri May 28 01:47:23.953 UTC
 Controller State: Up
 Transport Admin State: In Service
 Laser State: Off
 LED State: Off
 FEC State: FEC ENABLED
 Optics Status
         Optics Type:  QSFPDD 400G ZRP
         DWDM carrier Info: C BAND, MSA ITU Channel=80, Frequency=192.15THz,
         Wavelength=1560.200nm
         Alarm Status:
         -------------
         Detected Alarms: None
         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 0
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 mA
         Actual TX Power = -40.00 dBm
         RX Power = -40.00 dBm
         RX Signal Power = -40.00 dBm
         Frequency Offset = 0 MHz
         Laser Temperature = 0.00 Celsius
         Laser Age = 0 %
         DAC Rate = 1x1.25
         Performance Monitoring: Enable
         THRESHOLD VALUES
         ----------------
         Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
         ------------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         13.0      -24.0          10.0        -22.0
         Tx Power Threshold(dBm)          0.0      -16.0          -2.0        -14.0
         LBC Threshold(mA)               0.00       0.00          0.00         0.00
         Temp. Threshold(celsius)       80.00      -5.00         75.00         0.00
         Voltage Threshold(volt)         3.46       3.13          3.43         3.16
         LBC High Threshold = 98 %
         Configured Tx Power = -10.00 dBm
         Configured CD High Threshold = -5000 ps/nm
         Configured CD lower Threshold = -5000 ps/nm
         Configured OSNR lower Threshold = 9.00 dB
         Configured DGD Higher Threshold = 80.00 ps
         Baud Rate =  60.1385459900 GBd
         Modulation Type: 16QAM
         Chromatic Dispersion 0 ps/nm
         Configured CD-MIN -26000 ps/nm  CD-MAX 26000 ps/nm
         Second Order Polarization Mode Dispersion = 0.00 ps^2
         Optical Signal to Noise Ratio = 0.00 dB
         Polarization Dependent Loss = 0.00 dB
```

```
        Polarization Change Rate = 0.00 rad/s
        Differential Group Delay = 0.00 ps
        Temperature = 21.00 Celsius
        Voltage = 3.42 V
 Transceiver Vendor Details
        Form Factor            : QSFP-DD
        Optics type            : QSFPDD 400G ZRP
        Name                   : CISCO-ACACIA
        OUI Number             : 7c.b2.5c
        Part Number            : DP04QSDD-E30-19E
        Rev Number             : 10
        Serial Number          : ACA244900GN
        PID                    : QDD-400G-ZRP-S
        VID                    : ES03
        Firmware Version       : 161.06
        Date Code(yy/mm/dd)    : 20/12/08
!
```

Configuring Chromatic Dispersion

You can configure chromatic dispersion on optics controllers. When you configure the maximum and minimum values for chromatic dispersion for any data rate, ensure that the minimum difference between the configured values is equal to or greater than 1000 ps/nm.

The following table lists the default CD search range:

Table 52: Default CD Search Range

Muxponder Rate	FEC Value	Default CD Search Range (Min-Max)
400	OFEC	-26000 to +26000
400	CFEC	-2400 to +2400
300	OFEC	-50000 to +50000
200	OFEC	-50000 to +50000
100	OFEC	-80000 to +80000

Chromatic Dispersion Configuration Example

This example shows how to configure chromatic dispersion on the optics controller:

```
Router#configure
Router(config)#controller optics 0/0/0/13
Router(config-Optics)#cd-max 4000
Router(config-Optics)#cd-min -4000
Router(config-Optics)#commit
Router(config-Optics)#exit
Router(config)#exit
```

Running Configuration

This example shows the running configuration for the optics controller:

```
Router#show run controller optics 0/0/0/13
Thu May 13 12:24:42.353 UTC
```

```
controller Optics0/0/0/13
 cd-min -4000
 cd-max 4000
 !
```

Verification

This example shows how to verify the configured chromatic dispersion values for the optics controller:

```
Router#show controller optics 0/0/0/13
 Controller State: Up
 Transport Admin State: In Service
 Laser State: On
 LED State: Green
 FEC State: FEC ENABLED
 Optics Status
         Optics Type:  QSFPDD 400G ZR
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm
         Alarm Status:
         -------------
         Detected Alarms: None
         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 0
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 35
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = -7.87 dBm
         RX Power = -8.27 dBm
         RX Signal Power = -8.43 dBm
         Frequency Offset = 130 MHz
         Performance Monitoring: Enable
         THRESHOLD VALUES
         ----------------
         Parameter                High Alarm  Low Alarm  High Warning  Low Warning
         -----------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)          1.9      -28.2           0.0        -25.0
         Tx Power Threshold(dBm)          0.0      -15.0          -2.0        -16.0
         LBC Threshold(mA)               0.00       0.00          0.00         0.00
         Temp. Threshold(celsius)       80.00      -5.00         75.00        15.00
         Voltage Threshold(volt)         3.46       3.13          3.43         3.16
         LBC High Threshold = 98 %
         Configured Tx Power = -6.00 dBm
         Configured CD High Threshold = 80000 ps/nm
         Configured CD lower Threshold = -80000 ps/nm
         Configured OSNR lower Threshold = 9.00 dB
         Configured DGD Higher Threshold = 80.00 ps
         Baud Rate =  59.8437500000 GBd
         Modulation Type: 16QAM
         Chromatic Dispersion 0 ps/nm
         Configured CD-MIN -4000 ps/nm  CD-MAX 4000 ps/nm
         Second Order Polarization Mode Dispersion = 5.00 ps^2
         Optical Signal to Noise Ratio = 36.30 dB
         Polarization Dependent Loss = 0.40 dB
         Polarization Change Rate = 0.00 rad/s
         Differential Group Delay = 4.00 ps
         Temperature = 54.00 Celsius
         Voltage = 3.37 V
 Transceiver Vendor Details
```

```
Form Factor              : QSFP-DD
Optics type              : QSFPDD 400G ZR
Name                     : CISCO-ACACIA
OUI Number               : 7c.b2.5c
Part Number              : DP04QSDD-E20-19E
Rev Number               : 10
Serial Number            : ACA2447003L
PID                      : QDD-400G-ZR-S
VID                      : ES03
Firmware Version         : 61.12
Date Code(yy/mm/dd)      : 20/12/02
```

Configuring Optical Transmit Power

You can set the transmit power of the optical signal.

Each QDD-400G-ZR-S and QDD-400G-ZRP-S optical module has its own optical transmit (TX) power range. You can change the optical transmit (TX) power value based on the module capability. For "Transmitter specifications", see the *Cisco 400G Digital Coherent Optics QSFP-DD Optical Modules Data Sheet*.

Table 53: Optical Transmit Power Values

Optical Module	Trunk Speed[1,3]	Optical Transmit Power (Tx) Shaping	Interval	Supported Range of Optical Transmit Power (Tx) Values (in units of 0.1dBm)[2]		
				Minimum Value	Maximum Value - Typical	Maximum Value - Worst Case
QDD-400G-ZR-S	400G	No	1	-150	-100	-100
QDD-400G-ZRP-S	400G	Yes	1	-150	-110	-130
	300G			-150	-104	-119
	200G			-150	-90	-105
	100G			-150	-59	-75
DP04QSDD-HE0	400G	Yes	1	-100	20	10
	300G					
	200G					
	100G					

1. Release 7.3.15 supports 4x100G muxponder mode or trunk speed.

2. The default optical transmit power (Tx) value is -10 dBm, however with Tx shaping enabled the maximum power in 1x400G, 4x100G, 3x100G, 2x100G, and 1x100G modes may be less than -10 dBm.

3. Release 7.3.2 and future releases support 3x100G, 2x100G, and 1x100G muxponder modes or trunk speed.

Transmitting Power Configuration Example

The following example shows how to configure the optical transmit (TX) power on the optics controller:

```
Router#config
Router(config)#controller optics 0/2/0/16
Router(config-Optics)#transmit-power -125
Router(config-Optics)#commit
Router(config-Optics)#exit
Router(config)#exit
```

Running Configuration

This example shows the running configuration for the optics controller:

```
Router#show run controller optics 0/2/0/16
Thu May 13 12:52:35.020 UTC
controller Optics0/0/0/1
 cd-min -4000
 cd-max 4000
 transmit-power -125
!
```

Verification

This example shows how to verify the configured optical transmit power for the optics controller:

```
Router#show controller optics 0/2/0/16
Fri May 28 02:52:06.182 UTC
 Controller State: Up
 Transport Admin State: In Service
 Laser State: Off
 LED State: Off
 FEC State: FEC ENABLED
 Optics Status
         Optics Type:  QSFPDD 400G ZRP
         DWDM carrier Info: C BAND, MSA ITU Channel=80, Frequency=192.15THz,
         Wavelength=1560.200nm
         Alarm Status:
         -------------
         Detected Alarms: None
         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 0
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 mA
         Actual TX Power = -40.00 dBm
         RX Power = -40.00 dBm
         RX Signal Power = -40.00 dBm
         Frequency Offset = 0 MHz
         Laser Temperature = 0.00 Celsius
         Laser Age = 0 %
         DAC Rate = 1x1.25
         Performance Monitoring: Enable
         THRESHOLD VALUES
         ----------------
         Parameter                High Alarm  Low Alarm  High Warning  Low Warning
         -----------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         13.0      -24.0          10.0        -22.0
         Tx Power Threshold(dBm)          0.0      -16.0          -2.0        -14.0
         LBC Threshold(mA)               0.00       0.00          0.00         0.00
         Temp. Threshold(celsius)       80.00      -5.00         75.00         0.00
         Voltage Threshold(volt)         3.46       3.13          3.43         3.16
```

```
        LBC High Threshold = 98 %
        Configured Tx Power = -12.50 dBm
        Configured CD High Threshold = -5000 ps/nm
        Configured CD lower Threshold = -5000 ps/nm
        Configured OSNR lower Threshold = 9.00 dB
        Configured DGD Higher Threshold = 80.00 ps
        Baud Rate =  60.1385459900 GBd
        Modulation Type: 16QAM
        Chromatic Dispersion 0 ps/nm
        Configured CD-MIN -4000 ps/nm  CD-MAX 4000 ps/nm
        Second Order Polarization Mode Dispersion = 0.00 ps^2
        Optical Signal to Noise Ratio = 0.00 dB
        Polarization Dependent Loss = 0.00 dB
        Polarization Change Rate = 0.00 rad/s
        Differential Group Delay = 0.00 ps
        Temperature = 20.00 Celsius
        Voltage = 3.41 V
Transceiver Vendor Details
        Form Factor             : QSFP-DD
        Optics type             : QSFPDD 400G ZRP
        Name                    : CISCO-ACACIA
        OUI Number              : 7c.b2.5c
        Part Number             : DP04QSDD-E30-19E
        Rev Number              : 10
        Serial Number           : ACA244900GN
        PID                     : QDD-400G-ZRP-S
        VID                     : ES03
        Firmware Version        : 161.06
        Date Code(yy/mm/dd)     : 20/12/08
```

Configuring Muxponder Mode

By default, the Cisco IOS XR software configures the QDD-400G-ZR-S and QDD-400G-ZRP-S optical modules in the 400G transponder mode.

Using the **breakout** *muxponder mode* command, you can configure muxponder mode on optics controllers. Based on the muxponder mode, you can choose the modulation.

Muxponder mode options available for QDD-400G-ZR-S are:

- 4x100

Muxponder mode options available for QDD-400G-ZRP-S are:

- 4x100
- 3x100
- 2x100

Note Release 7.3.15 supports only 4x100 muxponder mode.

See the following tables for the modulation values, based on the muxponder mode:

- QDD-400G-ZR-S Transponder and Muxponder Configuration Values, on page 270
- QDD-400G-ZRP-S Transponder and Muxponder Configuration Values, on page 270

Using the **no breakout** *muxponder mode* command, you can switch from the muxponder mode to the transponder mode, on optics controllers.

Muxponder Mode Configuration Example

The following example shows how to configure muxponder mode on the optics controller:

```
Router#config
Router(config)#controller optics 0/0/0/13
Router(config-Optics)#breakout 4x100
Router(config-Optics)#commit
Router(config-Optics)#exit
Router(config)#exit
```

Note In the above example, the Cisco IOS XR software creates four Ethernet clients with 100GE speed, which can be verified using the **show interfaces brief | include** *R/S/I/P* command.

Running Configuration

This example shows the running configuration for the optics controller:

```
Router#show run controller optics 0/0/0/13
Thu May 13 12:24:42.353 UTC
controller Optics0/0/0/13
 cd-min -4000
 cd-max 4000
 breakout 4x100
 !
```

Verification

This example shows how to verify the muxponder mode configuration:

```
Router#show interfaces brief | include 0/0/0/13
 Hu0/0/0/13/0           up          up               ARPA  1514  100000000
 Hu0/0/0/13/1           up          up               ARPA  1514  100000000
 Hu0/0/0/13/2           up          up               ARPA  1514  100000000
 Hu0/0/0/13/3           up          up               ARPA  1514  100000000
```

Transponder Mode Configuration Example

The following example shows how to switch to the transponder mode, on the optics controller:

```
Router#config
Router(config)#controller optics 0/0/0/13
Router(config-Optics)#no breakout 4x100
Router(config-Optics)#commit
Router(config-Optics)#exit
Router(config)#exit
```

Note The Cisco IOS XR software creates a single 400GE interface, which can be verified using the **show interfaces brief | include** *R/S/I/P* command.

Running Configuration

This example shows the running configuration for the optics controller. The breakout configuration is absent in the running configuration.

```
Router#show run controller optics 0/0/0/13
Thu May 13 13:51:20.330 UTC
controller Optics0/0/0/13
 cd-min -4000
 cd-max 4000
 transmit-power -100
 !
```

Verification

This example shows how to verify the transponder mode configuration:

```
Router#show interfaces brief | include 0/0/0/13
FH0/0/0/13          up          up               ARPA  1514  400000000
```

Configure 100G operating modes with 200G DAC

The configuration support for 100G operating modes feature allows you to manually configure the speed of port as100G modes with 200G QSFP56 DAC cables.

Table 54: Feature History Table

Feature Name	Release Information	Feature Description
Configure 100G operating modes with 200G and 4x100 DAC	Release 25.3.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200](select variants only*), Modular Systems (8800 [LC ASIC: Q200]) (select variants only*) The feature supports 100G operating modes with 200G QSFP56 DAC, allowing the users to configure multi-rate optics and passive copper cables to operate at various speeds and lane combinations. This addresses the need for flexible speed configuration, particularly for connecting to custom servers that support specific speed and lane modes, and to prevent alarms when optics with different speeds are inserted. The feature introduces these changes: **CLI:** The **speed** keyword is included along with the 100G *[host-lanes < 4 | 2 >]* option in the controller optics command.

The support for configuring 100G operating modes with 200G DAC feature allows you to manually configure the speed of the port as 100G when using 200G DAC modules. This feature provides a CLI command to explicitly set the speed configuration to 100G operating modes and optionally specify the number of host lanes. The CLI command is implemented under the existing controller optics command which allows users to configure the speed of a port and optionally specify the number of host lanes.

Configuring 100G operational modes with 200G and 4x100 DAC

Procedure

Step 1 Configure 100G operational modes with 200G and 4x100 DAC.

Example:

This example shows how to configure the speed of port as 100G with host lane valuse as 2. The supported host lanes for 100G speed are 2 and 4.

Example:

```
Router#configure
            Router(config)#controller optics 0/0/0/0
            Router(config-Optics)# speed 100g host-lanes 2
            Router(config-Optics)#commit
```

Step 2 Use the **show running-config controller optics** CLI command to verify the running configuration of the speed port.

Example:

```
Router#show running-config controller optics 0/0/0/0
Thu Aug 14 01:16:52.946 UTC
controller Optics0/0/0/0
 speed 100g host-lanes 2
```

Step 3 *Optional:* Use the **show configuration failed** CLI command to verify if the speed port configuration is failed.

Example:

This example shows the failure scenario, when the breakout is configured on the same port.

```
Router#show config failed
Tue Oct 29 13:07:55.478 UTC
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.

 Controller Optics0/0/0/0
 speed 100g host-lanes 2
!!% Breakout is configured on this port, please remove breakout configuration before apply port speed
 configuration
!
end
```

Once the CLI is verified, if the optics is present, and optics driver cannot configure the optics in such speed or host lanes, the given alarm is declared:

```
Router#:Oct 29 12:25:42.808 UTC: optics_driver[274]: %PKT_INFRA-FM-3-
FAULT_MAJOR : ALARM_MAJOR : MODULE AND SPEED CONFIG MISMATCH :DECLARE
:0/RP0/CPU0:  Optics0/0/0/18
```

If you remove the module, the alarm will be cleared. Similarly, when a new module is inserted, the same alarm is triggered if the module does not support the configured speed.

Example:

This example shows the failure scenario, when the unsupported host lanes are configured.

```
Router#:ios(config)#show config failed
Tue Oct 29 13:07:55.478 UTC
!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.


controller Optics0/0/0/0
 speed 100g host-lanes 3
!!% The list of supported host lanes for speed 100g is 2, 4
!
end
```

Configuring Modulation

You can configure modulation on optics controllers. Based on the muxponder mode, you can choose the modulation.

Note The system accepts any modulation value that is entered. However, if the modulation value is outside the supported range, it is not configured on the optical module. Instead, the optical module is auto-configured with a valid modulation value. To view this value, use the **show controller optics** *R/S/I/P* command.

See the following tables for the supported modulation values:

- QDD-400G-ZR-S Transponder and Muxponder Configuration Values, on page 270
- QDD-400G-ZRP-S Transponder and Muxponder Configuration Values, on page 270

Modulation Configuration Example

The following example shows how to configure modulation on the optics controller:

```
Router#config
Router(config)#controller optics 0/0/0/1
Router(config-Optics)#modulation 16Qam
Router(config-Optics)#commit
Router(config-Optics)#exit
Router(config)#exit
```

Running Configuration

This example shows the running configuration:

```
Router#show run controller optics 0/0/0/1
controller Optics0/0/0/1
 cd-min -4000
 cd-max 4000
 transmit-power -100
```

```
 modulation 16Qam
!
```

Note Use the **show controller optics** *R/S/I/P* command to verify the modulation value of the optical module.

Verification

This example shows how to verify the configured modulation value for the optics controller:

```
Router#show controller optics 0/0/0/1
 Controller State: Up
 Transport Admin State: In Service
 Laser State: On
 LED State: Green
 FEC State: FEC ENABLED
 Optics Status
         Optics Type:  QSFPDD 400G ZR
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm
         Alarm Status:
         -------------
         Detected Alarms: None
         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 0
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 35
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = -7.87 dBm
         RX Power = -8.27 dBm
         RX Signal Power = -8.43 dBm
         Frequency Offset = 130 MHz
         Performance Monitoring: Enable
         THRESHOLD VALUES
         ----------------
         Parameter                High Alarm  Low Alarm  High Warning  Low Warning
         -----------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         1.9      -28.2           0.0        -25.0
         Tx Power Threshold(dBm)         0.0      -15.0          -2.0        -16.0
         LBC Threshold(mA)              0.00       0.00          0.00         0.00
         Temp. Threshold(celsius)      80.00      -5.00         75.00        15.00
         Voltage Threshold(volt)        3.46       3.13          3.43         3.16
         LBC High Threshold = 98 %
         Configured Tx Power = -6.00 dBm
         Configured CD High Threshold = 80000 ps/nm
         Configured CD lower Threshold = -80000 ps/nm
         Configured OSNR lower Threshold = 9.00 dB
         Configured DGD Higher Threshold = 80.00 ps
         Baud Rate =  59.8437500000 GBd
         Modulation Type: 16QAM
         Chromatic Dispersion 0 ps/nm
         Configured CD-MIN -4000 ps/nm  CD-MAX 4000 ps/nm
         Second Order Polarization Mode Dispersion = 5.00 ps^2
         Optical Signal to Noise Ratio = 36.30 dB
         Polarization Dependent Loss = 0.40 dB
         Polarization Change Rate = 0.00 rad/s
```

```
        Differential Group Delay = 4.00 ps
        Temperature = 54.00 Celsius
        Voltage = 3.37 V
 Transceiver Vendor Details
        Form Factor             : QSFP-DD
        Optics type             : QSFPDD 400G ZR
        Name                    : CISCO-ACACIA
        OUI Number              : 7c.b2.5c
        Part Number             : DP04QSDD-E20-19E
        Rev Number              : 10
        Serial Number           : ACA2447003L
        PID                     : QDD-400G-ZR-S
        VID                     : ES03
        Firmware Version        : 61.12
        Date Code(yy/mm/dd)     : 20/12/02
```

Configuring DAC Rate

You can set the DAC (digital to analog conversion) sampling rate on optics controllers. You can modify the DAC sampling rate only on the QDD-400G-ZRP-S and DP04QSDD-HE optical module.

Note QDD-400G-ZR-S supports 1x1 dac-rate in cFEC mode. QDD-400G-ZRP-S and DP04QSDD-HE supports 1x1 dac-rate in cFEC mode and 1x1.25 dac-rate in oFEC mode.

DAC Rate Configuration Example

The following example shows how to set the DAC rate on the optics controller:

```
Router#config
Router(config)#controller optics 0/0/0/1
Router(config-Optics)#dac-rate 1x1
```

Verification

This example shows the running configuration:

```
Router#show run controller optics 0/0/0/1
Thu May 13 12:52:35.020 UTC
controller Optics0/0/0/1
 cd-min -4000
 cd-max 4000
 transmit-power -100
 modulation 16Qam
 DAC-Rate 1x1
!
!
```

Verification

This example shows how to verify the configured DAC rate for the optics controller:

```
Router#show controller optics 0/0/0/1
 Controller State: Up
 Transport Admin State: In Service
 Laser State: On
 LED State: Green
 FEC State: FEC ENABLED
 Optics Status
        Optics Type:  QSFPDD 400G ZR
```

```
        DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
        Wavelength=1552.524nm
        Alarm Status:
        -------------
        Detected Alarms: None
        LOS/LOL/Fault Status:
        Alarm Statistics:
        -------------
        HIGH-RX-PWR = 0            LOW-RX-PWR = 0
        HIGH-TX-PWR = 0            LOW-TX-PWR = 0
        HIGH-LBC = 0               HIGH-DGD = 0
        OOR-CD = 0                 OSNR = 35
        WVL-OOL = 0                MEA  = 0
        IMPROPER-REM = 0
        TX-POWER-PROV-MISMATCH = 0
        Laser Bias Current = 0.0 %
        Actual TX Power = -7.87 dBm
        RX Power = -8.27 dBm
        RX Signal Power = -8.43 dBm
        Frequency Offset = 130 MHz
        DAC Rate = 1x1
        Performance Monitoring: Enable
        THRESHOLD VALUES
        ----------------
        Parameter                High Alarm  Low Alarm  High Warning  Low Warning
        -----------------------  ----------  ---------  ------------  -----------
        Rx Power Threshold(dBm)         1.9      -28.2           0.0        -25.0
        Tx Power Threshold(dBm)         0.0      -15.0          -2.0        -16.0
        LBC Threshold(mA)              0.00       0.00          0.00         0.00
        Temp. Threshold(celsius)      80.00      -5.00         75.00        15.00
        Voltage Threshold(volt)        3.46       3.13          3.43         3.16
        LBC High Threshold = 98 %
        Configured Tx Power = -6.00 dBm
        Configured CD High Threshold = 80000 ps/nm
        Configured CD lower Threshold = -80000 ps/nm
        Configured OSNR lower Threshold = 9.00 dB
        Configured DGD Higher Threshold = 80.00 ps
        Baud Rate =  59.8437500000 GBd
        Modulation Type: 16QAM
        Chromatic Dispersion 0 ps/nm
        Configured CD-MIN -4000 ps/nm  CD-MAX 4000 ps/nm
        Second Order Polarization Mode Dispersion = 5.00 ps^2
        Optical Signal to Noise Ratio = 36.30 dB
        Polarization Dependent Loss = 0.40 dB
        Polarization Change Rate = 0.00 rad/s
        Differential Group Delay = 4.00 ps
        Temperature = 54.00 Celsius
        Voltage = 3.37 V
Transceiver Vendor Details
        Form Factor            : QSFP-DD
        Optics type            : QSFPDD 400G ZR
        Name                   : CISCO-ACACIA
        OUI Number             : 7c.b2.5c
        Part Number            : DP04QSDD-E20-19E
        Rev Number             : 10
        Serial Number          : ACA2447003L
        PID                    : QDD-400G-ZR-S
        VID                    : ES03
        Firmware Version       : 61.12
        Date Code(yy/mm/dd)    : 20/12/02
```

Configuring FEC

You can configure forward error correction (FEC) only on optics controllers. You can modify FEC only on the QDD-400G-ZRP-S and DP04QSDD-HE optical module. FEC is a feature that is used for controlling errors during data transmission. This feature works by adding data redundancy to the transmitted message using an algorithm. This redundancy allows the receiver to detect and correct a limited number of errors occurring anywhere in the message, instead of having to ask the transmitter to resend the message.

Note QDD-400G-ZR-S supports cFEC (concatenated forward error correction). QDD-400G-ZRP-S and DP04QSDD-HE supports cFEC and oFEC (open forward error correction).

FEC Configuration Example

The following sample shows how to configure FEC on the optics controller:

```
Router#configure
Router(config)#controller optics 0/0/0/13
Router(config-Optics)#fec CFEC
Router(config-Optics)#commit
Router(config-Optics)#exit
Router(config)#exit
```

Running Configuration

This example shows the running configuration:

```
Router#show controllers optics 0/0/0/13
controller Optics0/0/0/1
 cd-min -4000
 cd-max 4000
 transmit-power -100
 fec CFEC
 modulation 16Qam
 DAC-Rate 1x1.25
!
```

Verification

This example shows how to verify the FEC configuration for the optics controller:

```
Router#show controller coherentdsp 0/0/0/13
Thu May 27 17:28:51.960 UTC
Port                                        : CoherentDSP 0/0/0/13
Controller State                            : Down
Inherited Secondary State                   : Normal
Configured Secondary State                  : Maintenance
Derived State                               : Maintenance
Loopback mode                               : Internal
BER Thresholds                              : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                      : Enable
Bandwidth                                   : 400.0Gb/s

Alarm Information:
LOS = 6 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0     FLEXO_GIDM = 0
```

```
FLEXO-MM = 0    FLEXO-LOM = 0   FLEXO-RDI = 0
FLEXO-LOF = 5
Detected Alarms                                  : LOS
Bit Error Rate Information
PREFEC  BER                                      : 5.0E-01
POSTFEC BER                                      : 0.0E+00
Q-Factor                                         : 0.00 dB
Q-Margin                                         : -7.20dB
OTU TTI Received

FEC mode                                         : C_FEC
```

Configuring Loopback

You can configure media loopback and host loopback on optics controllers. Loopback can be performed only in the maintenance mode.

Note Line loopback mode is supported only on Cisco 8000 series line cards and fixed-port routers based on Q100 and Q200 silicon.

Loopback Configuration Example

This example shows how to enable loopback configuration on optics controllers.

Use `show controllers optics R/S/I/P information loopback` command to check the supported loopback types.

```
Router#show controllers optics 0/0/0/4 information loopback
Supported Loopback Types :
============================
 [1.] Media Line
 [2.] Host Internal

 Unsupported Loopback Types :
============================

 [1.] Media Internal
 [2.] Host Line
 [3.] Host Per Lane
 [4.] Media Per Lane
 [5.] Simultaneous Media Host
 Media Configured Loopback : Media Loopback None
 Media Applied Loopback    : Media Loopback None

 Host Configured Loopback : Host Loopback None
 Host Applied Loopback    : Host Loopback None
```

Use **loopback** and **host loopback** commands in config-optics sub mode to configure the media and host loopback modes respectively. Loopback mode for both media and host can be configured to either internal or line, depending on the supported loopback types.

```
Router#config
Router(config)#controller optics 0/0/0/4
Router(config-Optics)#sec-admin-state maintenance
Router(config-Optics)#loopback line /* configures the media loopback to line */
Router(config-Optics)#host loopback internal /* configures the host loopback to internal
*/
Router(config-Optics)#commit
```

Running Configuration

This example shows the running configuration on optics controllers.

```
Router#show run controller optics 0/0/0/4
Thu May 13 19:51:08.175 UTC
controller Optics0/0/0/4
 loopback line
 host loopback internal
 sec-admin-state maintenance
!
```

Verification

This example shows how to verify the loopback configuration on optics controllers.

```
Router#show controllers optics 0/0/0/4
Controller State: Up
Transport Admin State: In Service
Laser State: On
Host Squelch Status: Enable
Media linkdown preFEC degrade : Disabled
LED State: Yellow
FEC State: FEC ENABLED
Power Mode: High
Dom Data Status: Ready
Last link flapped: 00:02:32
Optics Status
     Optics Type:  QSFPDD 400G ZR
     DWDM carrier Info: C BAND, MSA ITU Channel=1, Frequency=196.10THz,
     Wavelength=1528.773nm
     Loopback Host : Internal
     Loopback Media : Line

     Alarm Status:
     -------------
     Detected Alarms: None

     LOS/LOL/Fault Status:
 ...
```

```
Router#show controllers optics 0/0/0/4 information loopback
Supported Loopback Types :
============================
 [1.] Media Line
 [2.] Host Internal

 Unsupported Loopback Types :
==============================

 [1.] Media Internal
 [2.] Host Line
 [3.] Host Per Lane
 [4.] Media Per Lane
 [5.] Simultaneous Media Host
 Media Configured Loopback : Media Loopback Line
 Media Applied Loopback    : Media Loopback Line

 Host Configured Loopback : Host Loopback Internal
 Host Applied Loopback    : Host Loopback Internal
```

Disable Auto-Squelching

Table 55: Feature History Table

Feature Name	Release Information	Description
Disable Auto-Squelching	Release 7.11.1	This release introduces support to disable Auto squelching. This helps to detect weak signals that are hidden within the laser source noise. By disabling Auto squelch, you can reduce the processing overhead in systems that have stable laser sources and minimal noise, helping you optimize the performance of your system. When the Auto squelch function is enabled, the optical module will generate a local fault signal on the host side if it detects a fault on the media side. By default, Auto squelch is enabled. The feature introduces these changes: **CLI:**The following keyword has been added. • **host auto-squelch disable** **YANG DATA models:** • New XPaths for `Cisco-IOS-XR-controller-optics-cfg` (see Github, YANG Data Models Navigator)

This release introduces the support to disable auto-squelch functionality on the module on the host side. When enabled, the squelch function is activated on the module when no suitable media-side input signal from the remote end is available to be forwarded to the host-side output (example: Rx LOS is asserted). Auto squelching is commonly used to suppress unwanted noise from laser sources in communication systems. When disabled and no valid signal is detected on the module from the remote end, the module will generate a local fault towards the NPU. However, disabling auto-squelching provides you with expanded signal detection. This enables you to detect extremely weak signals that are embedded within the laser source noise. Also, by eliminating the need to continuously monitor and suppress unwanted noise, system resources can be allocated more efficiently, leading to improved performance.

In this feature, we introduced the **host auto-squelch disable** command to disable the auto-squelch functionality when there is an invalid input signal from the remote end. This feature provides you with the flexibility to customize the system's behavior according to your requirements.

Disabling Laser Squelching Configuration Example

This example shows how to disable laser squelching for a host on controller optics:

```
router#config
router(config)#controller 0/0/0/0
router(config-Optics)#host auto-squelch disable
router(config-Optics)#commit
```

Verification

This example shows how to verify the laser squelching disabled configuration:

```
router#show controllers optics 0/0/0/0
    Host Squelch Status: disable
```

Configuring Performance Monitoring

Performance monitoring (PM) parameters are used by service providers to gather, store, set thresholds for, and report performance data for early detection of problems. The user can retrieve both current and historical PM counters for the various controllers in 30-second, 15-minute, and 24-hour intervals.

Performance monitoring can be configured on optics controllers and coherent DSP controllers.

To stop performance monitoring on optics or coherent DSP controllers, use the **perf-mon disable** keyword.

Configuring PM Parameters

The performance monitorning (PM) threshold and the threshold crossing alert (TCA) reporting status can be configured for optics controllers and coherent DSP controllers:

Table 56: PM Thresholds and TCA Report Status for Optics Controllers

PM Parameters	Description
CD	Sets the CD (chromatic dispersion) threshold or TCA reporting status.
DGD	Sets the DGD (differential group delay) threshold or TCA reporting status.
LBC	Sets the LBC (laser bias current) threshold or TCA reporting status in mA.
FREQ-OFF	Sets the FREQ-OFF (low signal frequency offset) threshold or TCA reporting status in Mhz.
OPR	Sets the OPR (optical power RX) threshold or TCA reporting status in uW or dbm.
OPT	Sets the OPT (optical power TX) threshold or TCA reporting status in uW or dbm.

PM Parameters	Description
OSNR	Sets the OSNR (optical signal-to-noise ratio) threshold or TCA reporting status.
PCR	Sets the PCR (polarization change rate) threshold or TCA reporting status.
PDL	Sets the PDL (polarization dependent loss) threshold or TCA reporting status.
RX-SIG	Sets the RX-SIG (receiving signal power) threshold or TCA reporting status in uW or dbm.
SNR	Sets the SNR (signal-to-noise ratio) threshold or TCA reporting status.
SOPMD	Sets the SOPMD (second order polarization mode dispersion) threshold or TCA reporting status.

Table 57: PM Thresholds TCA Report Status for Coherent DSP Controllers

PM Parameters	Description
Q	Sets the Q threshold or TCA reporting status.
Q-margin	Sets the Q margin threshold or TCA reporting status.
EC-BITS	Sets the EC-BITS (error corrected bits) threshold or TCA reporting status.
PostFEC BER	Sets the post-FEC BER threshold or TCA reporting status.
PreFEC BER	Sets the pre-FEC BER threshold or TCA reporting status.
UC-WORDS	Sets the UC-WORDS (uncorrected words) threshold or TCA reporting status.
Host-Intf-0-FEC-BER	Sets the Host-Intf-0-FEC-BER threshold or TCA reporting status, where: • AVG - specifies the number of corrected bits received from the host interface prior to a PM interval. • MIN - specifies the minimum number of corrected bits received from the host interface over a sub-interval and prior to a PM interval. • MAX - specifies the maximum number of corrected bits received from the host interface over a sub-interval and prior to a PM interval.

PM Parameters	Description
Host-Intf-0-FEC-FERC	Sets the Host-Intf-0-FEC-FERC threshold or TCA reporting status, where: • AVG - specifies the number of frames received from the host interface during a sub-interval. • MIN - specifies the minimum number of frames received from the host interface with uncorrected errors over a sub-interval and prior to a PM interval. • MAX - specifies the maximum number of frames received from the host interface with uncorrected errors over a sub-interval and prior to a PM interval.

Performance Monitoring Configuration Example

This example shows how to enable performance monitoring and set PM thresholds on the optics controller:

```
Router#config
Router(config)#controller optics 0/2/0/16
Router(config-Optics)#perf-mon enable
Router(config-Optics)#pm 30-sec optics threshold cd max 100
Router(config-Optics)#pm 30-sec optics threshold cd min -100
Router(config-Optics)#commit
```

Running Configuration

This example shows the running configuration on optics controllers:

```
Router#show run controller optics 0/2/0/16
Thu May 13 20:18:55.957 UTC
controller Optics0/2/0/16
pm 30-sec optics threshold cd max 100
pm 30-sec optics threshold cd min -100
 perf-mon enable
!
```

Verification

This example shows how to verify the PM parameters on optics controllers. Verify the configuration changes in the Configured Threshold fields:

```
Router#show controller optics 0/2/0/16 pm current 30-sec optics 1
  Thu May 27 17:58:49.889 UTC
Optics in the current interval [17:58:30 - 17:58:49 Thu May 27 2021]
Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
      Configured     TCA
                                          Threshold(min)   Threshold(min) (min) Threshold(max)
   Threshold(max) (max)
LBC[mA ]       : 0.0       0.0       0.0      0.0               NA              NO   100.0
           NA              NO
OPT[dBm]      : -9.98     -9.98     -9.98    -15.09            NA              NO   0.00
          NA              NO
OPR[dBm]      : -40.00    -40.00    -40.00   -30.00            NA              NO   8.00
          NA              NO
CD[ps/nm]     : 0         0         0        -80000          -100          NO   100
```

```
    100          NO
DGD[ps ]      : 0.00       0.00       0.00      0.00              NA               NO   80.00
          NA                NO
SOPMD[ps^2]   : 0.00       0.00       0.00      0.00              NA               NO   2000.00
          NA                NO
OSNR[dB]      : 0.00       0.00       0.00      0.00              NA               NO   40.00
          NA                NO
PDL[dB]       : 0.00       0.00       0.00      0.00              NA               NO   7.00
          NA                NO
PCR[rad/s]    : 0.00      0.00      0.00     0.00             NA              NO   2500000.00
       NA               NO
RX_SIG[dBm]   : -40.00    -40.00    -40.00   -30.00            NA               NO   1.00
          NA                NO
FREQ_OFF[Mhz]: 0          0         0        -3600             NA               NO   3600
          NA                NO
SNR[dB]       : 0.00       0.00       0.00      7.00              NA               NO   100.00
          NA                NO

Last clearing of "show controllers OPTICS" counters never
!
```

Performance Monitoring Configuration Example

This example shows how to enable performance monitoring and set PM thresholds and TCA reporting status on the coherent DSP controller:

```
Router#config
Router(config)#controller CoherentDSP0/2/0/16
Router(config-CoDSP)#perf-mon enable
Router(config-CoDSP)#pm 30-sec fec report Q max-tca enable
Router(config-CoDSP)#pm 30-sec fec report Q-margin max-tca enable
Router(config-CoDSP)#pm 30-sec fec report Q min-tca enable
Router(config-CoDSP)#pm 30-sec fec report Q-margin min-tca enable
Router(config-CoDSP)#pm 30-sec fec threshold Q max 1200
Router(config-CoDSP)#pm 30-sec fec threshold Q-margin max 500
Router(config-CoDSP)#pm 30-sec fec threshold Q min 900
Router(config-CoDSP)#pm 30-sec fec threshold Q-margin min 280
Router(config-CoDSP)#commit
```

Running Configuration

This example shows the running configuration on coherent DSP controllers:

```
Router#show run controller coherentdsp 0/2/0/16
Thu May 13 19:56:09.136 UTC
controller CoherentDSP0/2/0/16
 pm 30-sec fec report Q max-tca enable
 pm 30-sec fec report Q-margin max-tca enable
 pm 30-sec fec report Q min-tca enable
 pm 30-sec fec report Q-margin min-tca enable
 pm 30-sec fec threshold Q max 1200
 pm 30-sec fec threshold Q-margin max 500
 pm 30-sec fec threshold Q min 900
 pm 30-sec fec threshold Q-margin min 280
 perf-mon enable
 !
```

Verification

This example shows how to verify the PM parameters on coherent DSP controllers. Verify the configuration changes in the highlighted fields:

```
Router#show controllers coherentdsp 0/2/0/16 pm current 30-sec fec
Thu May 27 23:04:54.167 UTC
g709 FEC in the current interval [23:04:30 - 23:04:54 Thu May 27 2021]
```

```
FEC current bucket type : Valid
    EC-BITS   : 0                       Threshold : 111484000000          TCA(enable)  :
YES
    UC-WORDS  : 0                       Threshold : 5                     TCA(enable)  :
YES
                                      MIN       AVG        MAX      Threshold      TCA
 Threshold     TCA
                                                                        (min)     (enable)
   (max)     (enable)
PreFEC BER                    :     0E-15     0E-15     0E-15      0E-15        NO
0E-15        NO
PostFEC BER                   :     0E-15     0E-15     0E-15      0E-15        NO
0E-15        NO
Q[dB]                         :      0.00      0.00      0.00      9.00  YES 120.00  YES
Q_Margin[dB]                  :      0.00      0.00      0.00      2.80  YES  5.00   YES
 !
```

Configuring Alarms Threshold

The alarms threshold can be configured for monitoring alarms on optics controllers:

Table 58: Alarms Threshold Parameters for Optics Controllers

Alarm Threshold Parameters	Description
CD	Sets the CD (chromatic dispersion) alarm threshold (cd-low-threshold and cd-high-threshold).
DGD	Sets the DGD (differential group delay) alarm threshold.
LBC	Sets the LBC (laser bias current) threshold in mA.
OSNR	Sets the OSNR (optical signal-to-noise ratio) alarm threshold.

Alarm Threshold Configuration Example

This example shows how to configure alarm threshold on the optics controller:

```
Router#config
Router(config)#controller optics 0/2/0/16
Router(config-Optics)#cd-low-threshold -2000
Router(config-Optics)#cd-high-threshold 2000
Router(config-Optics)#commit
```

Running Configuration

This example shows the running configuration on the optics controller:

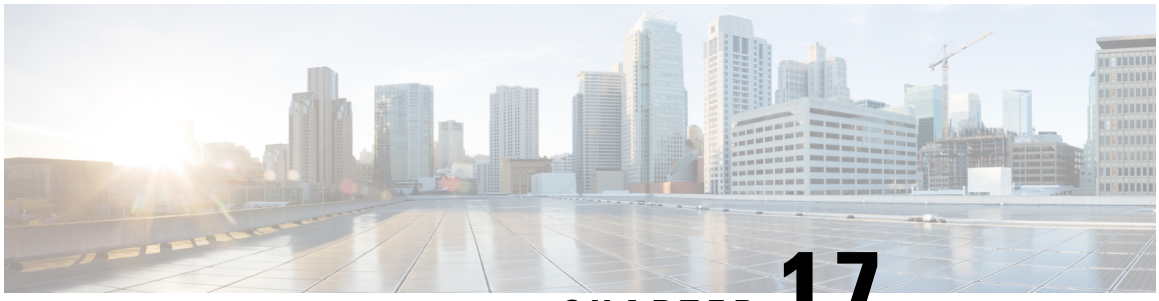
```
Router#show run controller optics 0/2/0/16
Thu May 13 20:18:55.957 UTC
controller Optics0/2/0/16
 cd-low-threshold 2000
 cd-high-threshold 2000
!
```

Verification

This example shows how to verify the alarm threshold on optics controllers:

```
Router#show controller optics 0/2/0/16
Fri May 28 01:04:33.604 UTC
 Controller State: Up
 Transport Admin State: In Service
 Laser State: Off
 LED State: Off
 FEC State: FEC ENABLED
 Optics Status
         Optics Type:  QSFPDD 400G ZRP
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm
         Alarm Status:
         -------------
         Detected Alarms: None
         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 0
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 mA
         Actual TX Power = -40.00 dBm
         RX Power = -40.00 dBm
         RX Signal Power = -40.00 dBm
         Frequency Offset = 0 MHz
         Laser Temperature = 0.00 Celsius
         Laser Age = 0 %
         DAC Rate = 1x1.25
         Performance Monitoring: Enable
         THRESHOLD VALUES
         ----------------
         Parameter                High Alarm  Low Alarm  High Warning  Low Warning
         -----------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         13.0      -24.0          10.0        -22.0
         Tx Power Threshold(dBm)          0.0      -16.0          -2.0        -14.0
         LBC Threshold(mA)               0.00       0.00          0.00         0.00
         Temp. Threshold(celsius)       80.00      -5.00         75.00         0.00
         Voltage Threshold(volt)         3.46       3.13          3.43         3.16
         LBC High Threshold = 98 %
         Configured Tx Power = -10.00 dBm
         Configured CD High Threshold = -5000 ps/nm
         Configured CD lower Threshold = -5000 ps/nm
         Configured OSNR lower Threshold = 9.00 dB
         Configured DGD Higher Threshold = 80.00 ps
         Baud Rate =  60.1385459900 GBd
         Modulation Type: 16QAM
         Chromatic Dispersion 0 ps/nm
         Configured CD-MIN -26000 ps/nm  CD-MAX 26000 ps/nm
         Second Order Polarization Mode Dispersion = 0.00 ps^2
         Optical Signal to Noise Ratio = 0.00 dB
         Polarization Dependent Loss = 0.00 dB
         Polarization Change Rate = 0.00 rad/s
         Differential Group Delay = 0.00 ps
         Temperature = 21.00 Celsius
         Voltage = 3.42 V
 Transceiver Vendor Details
         Form Factor            : QSFP-DD
         Optics type            : QSFPDD 400G ZRP
```

```
            Name                    : CISCO-ACACIA
            OUI Number              : 7c.b2.5c
            Part Number             : DP04QSDD-E30-19E
            Rev Number              : 10
            Serial Number           : ACA244900GN
            PID                     : QDD-400G-ZRP-S
            VID                     : ES03
            Firmware Version        : 161.06
            Date Code(yy/mm/dd)     : 20/12/08
!
```



CHAPTER 17

Configuring Controllers

This chapter describes the Optics Controller for the 36-port QSFP56-DD 400 GbE and 48-port QSFP28 100 GbE Line Cards. This chapter also describes the procedures used to configure the controllers.

Note When two MACsec enabled Cisco 8000 Series Routers with Coherent Line Cards are connected, there is no compatibility between Coherent Line Cards of IOS XR Release.

- breakout - Configure breakout mode (‘breakout 4x10’ only.)
- clear - Clear the uncommitted configuration.
- commit - Commit the configuration changes to running.
- do - Run an exec command.
- end - Exit from configure mode.
- exit - Exit from this submode.
- ext-description - Set ext-description for this controller.
- no - Negate a command or set its defaults.
- pwd - Commands used to reach current submode.
- root - Exit to the global configuration mode.
- show - Show contents of configuration.

Following controller configuration options are supported on the router:

How to Configure Controllers

This section contains the following procedures:

Configuring Optics Controller

Configuring optics controller of breakout 4x10:

```
RP/0/RP0/CPU0:uut#configure
Fri Oct 11 16:22:31.222 UTC
RP/0/RP0/CPU0:uut(config)#controller optics 0/1/0/28
RP/0/RP0/CPU0:uut(config-Optics)#breakout 4x10
RP/0/RP0/CPU0:uut(config-Optics)#commit
Fri Oct 11 16:23:26.868 UTC
RP/0/RP0/CPU0:uut(config-Optics)#end
RP/0/RP0/CPU0:uut#
RP/0/RP0/CPU0:uut#show running-config controller optics 0/1/0/28
Fri Oct 11 16:23:41.273 UTC
controller Optics0/1/0/28
breakout 4x10
!
```

Disabling Optical Modules

This feature provides the ability to disable and re-enable an optical module through CLI, which simulates online insertion and removal (OIR) by disabling power to the transceiver port.

Typical troubleshooting procedures for optical modules can include performing OIR by removing and re-installing the module, which requires onsite personnel to physically reseat the optical module. The ability to remotely disable and enable an optical module can significantly reduce operational expenses.

Example

The following output shows a QSFP28 module powered on and in UP state:

```
Router# show controllers optics 0/0/0/0

Controller State: Up

Transport Admin State: In Service

Laser State: Off

LED State: Not Applicable

FEC State: FEC ENABLED

 Optics Status

         Optics Type:  QSFP28 100G FR
         Wavelength = 1311.00 nm

         Alarm Status:
         -------------
         Detected Alarms: None
```

```
        LOS/LOL/Fault Status:

        Laser Bias Current = 26.2 mA
        Actual TX Power = 0.73 dBm
        RX Power = -0.68 dBm

        Performance Monitoring: Disable

        THRESHOLD VALUES
        ----------------

        Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
        ------------------------  ----------  ---------  ------------  -----------
        Rx Power Threshold(dBm)          7.4      -10.4           4.5         -6.3
        Tx Power Threshold(dBm)          7.0       -6.3           4.0         -2.4
        LBC Threshold(mA)             100.00       8.00         83.00        10.00
        Temp. Threshold(celsius)       75.00      -5.00         70.00         0.00
        Voltage Threshold(volt)         3.63       2.97          3.46         3.13

        Polarization parameters not supported by optics

        Temperature = 27.92 Celsius
        Voltage = 3.24 V

Transceiver Vendor Details

        Form Factor            : QSFP28
        Optics type            : QSFP28 100G FR
        Name                   : CISCO-CISCO
        OUI Number             : 00.00.0c
        Part Number            : 10-3248-01
        Rev Number             : 01
        Serial Number          : FBN2331A114
        PID                    : QSFP-100G-FR-S
        VID                    : ES0
        Date Code(yy/mm/dd)    : 19/09/19
```

To disable the module, use the **transceiver disable** command in controller optics configuration mode:

```
Router(config)# controller optics 0/0/0/0
Router(config-Optics)# transceiver disable
Router(config-Optics)# commit
Router(config-Optics)# end
```

The following example shows the QSFP28 module disabled and powered down:

```
Router# show controllers optics 0/0/0/0

Controller State: Down

Transport Admin State: In Service

Laser State: Off

 Optics Status

        Optics Type:  Unknown optics
        Wavelength = 0.00 nm

        Alarm Status:
        -------------
        Detected Alarms: None
```

```
        LOS/LOL/Fault Status:

        TX Power = N/A
        RX Power = N/A

        Performance Monitoring: Disable

        THRESHOLD VALUES
        ----------------

        Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
        ------------------------  ----------  ---------  ------------  -----------
        Rx Power Threshold(dBm)          7.4      -10.4           4.5         -6.3
        Tx Power Threshold(dBm)          7.0       -6.3           4.0         -2.4
        LBC Threshold(mA)             100.00       8.00         83.00        10.00
        Temp. Threshold(celsius)       75.00      -5.00         70.00         0.00
        Voltage Threshold(volt)         3.63       2.97          3.46         3.13

        Polarization parameters not supported by optics

        Temperature = 0.00 Celsius
        Voltage = 0.00 V

Transceiver Vendor Details
```

To re-enable the module, use the **no transceiver disable** command in controller optics configuration mode.

Diagnostic Parameters for Optical Transceivers

Table 59: Feature History Table

Feature Name	Release Information	Description
Diagnostic Parameters for Optical Transceivers	Release 7.11.1	

Feature Name	Release Information	Description
		You can analyze the diagnostic parameters for optical transceivers installed on a network device and detect potential issues with the optical transceivers, such as excessive power levels, abnormal temperature readings, or degradation of the optical signal. Such analysis is possible because the **show controllers optics** command now displays the following diagnostic parameters: • Effective Signal to Noise Ratio (eSNR) • Pulse Amplitude Modulation with Four Levels (PAM4) Level Transition Parameter (LTP) • Pre-Forward Error Correction (FEC) and Post-FEC Bit Error Rate (BER) • Frame Error Count (FERC) • Laser age • Thermoelectric Cooler (TEC) current • Laser frequency • Laser temperature For additional information on VDM (Versatile Diagnostics Monitoring), see the Common Management Interface Specification. The feature introduces these changes: **CLI:** • The **observable-info** keyword is added to the show controller optics command. **YANG Data Model:** • New XPath for `Cisco-IOS-XR-controller-optics-oper.yang`

Feature Name	Release Information	Description
		(see GitHub, YANG Data Models Navigator)

In order to monitor and report the performance of an optical transceiver and thereby enhancing the troubleshooting capabilities of the optical transceiver, the **observable-info** keyword is added to the **show controllers optics** command to display the diagnostics parameters. These parameters help in monitoring the health of the network when the optical transceiver heats up, when the link is down, when alarms are raised, or when there's traffic loss in the network. This improvement in the **show controllers optics** command now displays the following diagnostic parameters:

- Effective Signal to Noise Ratio (eSNR)
- Pulse Amplitude Modulation with Four Levels (PAM4) Level Transition Parameter (LTP)
- Pre-Forward Error Correction (FEC) and Post-FEC Bit Error Rate (BER)
- Frame Error Count (FERC)
- Laser age
- Thermoelectric Cooler (TEC) current
- Laser frequency
- Laser temperature

Note Not all optical transceivers support the **observable-info** keyword. Also, the parameters that are displayed depend on what the optical transceiver supports, that is, not all optical transceivers display the same parameters. For additional information on VDM (Versatile Diagnostics Monitoring), see the Common Management Interface Specification.

Verification

The following **show controllers optics observable-info** command displays the monitoring parameters of the optical transceiver present in the 0/0/0/9 location ID. The 0/0/0/9 location ID represents rack/slot/instance/port. Based on the requirement, the network administrators can use the displayed values of this command for monitoring and troubleshooting.

```
Router#show controllers optics 0/0/0/9 observable-info
Observable Information

[eSNR Media Input]
 Unit: dB
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                     Low High  Low High
 Lane0    21.30               0.00                0.00                0.00
0.00                 n   n     n   n
 Lane1    22.05               0.00                0.00                0.00
0.00                 n   n     n   n
 Lane2    22.62               0.00                0.00                0.00
0.00                 n   n     n   n
 Lane3    22.05               0.00                0.00                0.00
```

```
0.00                n   n     n   n

[PAM4 Level Transition Parameter Media Input]
 Unit: dB
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Lane0    47.79               0.00                0.00                0.00
0.00                n   n     n   n
 Lane1    54.70               0.00                0.00                0.00
0.00                n   n     n   n
 Lane2    64.34               0.00                0.00                0.00
0.00                n   n     n   n
 Lane3    59.64               0.00                0.00                0.00
0.00                n   n     n   n

[Pre-FEC BER Minimum Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[Pre-FEC BER Minimum Host Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[Pre-FEC BER Maximum Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[Pre-FEC BER Maximum Host Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[Pre-FEC BER Average Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[Pre-FEC BER Average Host Input]
 Unit: n/a
```

```
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[Pre-FEC BER Current Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[Pre-FEC BER Current Host Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Minimum Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Minimum Host Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Maximum Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Maximum Host Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Average Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
```

```
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Average Host Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Current Media Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n

[FERC Current Host Input]
 Unit: n/a
 Id       Value               LowThreshWarn       HighThresWarn       LowThreshAlarm
HighThreshAlarm     TCAWarn   TCAAlarm

                    Low High  Low High
 Module   0.000E+00           0.000E+00           0.000E+00           0.000E+00
0.000E+00           n   n     n   n
```

Loopback on Optical Transceivers

Table 60: Feature History Table

Feature Name	Release Information	Description
Loopback on Optical Transceivers	Release 7.11.1	

Feature Name	Release Information	Description
		You can now easily detect link failures between the optical transceiver and an external device such as a router by creating a loopback within the transceiver itself. Enabling loopback detects the fault in the physical or network connections, such as, traffic loss or a faulty optical transceiver. The loopback configuration allows incoming traffic within the transceiver to be redirected back to its source. By analyzing the loopback signals received at the source, it becomes possible to detect physical connectivity failures or network issues, such as packet loss or a malfunctioning transceiver. The feature introduces these changes: **CLI:** Modified the controller optics command by adding the following keywords: • **host loopback internal** • **host loopback line** • **loopback internal** • **loopback line** The **information loopback** keyword is added to the show controller optics command. **YANG Data Model:** • New XPaths for `Cisco-IOS-XR-controller-optics-cfg.yang` (see GitHub, YANG Data Models Navigator)

You can now enable loopback functionality on the optical transceivers. Loopback is the process of redirecting inbound traffic or data signals from an optical transceiver back to the module itself. Re-routing traffic to its source enables utilization of the received data for diagnostic purposes, particularly in the identification and

resolution of physical connectivity issues or network-related problems, such as traffic loss or a faulty optical transceiver.

The optical transceiver is divided into two sides, the host side, which is positioned towards the router, and the media side, which is positioned towards the wire or cable media. It is possible to enable loopback on both the host side and media side of the optical transceiver.

Note Loopback can be performed only when the controller state is active (UP) and in the maintenance mode.

There are four types of loopback:

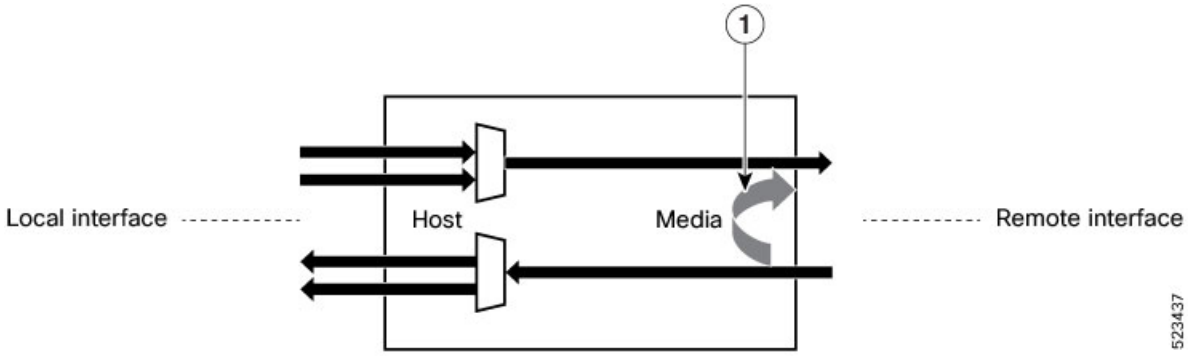
- Loopback Internal or Media Side Output Loopback
- Loopback Line or Media Side Input Loopback
- Host Loopback Internal or Host Side Input Loopback
- Host Loopback Line or Host Side Output Loopback

Note Configuring the internal loopback brings up the host interface and configuring the line loopback brings up the remote interface.

Media Side Input Loopback Configuration

In loopback line or media side input loopback, the signals received at the media side are looped back to the media side, indicating that the received data on the media is transmitted back to the media, that is, towards the remote interface. This is indicated by the arrow labeled as 1 in the illustration.

Figure 18: Media Side Input Loopback on the Optical Transceiver



Configuration Example

This example shows how to enable media side input loopback on the optical transceiver:

```
Router#config
Router(config)#controller optics 0/0/0/9
Router(config-Optics)#secondary-admin-state maintenance
Router(config-Optics)#loopback line
```

```
Loopback is a traffic-affecting operation
Router(config-Optics)#commit
Router(config-Optics)#end
```

Running Configuration

This example shows the running configuration of the media side input loopback on the optical transceiver:

```
Router#show run controller optics 0/0/0/9
controller Optics0/0/0/9
 secondary-admin-state maintenance
 loopback line
!
```

Verification

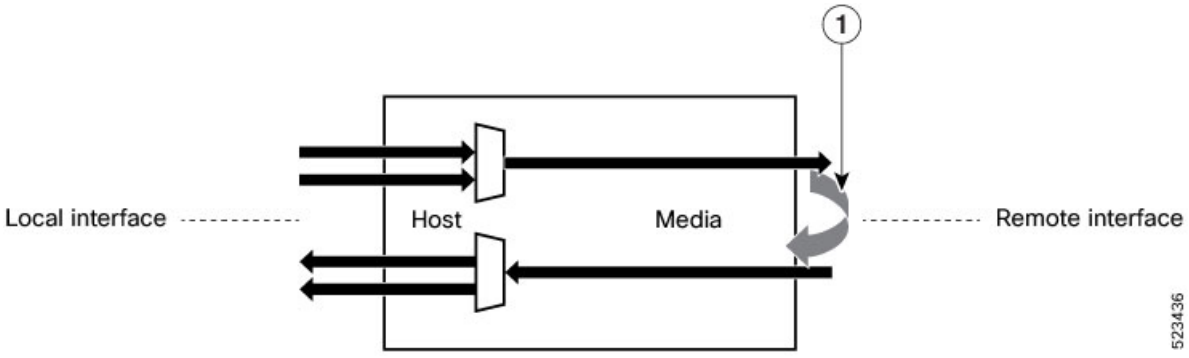
This example shows how to verify the media side input loopback configuration on the optical transceiver:

```
Router#show controller optics 0/0/0/9
Controller State: Up
Transport Admin State: Maintenance
Laser State: On
LED State: Green
FEC State: FEC ENABLED
Optics Status:

Optics Type: QSFPDD 400G FR4
Wavelength: 1301.00 nm
Loopback Host: None
Loopback Media: Line

Alarm Status:
----------------
Detected Alarms: None
LOS/LOL/Fault Status:
Performance Monitoring: Disable
```

Media Side Output Loopback

In loopback internal or media side output loopback, the loopback signal originating from the NPU is looped back to the same NPU on the media or line side, towards the remote interface. This is indicated by the arrow labeled as 1 in the illustration.

Figure 19: Media Side Output Loopback on the Optical Transceiver



Configuration Example

This example shows how to enable media side output loopback on the optical transceiver:

```
Router#config
Router(config)#controller optics 0/0/0/9
Router(config-Optics)#secondary-admin-state maintenance
Router(config-Optics)#loopback internal
Loopback is a traffic-affecting operation
Router(config-Optics)#commit
Router(config-Optics)#end
```

Running Configuration

This example shows the running configuration of the media side output loopback on the optical transceiver:

```
Router#show run controller optics 0/0/0/9
controller Optics0/0/0/9
 secondary-admin-state maintenance
 loopback internal
!
```

Verification

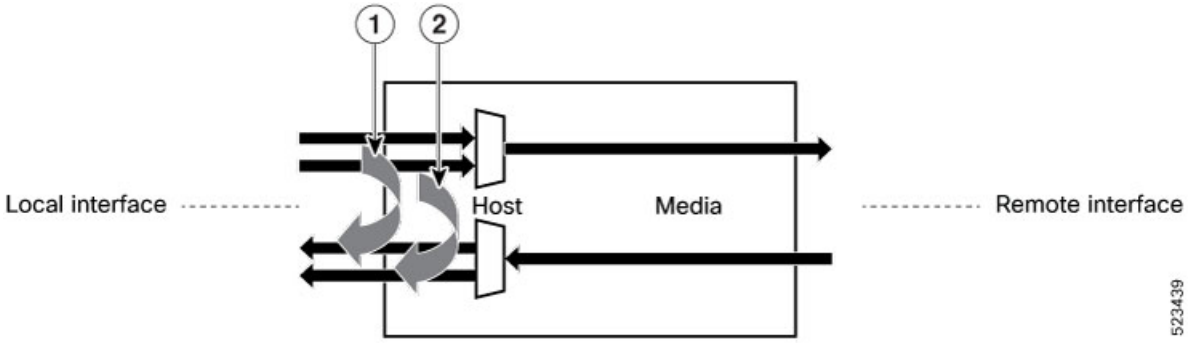
This example shows how to verify the media side output loopback configuration on the optical transceiver:

```
Router#show controller optics 0/0/0/9
Controller State: Up
Transport Admin State: Maintenance
Laser State: On
LED State: Green
FEC State: FEC ENABLED
Optics Status:

Optics Type: QSFPDD 400G FR4
Wavelength: 1301.00 nm
Loopback Host: None
Loopback Media: Internal

Alarm Status:
----------------
Detected Alarms: None
LOS/LOL/Fault Status:
Performance Monitoring: Disable
```

Host Side Input Loopback Configuration

In host loopback internal or host side input loopback, the loopback signal coming from the NPU is looped back to the NPU on the host, that is, towards the local interface. This is indicated by the arrows labeled as 1 and 2 in the illustration.

Figure 20: Host Side Input Loopback on the Optical Transceiver



Configuration Example

This example shows how to enable host side input loopback on the optical transceiver:

```
Router#config
Router(config)#controller optics 0/0/0/9
Router(config-Optics)#secondary-admin-state maintenance
Router(config-Optics)#host loopback line
Loopback host is a traffic-affecting operation
Router(config-Optics)#commit
Router(config-Optics)#end
```

Running Configuration

This example shows the running configuration of the host side input loopback on the optical transceiver:

```
Router#show run controller optics 0/0/0/9
controller Optics0/0/0/9
 secondary-admin-state maintenance
 host loopback line
!
```

Verification

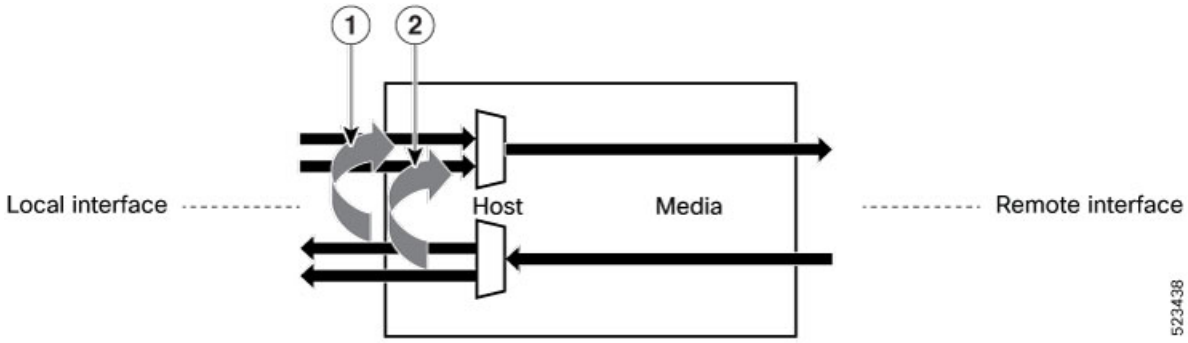
This example shows how to verify the host side input loopback configuration on the optical transceiver:

```
Router#show controller optics 0/0/0/9
Controller State: Up
Transport Admin State: Maintenance
Laser State: On
LED State: Green
FEC State: FEC ENABLED
Optics Status:

Optics Type: QSFPDD 400G FR4
Wavelength: 1301.00 nm
Loopback Host: Line
Loopback Media: None

Alarm Status:
----------------
Detected Alarms: None
LOS/LOL/Fault Status:
Performance Monitoring: Disable
```

Host Side Output Loopback Configuration

In host loopback line or host side output loopback, the signals received at the host side are looped back to the host side, indicating that the received data on the host is transmitted back to the host, that is, towards the local interface. This is indicated by the arrows labeled as 1 and 2 in the illustration.

Figure 21: Host Side Output Loopback on the Optical Transceiver



Configuration Example

This example shows how to enable host side output loopback on the optical transceiver:

```
Router#config
Router(config)#controller optics 0/0/0/9
Router(config-Optics)#secondary-admin-state maintenance
Router(config-Optics)#host loopback internal
Loopback host is a traffic-affecting operation
Router(config-Optics)#commit
Router(config-Optics)#end
```

Running Configuration

This example shows the running configuration on the optical transceiver:

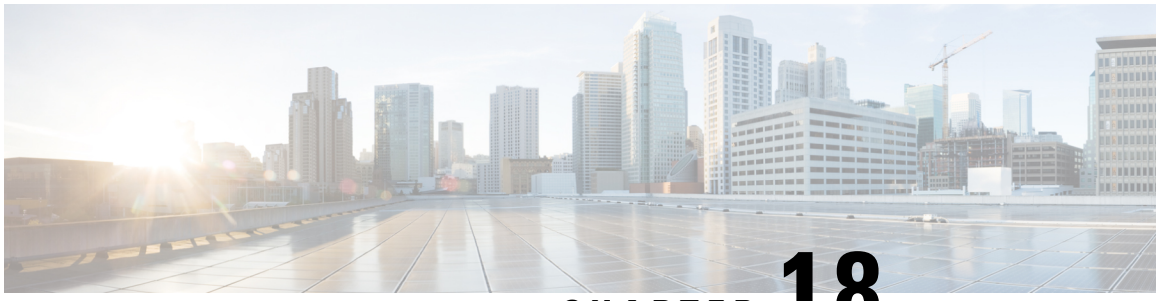
```
Router#show run controller optics 0/0/0/9
controller Optics0/0/0/9
 secondary-admin-state maintenance
 host loopback internal
!
```

Verification

This example shows how to verify the host side output loopback configuration on the optical transceiver:

```
Router#show controller optics 0/0/0/9
Controller State: Up
Transport Admin State: Maintenance
Laser State: On
LED State: Green
FEC State: FEC ENABLED
Optics Status:

Optics Type: QSFPDD 400G FR4
Wavelength: 1301.00 nm
Loopback Host: Internal
Loopback Media: None

Alarm Status:
```

```
----------------
Detected Alarms: None
LOS/LOL/Fault Status:
Performance Monitoring: Disable
```



CHAPTER 18

Managing Router Hardware

This chapter describes the concepts and tasks used to manage and configure the hardware components of a router running the Cisco IOS XR software.

This module contains the following topics:

MPA Reload

A Modular Port Adapter (MPA) is a hardware component used in networking equipment, such as routers and switches, to provide flexible and scalable port configurations.

A data path power-on timer is used during the power-on sequence of a network device to manage the initialization, stabilization, and diagnostic processes of the data path components. If an MPAcard doesn't come up within 20 minutes, the data path power-on timer expires, and the MPA goes for another reload to attempt recovery.

Note When a router enters an undefined state and disrupts the traffic due to the data path power-on timer expiry (timer associated with a data path has expired), reload the router using the **reload location** command.

RP Redundancy and Switchover

This section describes RP redundancy and switchover commands and issues.

Establishing RP Redundancy

Your router has two slots for RPs: RP0 and RP1 (see Figure 22: Redundant Set of RP Installed in Slots RP0 and RP1 in an Cisco 8608 8-Slot Centralized Chassis, on page 316 and Figure 23: Redundant Set of RP Installed in Slots RP0 and RP1 in an Cisco 8808 8-Slot Distributed Chassis, on page 317). RP0 is the slot on the left, facing the front of the chassis, and RP1 is the slot on right. These slots are configured for redundancy by default, and the redundancy cannot be eliminated. To establish RP redundancy, install RP into both slots.

Figure 22: Redundant Set of RP Installed in Slots RP0 and RP1 in an Cisco 8608 8-Slot Centralized Chassis

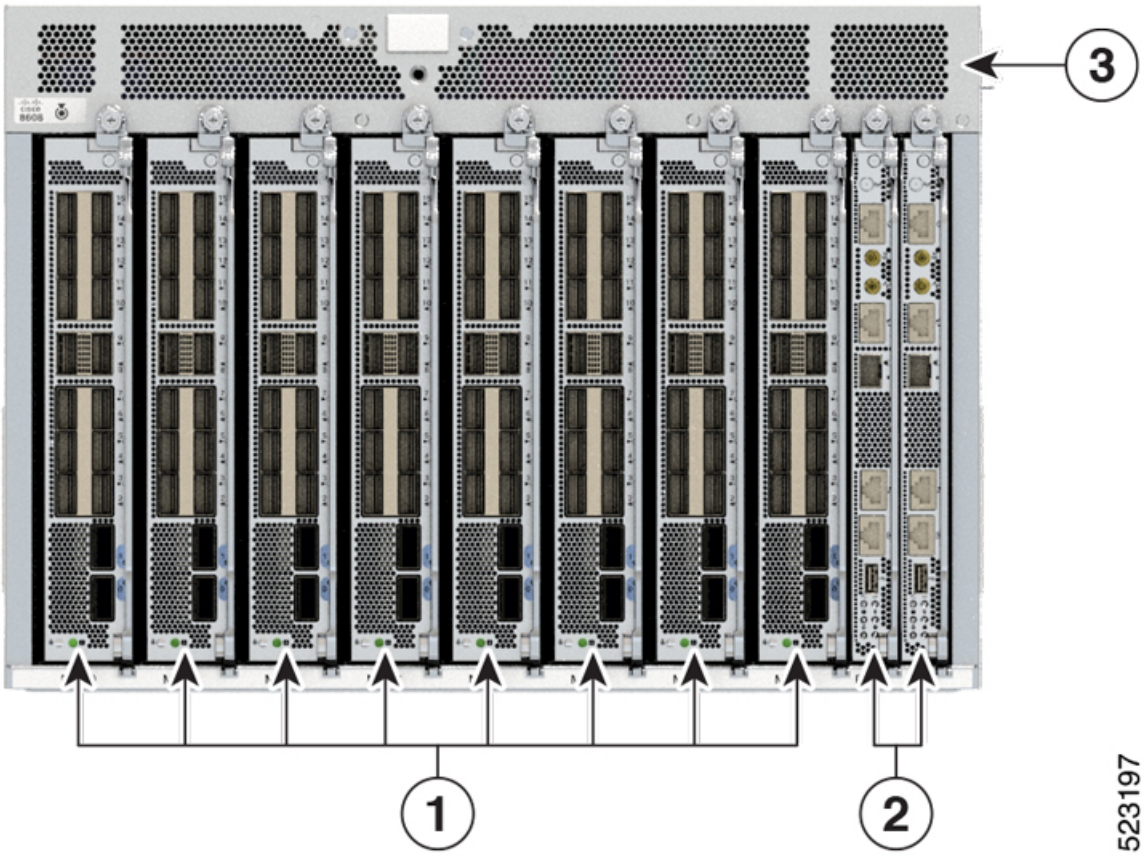
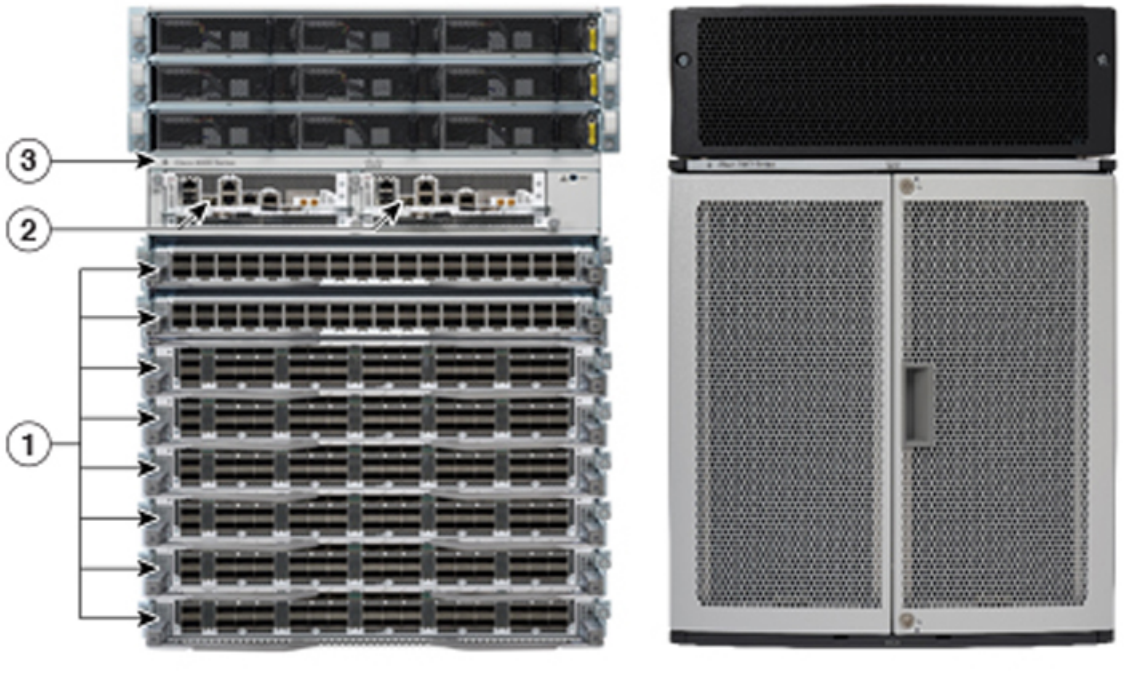


Figure 23: Redundant Set of RP Installed in Slots RP0 and RP1 in an Cisco 8808 8-Slot Distributed Chassis



1	Modular Port Adaptors (MPAs)
2	Route Processors (RPs)
3	Chassis

Determining the Active RP in a Redundant Pair

During system startup, one RP in each redundant pair becomes the active RP. You can tell which RP is the active RP in the following ways:

- The active RP can be identified by the green Active LED on the faceplate of the card. When the Active LED turns on, it indicates that the RP is active and when it turns off, it indicates that the RP is in standby.
- The slot of the active RP is indicated in the CLI prompt. For example:

```
RP/0/RP1/CPU0:router#
```

  In this example, the prompt indicates that you are communicating with the active RP in slot RP1.

- Enter the **show redundancy** command in EXEC mode to display a summary of the active and standby RP status. For example:

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

  This node (0/RP0/CPU0) is in ACTIVE role
  Partner node (0/RP1/CPU0) is in STANDBY role
  Standby node in 0/RP1/CPU0 is ready
```

```
Reload and boot info
----------------------
RP reloaded Fri Apr  9 03:44:28 2004: 16 hours, 51 minutes ago
This node booted Fri Apr  9 06:19:05 2004: 14 hours, 16 minutes ago
Last switch-over Fri Apr  9 06:53:18 2004: 13 hours, 42 minutes ago
Standby node boot Fri Apr  9 06:54:25 2004: 13 hours, 41 minutes ago
Standby node last not ready Fri Apr  9 20:35:23 2004: 0 minutes ago
Standby node last ready Fri Apr  9 20:35:23 2004: 0 minutes ago
There have been 2 switch-overs since reload
```

Role of the Standby RP

The second RP to boot in a redundant pair automatically becomes the standby RP. While the active RP manages the system and communicates with the user interface, the standby RP maintains a complete backup of the software and configurations for all cards in the system. If the active RP fails or goes off line for any reason, the standby RP immediately takes control of the system.

Summary of Redundancy Commands

RP redundancy is enabled by default in the Cisco IOS XR software, but you can use the commands described in Table 61: RP Redundancy Commands, on page 318 to display the redundancy status of the cards or force a manual switchover.

Table 61: RP Redundancy Commands

Command	Description
show redundancy	Displays the redundancy status of the RP. This command also displays the boot and switch-over history for the RP.
redundancy switchover	Forces a manual switchover to the standby RP. This command works only if the standby RP is installed and in the “ready” state.
show platform	Displays the status for node, including the redundancy status of the RP cards. In EXEC mode, this command displays status for the nodes assigned to the SDR. In administration EXEC mode, this command displays status for all nodes in the system.

Automatic Switchover

Automatic switchover from the active RP to the standby RP occurs only if the active RP encounters a serious system error, such as the loss of a mandatory process or a hardware failure. When an automatic switchover occurs, the RPs respond as follows:

- If a standby RP is installed and “ready” for switchover, the standby RP becomes the active RP. The original active RP attempts to reboot.
- If the standby RP is not in “ready” state, then both RPs reboot. The first RP to boot successfully assumes the role of active RP.

RP Redundancy During RP Reload

The **reload** command causes the active RP to reload the Cisco IOS XR software. When an RP reload occurs, the RPs respond as follows:

- If a standby RP is installed and "ready" for switchover, the standby RP becomes the active RP. The original active RP reboots and becomes the standby RP.
- If the standby RP is not in the "ready" state, then both RPs reboot. The first RP to boot successfully assumes the role of active RP.

Manual Switchover

If a standby RP is installed and ready for switchover, you can force a manual switchover using the **redundancy switchover** command or reloading the active RP using the **reload** command.

Manual Switchover Using the Reload Command

You can force a manual switchover from the active RP to the standby RP by reloading the active RP using the **reload** command. As active RP reboots, the current standby RP becomes active RP, and rebooting RP switches to standby RP.

```
RP/0/RP0/CPU0:router# reload
RP/0/RP1/CPU0:router#
```

Manual Switchover Using the Redundancy Switchover Command

You can force a manual switchover from the active RP to the standby RP using the **redundancy switchover** command.

If a standby RP is installed and ready for switchover, the standby RP becomes the active RP. The original active RP becomes the standby RP. In the following example, partial output for a successful redundancy switchover operation is shown:

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

  This node (0/RP0/CPU0) is in ACTIVE role
  Partner node (0/RP1/CPU0) is in STANDBY role
  Standby node in 0/RP1/CPU0 is ready

  RP/0/RP0/CPU0:router# redundancy switchover
  Updating Commit Database.  Please wait...[OK]
  Proceed with switchover 0/RP0/CPU0 -> 0/RP1/CPU0? [confirm]
  Initiating switch-over.
  RP/0/RP0/CPU0:router#

  <Your 'TELNET' connection has terminated>
```

In the preceding example, the Telnet connection is lost when the previously active RP resets. To continue management of the router, you must connect to the newly activated RP as shown in the following example:

```
User Access Verification

  Username: xxxxx
```

```
  Password: xxxxx
  Last switch-over Sat Apr 15 12:26:47 2009: 1 minute ago

RP/0/RP1/CPU0:router#
```

If the standby RP is not in “ready” state, the switchover operation is not allowed. In the following example, partial output for a failed redundancy switchover attempt is shown:

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

  Redundancy information for node 0/RP1/CPU0:
  ==========================================
  Node 0/RP0/CPU0 is in ACTIVE role
  Partner node (0/RP1/CPU0) is in UNKNOWN role

  Reload and boot info
  ----------------------
  RP reloaded Wed Mar 29 17:22:08 2009: 2 weeks, 2 days, 19 hours, 14 minutes ago
  Active node booted Sat Apr 15 12:27:58 2009: 8 minutes ago
  Last switch-over Sat Apr 15 12:35:42 2009: 1 minute ago
  There have been 4 switch-overs since reload

  RP/0/RP0/CPU0:router# redundancy switchover

  Switchover disallowed: Standby node is not ready.
```

Communicating with a Standby RP

The active RP automatically synchronizes all system software, settings, and configurations with the standby RP.

If you connect to the standby RP through the console port, you can view the status messages for the standby RP. The standby RP does not display a CLI prompt, so you cannot manage the standby card while it is in standby mode.

If you connect to the standby RP through the management Ethernet port, the prompt that appears is for the active RP, and you can manage the router the same as if you had connected through the management Ethernet port on the active RP.

NPU Power Optimization

Table 62: Feature History Table

Feature Name	Release Information	Description
NPU Power Optimization	Release 7.3.15	This feature lets you choose a predefined NPU power mode based on your network's individual requirements, and consequently reducing NPU power consumption. The **hw-module npu-power-profile** command is introduced for this feature.

Cisco 8000 series routers are powered by Cisco Silicon One Q200 and Q100 series processors. Cisco Silicon One processors offer high performance, flexible, and power-efficient routing silicon in the market.

NPU Power Optimization feature helps to reduce NPU power consumption by running a processor in a predefined mode. There are three NPU power modes—high, medium, and low. Based on your network traffic and power consumption requirements, you can choose to run the processor in any one of the three NPU power modes.

- High: The router will use the maximum amount of power, resulting in the best possible performance.
- Medium: The router power consumption and performance levels are both average.
- Low: The router operates with optimal energy efficiency while providing a modest level of performance.

Note We recommend that you work with your Cisco account representatives before implementing this feature in your network.

On a Q200-based Cisco 8200 series chassis, you can configure an NPU power mode on the entire router.

On a Q200-based Cisco 8800 series chassis, you can configure an NPU power mode only on fabric cards and line cards.

The following table lists the supported hardware, and their default NPU power mode:

Table 63: Supported Hardware and Default Modes

Supported Hardware	Default NPU Power Mode
Cisco 8200 32x400 GE 1RU fixed chassis (8201-32FH)	High
88-LC0-36FH without MACSec, based on Q200 Silicon Chip	Medium
88-LC0-36FH-M with MACSec, based on Q200 Silicon Chip	Medium
8808-FC0 Fabric Card, based on Q200 Silicon Chip	Low
8818-FC0 Fabric Card, based on Q200 Silicon Chip	Medium

Caution We recommend that you use the default NPU power mode on your router.

Limitations

The NPU power optimization is not supported on the Q100-based systems.

The NPU Power Profile mode is not supported on the following Q200-based line cards:

Table 64: Limitation on Hardware and Power Profile Modes

Hardware	Power Profile Mode
88-LC0-36FH-M	High
88-LC0-34H14FH	High

Configuring NPU Power Mode

Configuring NPU power mode on a fixed chassis:

The following example shows how to configure an NPU power mode on a fixed chassis:

```
RP/0/RP0/CPU0:ios(config)#hw-module npu-power-profile high
RP/0/RP0/CPU0:ios(config)#commit

RP/0/RP0/CPU0:ios(config)#reload
```

Note Note: Reload the chassis for the configurations changes to take effect.

Verifying NPU power mode configuration on a fixed chassis:

Use the **show controllers npu driver** command to verify the NPU power mode configuration:

```
RP/0/RP0/CPU0:ios#show controllers npu driver location 0/RP0/CPU0
Mon Aug 24 23:29:34.302 UTC
==========================================
NPU Driver Information
==========================================
Driver Version: 1
SDK Version: 1.32.0.1
Functional role: Active,     Rack: 8203, Type: lcc, Node: 0
Driver ready      : Yes
NPU first started : Mon Aug 24 23:07:41 2020
Fabric Mode:
NPU Power profile: High
Driver Scope: Node
Respawn count     : 1
Availablity masks :
        card: 0x1,     asic: 0x1,    exp asic: 0x1
...
```

Configuring NPU power mode on a modular chassis

The following example shows how to configure an NPU power mode on a fabric card and a line card:

```
RP/0/RP0/CPU0:ios(config)#hw-module npu-power-profile card-type FC high
RP/0/RP0/CPU0:ios(config)#hw-module npu-power-profile card-type LC low location 0/1/cpu0
RP/0/RP0/CPU0:ios(config)#commit
```

Note For the configurations to take effect, you must:

- Reload a line card if the configuration is applied on the line card.
- Reload a router if the configuration is applied on a fabric card.

Verifying the NPU power mode configuration on a modular chassis

Use the **show controllers npu driver location** command to verify the NPU power mode configuration:

```
RP/0/RP0/CPU0:ios#show controllers npu driver location 0/1/CPU0

Functional role: Active,     Rack: 8808, Type: lcc, Node: 0/RP0/CPU0
Driver ready      : Yes
NPU first started : Mon Apr 12 09:57:27 2021
Fabric Mode: FABRIC/8FC
NPU Power profile: High
Driver Scope: Rack
Respawn count     : 1
Availablity masks :
        card: 0xba,     asic: 0xcfcc,    exp asic: 0xcfcc
Weight distribution:
        Unicast: 80,      Multicast: 20
+-------------------------------------------------------------------+
| Process | Connection | Registration | Connection | DLL            |
| /Lib    | status     | status       | requests   | registration|
+-------------------------------------------------------------------+
| FSDB    | Active     | Active       |           1|  n/a        |
| FGID    | Active     | Active       |           1|  n/a        |
| AEL     | n/a        | n/a          |         n/a|  Yes        |
| SM      | n/a        | n/a          |         n/a|  Yes        |
+-------------------------------------------------------------------+

Asics :
HP - HotPlug event, PON - Power On reset
HR - Hard Reset,    WB  - Warm Boot
+-------------------------------------------------------------------------------------+
| Asic inst. | fap|HP|Slice|Asic|Admin|Oper | Asic state | Last |PON|HR |  FW  |
|   (R/S/A)  | id |  |state|type|state|state|            | init |(#)|(#)|  Rev |
+-------------------------------------------------------------------------------------+
| 0/FC1/2    | 202| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC1/3    | 203| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC3/6    | 206| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC3/7    | 207| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC4/8    | 208| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC4/9    | 209| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC5/10   | 210| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC5/11   | 211| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC7/14   | 214| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/FC7/15   | 215| 1| UP  |s123| UP  | UP  |NRML        |PON   |  1|  0|0x0000|
+-------------------------------------------------------------------------------------+

SI Info :
+-------------------------------------------------------------------------------------------------+
|   Card  | Board      | SI Board | SI Param | Retimer SI    | Retimer SI    | Front Panel
   |
|         | HW Version | Version  | Version  | Board Version | Param Version | PHY
   |
+-------------------------------------------------------------------------------------------------+
```

```
|  FC1    |  0.22      |     1    |     6    |       NA      |       NA      |       NA
   |
|  FC3    |  0.21      |     1    |     6    |       NA      |       NA      |       NA
   |
|  FC4    |  0.21      |     1    |     6    |       NA      |       NA      |       NA
   |
|  FC5    |  0.21      |     1    |     6    |       NA      |       NA      |       NA
   |
|  FC7    |  0.21      |     1    |     6    |       NA      |       NA      |       NA
   |
+-------------------------------------------------------------------------------------------+
Functional role: Active,     Rack: 8808, Type: lcc, Node: 0/1/CPU0
Driver ready      : Yes
NPU first started : Mon Apr 12 09:58:10 2021
Fabric Mode: FABRIC/8FC
NPU Power profile: Low
Driver Scope: Node
Respawn count     : 1
Availablity masks :
        card: 0x1,     asic: 0x7,    exp asic: 0x7
Weight distribution:
        Unicast: 80,      Multicast: 20
+-----------------------------------------------------------------+
| Process | Connection | Registration | Connection | DLL          |
| /Lib    | status     | status       | requests   | registration|
+-----------------------------------------------------------------+
| FSDB    | Active     | Active       |           1|  n/a         |
| FGID    | Inactive   | Inactive     |           0|  n/a         |
| AEL     | n/a        | n/a          |         n/a|  Yes         |
| SM      | n/a        | n/a          |         n/a|  Yes         |
+-----------------------------------------------------------------+

Asics :
HP - HotPlug event, PON - Power On reset
HR - Hard Reset,    WB  - Warm Boot
+-------------------------------------------------------------------------------+
| Asic inst. | fap|HP|Slice|Asic|Admin|Oper | Asic state | Last |PON|HR |  FW   |
|   (R/S/A)  | id |  |state|type|state|state|            | init |(#)|(#)|  Rev |
+-------------------------------------------------------------------------------+
| 0/2/0      |   8| 1| UP  |npu | UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/2/1      |   9| 1| UP  |npu | UP  | UP  |NRML        |PON   |  1|  0|0x0000|
| 0/2/2      |  10| 1| UP  |npu | UP  | UP  |NRML        |PON   |  1|  0|0x0000|
+-------------------------------------------------------------------------------+

SI Info :
+-------------------------------------------------------------------------------------------+
|   Card  | Board      | SI Board | SI Param | Retimer SI    | Retimer SI    | Front Panel
   |
|         | HW Version | Version  | Version  | Board Version | Param Version | PHY
   |
+-------------------------------------------------------------------------------------------+
|  LC2    |  0.41      |     1    |     9    |       NA      |       NA      |  DEFAULT
   |
+-------------------------------------------------------------------------------------------+
```

Dynamic Power Management

Table 65: Feature History Table

Feature Name	Release Information	Description
Dynamic Power Management	Release 7.3.15	The Dynamic Power Management feature considers certain dynamic factors before allocating power to the fabric and line cards. This feature has the following benefits: • Reduces number of PSUs required by accurately representing the maximum power consumption • Improves PSU efficiency by providing more accurate power allocation This feature thus optimizes power allocation and avoids overprovisioning power to a router.
Dynamic Power Management	Release 7.3.2	Previously available for fabric and line cards, this feature that helps avoid excess power allocation by considering dynamic factors before allocating power to them is now available for optical modules. To view the power allocation on a per port basis, a new command "**show environment power allocated [details]**" is introduced.
Dynamic Power Management	Release 7.3.3	The Dynamic Power Management feature is now supported on the following Cisco 8100 and 8200 series routers: • Cisco 8201 • Cisco 8202 • Cisco 8201-32-FH • Cisco 8101-32-FH
Dynamic Power Management	Release 7.5.2	The Cisco 8202-32FH-M router will now consider dynamic factors, such as optical modules, NPU power profile, and MACsec mode to enable improved power allocation and utilization.

Prior to Cisco IOS XR Release 7.3.15, when Cisco 8000 series routers were powered on or reloaded, the power management feature reserved power to fabric cards and allocated maximum power to line cards. The

power management feature wouldn't consider dynamic factors, such as the type of fabric or line cards in the chassis, or whether a fabric or line card was present in a slot.

The Dynamic Power Management feature considers such dynamic factors before allocating power to the fabric and line cards.

This feature has the following benefits:

- Reduces number of PSUs required by accurately representing the maximum power consumption
- Improves PSU efficiency by providing more accurate power allocation

This feature thus optimizes power allocation and avoids overprovisioning power to a router.

This feature is supported on the following Cisco 8000 series routers:

- Cisco 8804, 8808, 8812, and 8818 routers
- Cisco 8201, 8202, 8201-32-FH, and 8202-32FH-M routers
- Cisco 8101-32-FH

By default, this feature is enabled on the router.

The Dynamic Power Management feature allocates the total power to a router and its fabric card or line card based on the following parameters:

- Number and type of fabric cards installed on the router
- Fabric cards operating modes (5FC or 8FC)
- Number and type of line cards installed on the router
- Combination of line card and fabric card types installed
- NPU power mode configured on a fabric card
- Number and type of optics installed (supported in Cisco IOS XR Software Release 7.3.2 and later)
- MACSec-enabled ports (supported from Cisco IOS XR Software Release 7.3.3 and later)

  For details, see *Dynamic Power Management for MACSec-Enabled Ports* section in the *Configuring MACSec* chapter in the *System Security Configuration Guide for Cisco 8000 Series Routers*.

On 8202-32FH-M router, the Dynamic Power Management feature allocates the total power to a router based on the following parameters:

- Optical modules installed.
- NPU power profile. To identify the mode on which the router is operating, use the `hw-module npu-power-profile` command.
- MACSec mode. By default, MACSec mode is disabled on 8202-32FH-M router.

Note We recommend you work with your Cisco account representatives to calculate power requirements for the Cisco 8000 series router.

Power Allocation to Empty Card Slot

This feature allocates a minimum required power for all empty LC or FC slots. This minimum power is required to boot the CPU and FPGAs immediately when a card is inserted. The feature doesn't control booting up the CPU and FPGAs. Also, the minimum power is required to detect the card type before the feature decides if there's enough power to power up the data path.

For example, the following **show environment power** command output displays various LC or FC card statuses, and also shows allocated and used power.

Note The allocated power capacity shown in the following **show** command output isn't standard capacity. The allocated power capacity varies depending on various other factors.

```
Router# show environment power
Thu Apr 22 12:03:06.754 UTC
===============================================================================
CHASSIS LEVEL POWER INFO: 0
===============================================================================
   Total output power capacity (N + 1)             :    9600W +     6300W
   Total output power required                     :    9241W
   Total power input                               :    6146W
   Total power output                              :    5826W

===============================================================================
   Power       Supply         -------Input--------   -----Output---     Status
   Module      Type            Volts A/B   Amps A/B   Volts     Amps
===============================================================================
   0/PT0-PM0   PSU6.3KW-HV     245.5/245.7 5.1/5.0    54.7      43.1     OK
   0/PT0-PM1   PSU6.3KW-HV     0.0/245.2   0.0/7.4    54.3      31.7     OK
   0/PT0-PM2   PSU6.3KW-HV     0.0/246.9   0.0/7.5    54.1      32.3     OK

Total of Power Modules:       6146W/25.0A              5826W/107.1A

===============================================================================
   Location     Card Type               Power       Power        Status
                                        Allocated   Used
                                        Watts       Watts
===============================================================================
   0/RP0/CPU0   8800-RP                 95          69           ON
   0/RP1/CPU0   -                       95          -            RESERVED
   0/0/CPU0     88-LC0-36FH             796         430          ON
   0/1/CPU0     -                       102         -            RESERVED
   0/2/CPU0     88-LC0-36FH             796         430          ON
   0/3/CPU0     -                       102         -            RESERVED
   0/4/CPU0     -                       102         -            RESERVED
   0/5/CPU0     -                       102         -            RESERVED
   0/6/CPU0     -                       102         -            RESERVED
   0/7/CPU0     -                       102         -            RESERVED
   0/8/CPU0     -                       102         -            RESERVED
   0/9/CPU0     88-LC0-36FH             102         -            OFF
   0/10/CPU0    -                       102         -            RESERVED
   0/11/CPU0    -                       102         -            RESERVED
   0/FC0        -                       26          -            RESERVED
   0/FC1        -                       26          -            RESERVED
   0/FC2        -                       26          -            RESERVED
   0/FC3        8812-FC                 784         509          ON
   0/FC4        8812-FC                 784         503          ON
   0/FC5        8812-FC                 26          -            OFF
   0/FC6        8812-FC                 26          -            OFF
   0/FC7        8812-FC                 26          -            OFF
```

```
0/FT0        8812-FAN              1072        1000         ON
0/FT1        8812-FAN              1072        1012         ON
0/FT2        8812-FAN              1072        861          ON
0/FT3        8812-FAN              1072        1033         ON
```

This table describes the card slot statuses:

Table 66: Router Card Slot Status

Status	Description
RESERVED	When a slot is empty
OFF	When a card is inserted in a slot but power isn't allocated to the card
ON	When a card is allocated power and the card is in operational state

Low-Power Condition

When you insert an LC or FC in a card slot at the time when the router doesn't have enough power available to allocate to the new card, the dynamic power management feature doesn't provision power to the card. It raises the *ev_power_budget_not_ok* alarm, and gracefully shuts down the card.

In the following **show** command output, an FC inserted in the card slot location 0/FC6 is gracefully shut down due to lack of power:

```
Router# show shelfmgr history events location 0/FC6
Thu Apr 22 12:03:11.763 UTC
NODE NAME     : 0/FC6
CURRENT STATE : CARD_SHUT_POWERED_OFF
TIME STAMP    : Apr 20 2021 16:49:52
--------------------------------------------------------------------------------
DATE        TIME (UTC)  EVENT                    STATE
--------------------------------------------------------------------------------
Apr 20 2021 16:49:52    ev_powered_off           CARD_SHUT_POWERED_OFF
Apr 20 2021 16:49:52    ev_device_offline        STATE_NOT_CHANGED
Apr 20 2021 16:49:52    ev_unmapped_event        STATE_NOT_CHANGED
Apr 20 2021 16:49:48    transient_condition      CARD_SHUTDOWN
Apr 20 2021 16:49:48    ev_check_card_down_reaso CHECKING_DOWN_REASON
Apr 20 2021 16:49:48    ev_timer_expiry          CARD_SHUTDOWN_IN_PROGRESS
Apr 20 2021 16:48:46    ev_power_budget_not_ok   CARD_SHUTDOWN_IN_PROGRESS
Apr 20 2021 16:48:45    transient_condition      POWER_BUDGET_CHECK
Apr 20 2021 16:48:45    ev_fpd_upgrade_not_reqd  CARD_STATUS_CHECK_COMPLETE
Apr 20 2021 16:47:45    ev_card_status_check     CARD_STATUS_CHECK
Apr 20 2021 16:47:45    ev_card_info_rcvd        CARD_INFO_RCVD
Apr 20 2021 16:47:44    ev_device_online         DEVICE_ONLINE
Apr 20 2021 16:47:43    ev_timer_expiry          CARD_POWERED_ON
Apr 20 2021 16:47:33    ev_powered_on            CARD_POWERED_ON
Apr 20 2021 16:47:33    init                     CARD_DISCOVERED
--------------------------------------------------------------------------------
```

However, after an LC, FC, or chassis reload, the dynamic power management feature can't ensure that the same LCs, FCs, optics, or interfaces, which were operational earlier (before the reload), would become active again.

Note During a low-power condition, this feature doesn't borrow power from a redundant power supply.

Power Allocation to Optics

From Cisco IOS XR Release 7.3.2 onwards, power requirement for optics is also considered before allocating power to them.

To identify the power allocated for a particular interface, use the **show environment power allocated** [**details**] **location** *location* command.

When the optical modules are inserted, power is automatically allocated for that interface. If power has been allocated to the interface, then use the "**no shut**" command to enable the interface.

```
Router# show environment power allocated location 0/3/CPU0
Thu Oct  7 22:27:35.732 UTC
================================================================================
   Location    Components               Power
                                        Allocated
                                        Watts
================================================================================
  0/3/CPU0     Data-path                772
               OPTICS                   138
================================================================================
               Total                    910

Router# show environment power allocated details location 0/3/CPU0
Thu Oct  7 22:27:42.221 UTC
================================================================================
   Location    Components               Power
                                        Allocated
                                        Watts
================================================================================
  0/3/CPU0     Data-path                772
               0/3/0/0                  3
               0/3/0/1                  3
               0/3/0/2                  3
               0/3/0/3                  3
               0/3/0/4                  3
               0/3/0/5                  3
               0/3/0/6                  3
               0/3/0/7                  3
               0/3/0/8                  3
               0/3/0/9                  3
               0/3/0/10                 3
               0/3/0/11                 3
               0/3/0/12                 3
               0/3/0/13                 3
               0/3/0/14                 3
               0/3/0/15                 3
               0/3/0/16                 3
               0/3/0/17                 3
               0/3/0/18                 3
               0/3/0/19                 3
               0/3/0/20                 3
               0/3/0/21                 3
               0/3/0/22                 3
               0/3/0/23                 3
               0/3/0/24                 3
```

```
                0/3/0/25                3
                0/3/0/26                3
                0/3/0/27                3
                0/3/0/28                3
                0/3/0/29                3
                0/3/0/30                3
                0/3/0/31                3
                0/3/0/32                3
                0/3/0/33                3
                0/3/0/34                3
                0/3/0/35                3
                0/3/0/36                3
                0/3/0/37                3
                0/3/0/38                3
                0/3/0/39                3
                0/3/0/40                3
                0/3/0/41                3
                0/3/0/42                3
                0/3/0/43                3
                0/3/0/44                3
                0/3/0/46                3
==============================================================================
                Total                   910
```

When the power is not allocated to the interface, the following syslog error and alarms are displayed

```
!<--Syslog Error-->!
#LC/0/3/CPU0:Oct  7 22:46:48.114 UTC: optics_driver[165]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :POWER ALLOCATION FAIL :DECLARE :0/3/CPU0:  Optics0/3/0/44
LC/0/3/CPU0:Oct  7 22:46:48.114 UTC: optics_driver[165]:
%L2-OPTICS-2-QSFP_POWER_ALLOCATION_FAILURE : Not enough power available to enable Optics
0/3/0/44

!<--Alarm-->!
Router#show alarms brief system active
Thu Oct  7 22:47:19.569 UTC

------------------------------------------------------------------------------------------
Active Alarms
------------------------------------------------------------------------------------------
Location        Severity     Group            Set Time                    Description



------------------------------------------------------------------------------------------
0/3/CPU0        Major        Software         10/07/2021 22:46:48 UTC     Optics0/3/0/44 -
 hw_optics:  Lack of available power to enable the optical module


0/3/CPU0        Major        Software         10/07/2021 22:47:06 UTC     Optics0/3/0/46 -
 hw_optics:  Lack of available power to enable the optical module
```

If power is not allocated to an interface and you attempt to enable that interface using the "**no shut**" command, the following syslog error is displayed:

```
LC/0/2/CPU0:Aug 30 18:01:14.930 UTC: eth_intf_ea[262]: %PLATFORM-VEEA-1-PORT_NOT_ENABLED :
 Power not allocated to enable the interface HundredGigE0_2_0_6.
```

Power Allocation to Fixed-Port Routers

The following **show environment power** command output displays power information for fixed-port routers and components.

```
Router# show environment power
Wed Feb 16 21:05:10.001 UTC
===============================================================================
CHASSIS LEVEL POWER INFO: 0
===============================================================================
   Total output power capacity (Group 0 + Group 1) :    1400W +     1400W
   Total output power required                     :    1033W
   Total power input                               :     390W
   Total power output                              :     255W

Power Group 0:
===============================================================================
   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
===============================================================================
   0/PM0       PSU1.4KW-ACPE   244.5     0.8     12.0      11.1     OK

Total of Group 0:              195W/0.8A         133W/11.1A

Power Group 1:
===============================================================================
   Power       Supply         ------Input----   ------Output---     Status
   Module      Type            Volts     Amps    Volts     Amps
===============================================================================
   0/PM1       PSU1.4KW-ACPE   244.2     0.8     12.0      10.2     OK

Total of Group 1:              195W/0.8A         122W/10.2A

===============================================================================
   Location     Card Type               Power       Power        Status
                                        Allocated   Used
                                        Watts       Watts
===============================================================================
   0/RP0/CPU0   8201                    893         -            ON
   0/FT0        FAN-1RU-PE              28          -            ON
   0/FT1        FAN-1RU-PE              28          -            ON
   0/FT2        FAN-1RU-PE              28          -            ON
   0/FT3        FAN-1RU-PE              28          -            ON
   0/FT4        FAN-1RU-PE              28          -            ON
```

To identify the power allocated for a particular interface, use the **show environment power allocated** [**details**] **location** *location* command.

```
Router# show environment power allocated location 0/RP0/CPU0
Wed Feb 16 21:05:21.360 UTC
===============================================================================
   Location    Components              Power
                                       Allocated
                                       Watts
===============================================================================
  0/RP0/CPU0   Data-path               858
               OPTICS                  35
===============================================================================
               Total                   893


Router# show environment power allocated details location 0/RP0/CPU0
Wed Feb 16 21:05:36.142 UTC
===============================================================================
   Location    Components              Power
                                       Allocated
                                       Watts
===============================================================================
  0/RP0/CPU0   Data-path               858
```

```
                0/0/0/19                21
                0/0/0/18                14
================================================================================
                Total                   893
```

Disabling Dynamic Power Management

By default, the dynamic power management is enabled on a router. The following example shows how to disable dynamic power management:

```
RP/0/RP0/CPU0:ios(config)#power-mgmt action disable
RP/0/RP0/CPU0:ios(config)#commit
```

Caution After disabling the dynamic power management feature, you must manage the router power on your own. So, use this command with caution.

Note To reenable dynamic power management, use the **no power-mgmt action disable** command.

On-demand transfer of Redundant Power Modules to Power Reservation Pool

Table 67: Feature History Table

Feature Name	Release Information	Feature Description
On-demand transfer of Redundant Power Modules to Power Reservation Pool	Release 7.11.1	The Cisco 8800 Series Modular Routers now have a functionality that allows them to transfer their redundant Power Supply Units (PSUs) to the power reservation pool when there is inadequate power supply. This capability helps prevent the router from shutting down hardware components due to a lack of power in the reservation pool, which used to occur due to the router prioritizing redundancy over power availability in the power reservation pool. Consequently, the router now raises an alarm indicating redundancy loss when it transfers PSUs to the power reservation pool. This feature ensures that the router components reserve the necessary power, even when redundancy is enabled.

The Cisco 8000 Series Modular Routers offer redundancy while managing Power Supply Units (PSUs), providing continuous operation if there is PSU failure. By default, the router operates in N+1 redundancy, where N represents the number of PSUs allotted to the power reservation pool for powering the router components, and 1 indicates the backup PSU. You can use the **power-mgmt redundancy-num-pms** *number* command in XR Config mode mode to configure the PSU redundancy from N+1 to N+x, where x is the number of redundant PSUs required. The total number of functioning PSUs must be at least x more than the number of PSUs required to support the power demanded by all the components in the system for optimal router functionality. The range of values assigned to x is 0–11, where 0 implies no power redundancy. The router uses the redundant PSUs only when there is a PSU failure. But, if the power requirement of the router increases than the available power offered by PSUs, the router prioritizes maintaining PSU redundancy overpowering the components.

Starting from Cisco IOS XR Release 7.11.1, the Cisco 8800 Modular Routers prioritize powering the router components over preserving redundancy. The router transfers the redundant PSUs to a power reservation pool to power the router components on demand. The router utilizes the redundant PSUs to increase the power capacity in the power reservation pool rather than maintaining redundancy. For example, consider a scenario with 18900W (3 6300W PSUs) available power. Initially, the router reserves 12600W (using 2 PSUs) in the power reservation pool and retains 6300W (one PSU) as a backup to maintain N+1 redundancy. Suppose the router needs to reserve power for any components to power up and needs more power than is available in the reservation pool. In that case, the router uses the entire 18900W with all three PSUs to power the components by transferring the redundant PSU to the power reservation pool. The router then triggers a redundancy loss alarm with such an assignment. However, if any further actions result in reduced power consumption in the router, the system automatically restores redundancy and clears the redundancy lost alarm.

On redundancy loss, the router raises a **Critical** severity **Power Module redundancy lost** alarm. You can use the **show alarms brief** command to view the redundancy lost alarm.

Syslog messages for transforming redundant PSU into borrowable resource:

Syslog message created while redundancy loss (transforming redundant PSU to functional PSU):

```
RP/0/RP0/CPU0:Jul 24 11:49:01.316 UTC: envmon[214]: %PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR
 :Power Module redundancy lost :DECLARE :0:
```

Syslog message created while restoring redundancy:

```
RP/0/RP0/CPU0:Jul 24 11:49:11.375 UTC: envmon[214]: %PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR
 :Power Module redundancy lost :CLEAR :0:
```

You can also use the **show environment** view the redundancy status of the PSUs in the router.

The following section details the commands to verify the redundancy status in the router:

Router with N+1 redundancy:

```
Router:ios# show environment power
          ================================================================================

            CHASSIS LEVEL POWER INFO: 0
          ================================================================================

            Total output power capacity (N + 1)             :   12600W +     6300W
            Total output power required                     :   11545W
            Total power input                               :    3302W
            Total power output                              :    3004W

          ================================================================================

            Power       Supply         -------Input--------   -----Output---     Status
            Module      Type            Volts A/B   Amps A/B   Volts     Amps
          ================================================================================
```

```
 0/PT5-PM0   PSU6.3KW-HV     240.5/241.3 2.2/2.4    55.1      18.3     OK
 0/PT5-PM1   PSU6.3KW-HV     240.5/240.8 2.1/2.3    54.8      17.3     OK
 0/PT5-PM2   PSU6.3KW-HV     242.2/241.1 2.3/2.4    54.9      19.1     OK

 Total of Power Modules:       3302W/13.7A              3004W/54.7A

==============================================================================

 Location     Card Type              Power       Power        Status
 Allocated   Used
 Watts       Watts
==============================================================================

 0/RP0/CPU0   8800-RP                105         78           ON
 0/RP1/CPU0   -                      105         -            RESERVED
 0/0/CPU0     8800-LC-36FH           1097        513          ON
 0/1/CPU0     -                      102         -            RESERVED
 0/2/CPU0     88-LC0-36FH            102         0            OFF
 0/3/CPU0     -                      102         -            RESERVED
 0/4/CPU0     -                      102         -            RESERVED
 0/5/CPU0     -                      102         -            RESERVED
 0/6/CPU0     -                      102         -            RESERVED
 0/7/CPU0     -                      102         -            RESERVED
 0/8/CPU0     -                      102         -            RESERVED
 0/9/CPU0     -                      102         -            RESERVED
 0/10/CPU0    -                      102         -            RESERVED
 0/11/CPU0    -                      102         -            RESERVED
 0/12/CPU0    -                      102         -            RESERVED
 0/13/CPU0    -                      102         -            RESERVED
 0/14/CPU0    -                      102         -            RESERVED
 0/15/CPU0    -                      102         -            RESERVED
 0/16/CPU0    -                      102         -            RESERVED
 0/17/CPU0    -                      102         -            RESERVED
 0/FC0        -                      32          -            RESERVED
 0/FC1        -                      32          -            RESERVED
 0/FC2        8818-FC0               584         475          ON
 0/FC3        -                      32          -            RESERVED
 0/FC4        8818-FC0               584         472          ON
 0/FC5        -                      32          -            RESERVED
 0/FC6        -                      32          -            RESERVED
 0/FC7        -                      32          -            RESERVED
 0/FT0        8818-FAN               1786        237          ON
 0/FT1        8818-FAN               1786        228          ON
 0/FT2        8818-FAN               1786        234          ON
 0/FT3        8818-FAN               1786        228          ON
```

Router with redundancy loss:

```
Router:ios# sh env power
==============================================================================

 CHASSIS LEVEL POWER INFO: 0
==============================================================================

 Total output power capacity (N + 1)             :   18900W +        0W
 Total output power required                     :   12689W
 Total power input                               :    3302W
 Total power output                              :    3004W

==============================================================================

 Power       Supply         -------Input--------   -----Output---     Status
 Module      Type            Volts A/B   Amps A/B   Volts     Amps
==============================================================================
```

```
  0/PT5-PM0   PSU6.3KW-HV     240.5/241.3 2.2/2.4    55.1      18.3     OK
  0/PT5-PM1   PSU6.3KW-HV     240.5/240.8 2.1/2.3    54.8      17.3     OK
  0/PT5-PM2   PSU6.3KW-HV     242.2/241.1 2.3/2.4    54.9      19.1     OK

  Total of Power Modules:       3302W/13.7A              3004W/54.7A

==============================================================================

  Location     Card Type              Power       Power        Status
  Allocated   Used
  Watts       Watts
==============================================================================

  0/RP0/CPU0   8800-RP                105         78           ON
  0/RP1/CPU0   -                      105         -            RESERVED
  0/0/CPU0     8800-LC-36FH           1097        513          ON
  0/1/CPU0     -                      102         -            RESERVED
  0/2/CPU0     88-LC0-36FH            916         510          ON
  0/3/CPU0     -                      102         -            RESERVED
  0/4/CPU0     -                      102         -            RESERVED
  0/5/CPU0     -                      102         -            RESERVED
  0/6/CPU0     -                      102         -            RESERVED
  0/7/CPU0     -                      102         -            RESERVED
  0/8/CPU0     -                      102         -            RESERVED
  0/9/CPU0     -                      102         -            RESERVED
  0/10/CPU0    -                      102         -            RESERVED
  0/11/CPU0    -                      102         -            RESERVED
  0/12/CPU0    -                      102         -            RESERVED
  0/13/CPU0    -                      102         -            RESERVED
  0/14/CPU0    -                      102         -            RESERVED
  0/15/CPU0    -                      102         -            RESERVED
  0/16/CPU0    -                      102         -            RESERVED
  0/17/CPU0    -                      102         -            RESERVED
  0/FC0        -                      32          -            RESERVED
  0/FC1        -                      32          -            RESERVED
  0/FC2        8818-FC0               749         475          ON
  0/FC3        -                      32          -            RESERVED
  0/FC4        8818-FC0               749         472          ON
  0/FC5        -                      32          -            RESERVED
  0/FC6        -                      32          -            RESERVED
  0/FC7        -                      32          -            RESERVED
  0/FT0        8818-FAN               1786        237          ON
  0/FT1        8818-FAN               1786        225          ON
  0/FT2        8818-FAN               1786        234          ON
  0/FT3        8818-FAN               1786        228          ON
Router:ios# sh alarms brief system active

------------------------------------------------------------------------------
            Active Alarms

------------------------------------------------------------------------------
            Location        Severity     Group           Set Time
Description



------------------------------------------------------------------------------
            0/RP0/CPU0      Critical     Software        10/27/2023 00:22:08 UTC
Redundancy Partner Not Present


            0               Major        Environ         10/27/2023 00:23:48 UTC    Power
 Module redundancy lost
```

```
           0/RP0/CPU0      Minor        Fabric          10/27/2023 00:22:39 UTC    Fabric
Plane-0 status

           0/RP0/CPU0      Minor        Fabric          10/27/2023 00:22:39 UTC    Fabric
Plane-1 status

           0/RP0/CPU0      Minor        Fabric          10/27/2023 00:22:39 UTC    Fabric
Plane-3 status

           0/RP0/CPU0      Minor        Fabric          10/27/2023 00:22:39 UTC    Fabric
Plane-5 status

           0/RP0/CPU0      Minor        Fabric          10/27/2023 00:22:39 UTC    Fabric
Plane-6 status

           0/RP0/CPU0      Minor        Fabric          10/27/2023 00:22:39 UTC    Fabric
Plane-7 status

           0/RP0/CPU0      Major        Software        10/27/2023 00:22:59 UTC
Communications Failure With Cisco Licensing Cloud

           0               Major        Environ         10/27/2023 00:23:48 UTC    Power
Module redundancy lost
```

Ability to Set Maximum Power Limit for the Router

Table 68: Feature History Table

Feature Name	Release Information	Feature Description
Ability to Set Maximum Power Limit for the Router	Release 7.11.1	We are introducing functionality to set the maximum power limit for a router to improve power management and distribution in the PSUs. It prevents a router from using more than the configured power and also gives the ability to limit the reservation pool regardless of how many power supplies are present. In the previous releases, the ability to prevent a router from using more than a configured amount of power was unavailable. This feature introduces the following change: **CLI** • **power-mgmt configured-power-capacity**

In the earlier releases, there was no mechanism to limit the power a router consumed. Routers could draw more than the infrastructure could handle. Over power consumption could result in system brownout.

With the Cisco IOS XR Software Release 7.11.1, you can allocate system power based on max power capacity configuration. This prevents the router from allocating more power than the infrastructure can handle. It also gives you the ability to limit power to a router according to your infrastructure requirements. The max power capacity parameter doesn't allow power consumed by the hardware to cross the configured amount.

The criteria to set maximum power limit is that the value must be set between the current allocated power and the available maximum power at time of configuration.

This feature is not applicable for fixed routers.

A new command **power-mgmt configured-power-capacity** has been introduced with this feature.

A new alarm **PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR :Power reservation exceeds configured power** is introduced to be raised when the max power capacity is crossed.

Note This alarm is extremely rare and is raised only when the power reservation exceeds configured power. This can only happen when hardware is inserted, it is granted power without a request, such as a fan tray.

Configuring the Compatibility Mode for Various NPU Types

Table 69: Feature History Table

Feature Name	Release Information	Description
Configure Compatibility Mode for Q100 and Q200-based Line Cards	Release 7.7.1	You can now configure the compatibility behavior of line cards to operate in Q100 mode (default behavior) or in Q200 mode when you have a mix of Q100-based line cards and Q200-based line cards that are installed in a router. In earlier releases, in a mixed mode combination, where multiple generations of line cards were installed on a distributed chassis, the behavior was to make the second-generation line cards interoperate with the first-generation line cards. However, this led the NPUs to set lower resource limits for the newer generation line cards to ensure backward compatibility. Also, the router didn't fully utilize the improved scale, higher capacity, and feature-rich capabilities of the newer generation line cards. This compatibility feature now enables you to select if you want the line cards to operate in Q100 or Q200 NPU mode. The **hw-module profile npu-compatibility** command is introduced for this feature.

In earlier releases, if you install a mix of Q100-based line cards and Q200-based line cards, the Q200-based line cards operate in a scaled-down (Q100) mode by default.

The compatibility feature, applicable to Cisco 8800 Series modular/distributed chassis, now allows you to choose if you want line cards to operate in Q100 (default behavior), Q200, or P100 mode. In Q200 mode, the router boots only the Q200-based line cards and gracefully shuts down the Q100-based line cards.

For example, if a router has a Q100 NPU-based line card and you try to add a line card from the Q200 NPU-based line card, the Q200 NPU line card operates in a scaled down mode to be able to work with the older generation-Q100 line cards. With the new implementation, you can choose if you want the router to work in the Q100 mode or shutdown the Q100-based linecards, and use the Q200 NPU-based line cards in the Q200 mode.

FAQs About the Compatibility Modes for Various NPU Types

- **Can the line cards still be used in scaled down mode, like in the previous scenario?**

  Yes, you can still switch to the previous implementation, if you may, to the scaled down mode.

- **What all ASICs can participate in the compatibility mode implementation?**

  P100, Q200, Q100.

- **Is there any default ASIC set by the system?**

  The ASIC default is based on the Fabric Cards (FCs) and route processor cards used in a distributed chassis. However, you can choose to change the ASIC mode to Q200, Q100.

- **Do I need to reboot the router after implementing a new NPU mode?**

  Yes, reboot the router for the new NPU mode to take effect.

- **What defines an NPU mode?**

  NPU mode is determined by the Route Processor (RP) and the Fabric Card (FC). During the router's boot-up process, it initially identifies the RP and the FC, setting the corresponding NPU mode regardless of the line cards present in the router.

Usage Guidelines and Limitations

The following guidelines and limitations apply when you configure the line cards from different ASIC families:

- By default, a mix of Q100 and Q200 line cards results in the Q200 line cards operating in Q100 (scaled-down) mode. Configuring Q100 mode results in the same (default) behavior. Similarly, a mix of P100 and Q200 line cards results in the Q200 line cards operating in P100 (scaled-down) mode. Configuring P100 mode results in the same (default) behavior.
- To be able to use the improved scale, higher capacity, and feature-rich capabilities of the Q200-based line cards, use the **hw-module profile npu-compatibility** command and set it to operate in the Q200 mode. Else, the Q200-based line cards scale down to the Q100 mode, which is the default behavior.
- Reboot the router for the compatibility mode to take effect. If the system detects a noncompatible line card, it shuts down that line card. For example, in Q200 mode, the router boots only the Q200-based line cards and gracefully shuts down the Q100-based line cards.
- The **hw-module profile npu-compatibility** command isn't configurable on the Cisco 8200 Series fixed router and Cisco 8608 router.
- For 8800-RP, the default NPU mode is Q100. For 8800-RP2, the default NPU mode is Q200.
- For the various fabric card types available, the following scenarios may be applicable:
  - 8800-RP Route Processor Card - if the router boots up with an 8800-RP route processor card without any fabric card, then the default mode is set to Q100.
  - 8800-RP2 Route Processor Card - if the router boots up with a 8800-RP2 route processor card without any fabric card, then the router sets the default mode to P100. If you insert a Q200 fabric card, then router reload is required.
  - Swapping Fabric Cards - if the router initially boots with Q200 fabric cards and you later replace them with F100 fabric cards, a router reload is necessary.

This table lists the Q100, Q200 line cards that support the compatibility mode:

ASIC Family	Line Card
Q100-based line cards	8800-LC-48H
	8800-LC-36FH

ASIC Family	Line Card
Q200-based line cards	88-LC0-34H14FH
	88-LC0-36FH
	88-LC0-36FH-M

Line Card Behavior with NPUs

The following table explains how the various line cards take precendence when installed from different ASIC families. The precedence followed by the system is: Q200 > Q100, where the newer generation line cards take precedence over an older generation line card.

NPU Family of Installed Line Cards	Compatibility Mode Configured?	Compatibility Mode	Router Behavior during Bootup for the Line Cards
Q200 and Q100	N	Default (Q100)	Q200 line cards boot up and operate in Q100 mode, Q100 up.
	Y	Q200	Q200 line cards boot up, Q100 line cards shut down.
	Y	Q100	All line cards boot up, Q200 line cards operate in Q100 mode.
Q200 and Q200	N	Default (Q100)	Both the Q200 line cards boot up and operate in Q100 mode.
	Y	Q200	Both the Q200 line cards boot up
Q100 and P100	N	Default (Q100)	P100 line cards boot up and operate in Q100 mode, Q100 up.
	Y	P100	P100 line cards boot up, Q100 line cards shut down.

Supported Compatibility Modes on Fabric Cards, RP Cards, and Line Cards

The following table provides details on the fabric cards (FCs), supported route processors (RPs), compatible ASIC families, supported line cards, and the ability to configure the **hw-module profile npu-compatibility** command on those line cards within a router:

Router	Route Processor	Fabric Card	Supported ASIC families to co-exist	Supported Line Cards	Configure NPU Compatibility?
Cisco 8812 Cisco 8818	8800-RP	8812-FC 8818-FC	Q100, Q200	8800-LC-48H 8800-LC-36FH 88-LC0-34H14FH 88-LC0-36FH 88-LC0-36FH-M	Yes
		8818-FC0	Q100, Q200	8800-LC-48H 8800-LC-36FH 88-LC0-34H14FH 88-LC0-36FH 88-LC0-36FH-M	Yes
	8800-RP2	8818-FC0	Q200	8800-LC-48H 8800-LC-36FH 88-LC0-34H14FH 88-LC0-36FH 88-LC0-36FH-M	Yes
	8800-RP2-S	8818-FC0	Q200	88-LC0-36FH 88-LC0-36FH-M 88-LC0-34H14FH	Yes

Router	Route Processor	Fabric Card	Supported ASIC families to co-exist	Supported Line Cards	Configure NPU Compatibility?
Cisco 8804 Cisco 8808	8800-RP	8808-FC	Q100, Q200	8800-LC-48H 8800-LC-36FH 88-LC0-34H14FH 88-LC0-36FH 88-LC0-36FH-M	Yes
		8804-FC0 8808-FC0	Q100, Q200	8800-LC-48H 8800-LC-36FH 88-LC0-34H14FH 88-LC0-36FH 88-LC0-36FH-M	Yes
	8800-RP2	8804-FC0 8808-FC0	Q200	8800-LC-48H 8800-LC-36FH 88-LC0-34H14FH 88-LC0-36FH 88-LC0-36FH-M	Yes
		8804-FC1 8808-FC1	Q200, P100	88-LC0-34H14FH 88-LC0-36FH 88-LC0-36FH-M 88-LC1-36EH	Yes
		8804-FC1 8808-FC1	P100	88-LC1-36EH 88-LC1-12TH24FH-E 88-LC1-52Y8H-EM	Yes
	8800-RP2-S	8808-FC0 8804-FC0	Q200 Default mode is Q200	88-LC0-36FH 88-LC0-36FH-M 88-LC0-34H14FH	Yes
		8808-FC1	P100 Default mode is P100	88-LC1-36EH 88-LC1-12TH24FH-E 88-LC1-52Y8H-EM Q200-based ASIC line cards	Yes
		8804-FC1			Yes

Router	Route Processor	Fabric Card	Supported ASIC families to co-exist	Supported Line Cards	Configure NPU Compatibility?
			P100 Default mode is P100	88-LC1-36EH 88-LC1-12TH24FH-E 88-LC1-52Y8H-EM Q200-based ASIC line cards	

Note Q100-based ASIC is not supported with 8800-RP2-S.

These are details of the compatibility mode for 8800-RP2 card with various fabric cards, line cards, and the supported default mode:

Table 70: 8800-RP Compatibility with Fabric Cards, Line Cards, and Supported Default Mode

Fabric Card	Fabric Card ASIC	Default ASIC	Supported Line Cards	Configure NPU Compatibility?
8808-FC 8812-FC 8818-FC	Q100	Q100	Q100-based and Q200-based	Yes You can configure the NPU mode to Q200 if you have only Q200-based line cards installed on your chassis.
8804-FC0 8808-FC0 8818-FC0	Q200	Q100	Q100-based and Q200-based	Yes You can configure the NPU mode to Q200 if you have only Q200-based line cards installed on your chassis.
8808-FC1	F100	NA	NA	NA
8804-FC1	F100	NA	NA	NA

These are details of the compatibility mode for 8800-RP2 card with various fabric cards, line cards, and the supported default mode:

Table 71: 8800-RP2 Compatibility with Fabric Cards, Line Cards, and Supported Default Mode

Fabric Card	Fabric Card ASIC	RP(8800-RP2)	Default ASIC	Supported Line Cards	Configure NPU Compatibility?
8808-FC 8812-FC 8818-FC	Q100	Not Supported	NA	NA	NA
8812-FC0	Q200	Supported	Q200	Q200-based	NA
8808-FC1	F100	Supported	P100	P100-based	Yes You can configure the NPU mode to Q200 if you have both Q200-based and P100-based line cards installed on your chassis.

These are details of the compatibility mode for 8800-RP2-S card with various fabric cards, line cards, and the supported default mode:

Table 72: 8800-RP2-S Compatibility with Fabric Cards, Line Cards, and Supported Default Mode

Fabric Card	Line Card	Default Mode
8804-FC0 8808-FC0 8818-FC0	Q200-based ASIC line cards	Q200
8808-FC1	88-LC1-36EH 88-LC1-12TH24FH-E 88-LC1-52Y8H-EM Q200-based ASIC line cards	P100
8804-FC1	88-LC1-36EH 88-LC1-12TH24FH-E 88-LC1-52Y8H-EM Q200-based ASIC line cards	P100

Configuring NPU compatibility for Line Cards

To configure a router for handling line cards of different NPU-based line cards, use the **hw-module profile npu-compatibility** command. To go back to the default mode, use the **no** form of this command.

The following are the options available in command and their descriptions:

npu-compatibility	Allows you to make a router compatible with a NPU family.
mode-name	Allows you to set the mode, such as Q100, Q200, .

The following is a configuration example:

```
Router:ios(config)#hw-module profile npu-compatibility q200
Tue Dec 7 15:06:53.697 UTC
Chassis mode will be activated after a manual reload of chassis/all line cards
Router:ios(config)#commit
Tue Dec 7 15:06:54.646 UTC
LC/0/1/CPU0:Dec 7 15:06:54.796 UTC: npu_drvr292:
%FABRIC-NPU_DRVR-3-HW_MODULE_PROFILE_NPU_COMPATIBILITY_CHASSIS_CFG_CHANGED : Please reload
chassis for the configuration to take effect
end
Router:ios(config)#end
Router:ios#
```

Running Configuration

```
RP/0/RP0/CPU0:ios# show ver
Mon Jun 27 19:25:52.947 UTC
Cisco IOS XR Software, Version 7.7.1.27I LNT
Copyright (c) 2013-2022 by Cisco Systems, Inc.

Build Information:
 Built By     : ingunawa
 Built On     : Wed Jun 01 23:50:09 UTC 2022
 Build Host   : iox-ucs-060
 Workspace    : /auto/iox-ucs-060-san1/prod/7.7.1.27I.SIT_IMAGE/8000/ws
 Version      : 7.7.1.27I
 Label        : 7.7.1.27I

cisco 8000 (VXR)
cisco 8808 (VXR) processor with 32GB of memory
ios uptime is 3 minutes
Cisco 8808 8-slot Chassis

RP/0/RP0/CPU0:ios#

RP/0/RP0/CPU0:ios# conf
Mon Jun 27 19:24:40.621 UTCRP/0/RP0/CPU0:ios(config)#hw-module profile npu-compatibility ?
  P100  Use P100 for Chassis mode
  Q100  Use Q100 for Chassis mode
  Q200  Use Q200 for Chassis mode
```

Verification

```
RP/0/RP0/CPU0:ios# show hw-module profile npu-compatibility matrix
Wed Nov 17 02:00:28.652 UTC
Node              Card Type                NPU Type
-----------------------------------------------------------
0/0/CPU0          88-LC0-36FH              Q200
0/1/CPU0          88-LC1-36EH              P100
0/2/CPU0          88-LC1-36EH              P100
0/3/CPU0          88-LC1-36EH              P100

            Compatibility       Compatibility       Compatibility       Compatibility
  Compatibility       Compatibility       Compatibility
NPU Type   Mode Q100            Mode Q200            Mode G100            Mode P100
     Mode A100            Mode K100            Mode F100
```

```
------------------------------------------------------------------------
Q100        Compatible          Not Compatible      Not Compatible      Not Compatible
  Not Compatible      Not Compatible      Not Compatible
Q200        Compatible          Compatible          Not Compatible      Not Compatible
  Not Compatible      Not Compatible      Not Compatible
G100        Not Compatible      Compatible          Compatible          Not Compatible
  Not Compatible      Not Compatible      Not Compatible
P100        Not Compatible      Not Compatible      Not Compatible      Not Compatible
  Not Compatible      Not Compatible      Not Compatible
A100        Not Compatible      Not Compatible      Not Compatible      Not Compatible
  Not Compatible      Not Compatible      Not Compatible
K100        Not Compatible      Not Compatible      Not Compatible      Not Compatible
  Not Compatible      Not Compatible      Not Compatible
F100        Not Compatible      Not Compatible      Not Compatible      Not Compatible
  Not Compatible      Not Compatible      Not Compatible
Default mode : P100
RP/0/RP0/CPU0:ios#
```

Storage Media Sanitization

Table 73: Feature History Table

Feature Name	Release Information	Feature Description
Storage Media Sanitization	Release 7.3.4	To comply with NIST SP 800-88 guidelines for Media Sanitization, it is important that your organization ensures that no easily reconstructible data is stored in the router and associated devices after it has left the control of your organization or is no longer protected by confidentiality categorization. With this feature, you can erase and overwrite any sensitive data, configuration, or keys present in the route processor or line card, ensuring media sanitization and preventing unauthorized data retrieval.

When you identify an RP or line card for RMA, or you require to ship it outside your organization, a service personnel may not be available on-site to remove the card immediately. However, you can reset your RP or line card to erase customer-sensitive data and let the RP or line card remain in the slot.

Guidelines and restrictions for factory reset functionality

These guidelines and restrictions apply to factory reset functionality on routers:

- You cannot initiate factory reset if the entire system is down or if no active RP is booted to IOS XR OS.

- We recommend using **factory-reset** without performing **commit replace** for securely removing the files in the misc/config folder.
- The RP or line card shuts down automatically if the factory reset takes more than 30 minutes, you can perform the factory reset again. The console displays this log message during automatic shutdown:

```
[ TIME ] Timed out starting Power-Off.
[ !!  ] Forcibly powering off as result of failure.
```

- If your router has dual RPs, and to perform the factory reset on both the RPs, first reset the standby RP from the active RP. After the reset is complete, you can then reset the active RP.
- The factory reset operation does not completely wipe out the data on the hard disk of the active RP because the disaster recovery partitioning is not removed.

Perform factory reset on a router

Factory reset functionality supports these scenarios:

- Reload option: resets the router and reboots it
- Shutdown option: resets the router and shuts it down
- Location option: applies the reset operation to specific locations such as individual line card (LC) or route processor (RP)

Use the **factory-reset** command for erasing these folders of RP or LC :

- /misc/disk1
- /misc/scratch
- /var/log
- /misc/config

Before you begin

- Device must be operational and booted to IOS XR OS to initiate factory reset.
- Ensure that there is no immediate requirement for the router after the operation, as it involves complete data removal and shutdown.
- Take a backup of the router data as a precautionary measure.

Procedure

Step 1 Initiate factory reset process on the router CLI.

- Reload option:

```
Router#factory-reset reload location 0/RP1/CPU0
Tue Mar 11 11:18:43.222 UTC
Performing factory-reset may affect the stability of the system. Re-imaging maybe required to
```

```
recover. Continue?
[confirm]
```

- Shutdown option:

```
Router#factory-reset shutdown location 0/RP1/CPU0
Tue Mar 11 11:18:43.222 UTC
Performing factory-reset may affect the stability of the system. Re-imaging maybe required to
recover. Continue?
[confirm]
```

The factory reset command with the **location** *location-id* option erases customer-sensitive data in the specified location.

Step 2 Check the system logs to confirm that the factory reset process is completed.

Example:

The logs are displayed on the console port of the node where the reset is performed.

Step 3 Verify that the factory reset process is completed.

Example:

This example shows how to verify the factory reset process that is performed with the **shutdown** option:

```
Router#show shelfmgr history events location 0/RP1/CPU0
Tue Mar 15 01:45:56.402 UTC
NODE NAME     : 0/RP1/CPU0
CURRENT STATE : CARD_SHUT_POWERED_OFF
TIME STAMP    : Mar 15 2022 01:44:47
--------------------------------------------------------------------------------
DATE        TIME (UTC)  EVENT                    STATE
--------------------------------------------------------------------------------
Mar 15 2022 01:44:47    ev_powered_off           CARD_SHUT_POWERED_OFF
Mar 15 2022 01:44:47    transient_condition      CARD_SHUTDOWN
Mar 15 2022 01:44:47    ev_check_card_down_reaso CHECKING_DOWN_REASON
Mar 15 2022 01:44:47    ev_os_halted             OS_HALTED
Mar 15 2022 01:44:43    ev_factory_reset_done    FACTORY_RESET_DONE
Mar 15 2022 01:33:16    ev_factory_reset_started FACTORY_RESET_IN_PROGRESS
Mar 15 2022 01:33:11    ev_os_halting            OS_HALT_IN_PROGRESS
Mar 15 2022 01:33:10    ev_xr_shut               START_OS_HALT
Mar 15 2022 01:33:09    ev_ack_ok                STATE_NOT_CHANGED
Mar 15 2022 01:33:09    ev_graceful_shut         CARD_SHUTDOWN_IN_PROGRESS
Mar 15 2022 00:55:31    ev_xr_ready              XR_RUN
```

This example shows how to verify the factory reset process that is performed with the **reload** option:

```
Router#show shelfmgr history events location 0/RP0/CPU0
Tue Mar 15 01:45:56.402 UTC
NODE NAME     : 0/RP0/CPU0
CURRENT STATE : CARD_SHUT_POWERED_OFF
TIME STAMP    : Mar 15 2022 01:44:47
--------------------------------------------------------------------------------
DATE        TIME (UTC)  EVENT                    STATE
--------------------------------------------------------------------------------
Jun 29 2022 13:48:34    ev_xr_ready              XR_RUN
Jun 29 2022 13:48:10    ev_card_info_rcvd        CARD_INFO_RCVD
Jun 29 2022 13:47:52    ev_xr_init               XR_INITIALIZING
Jun 29 2022 13:47:44    ev_kernel_booting        STATE_NOT_CHANGED
Jun 29 2022 13:47:14    ev_kernel_booting        KERNEL_BOOTING
Jun 29 2022 13:46:53    ev_unmapped_event        STATE_NOT_CHANGED
Jun 29 2022 13:46:53    ev_bios_started          BIOS_STARTED
```

```
Jun 29 2022 13:46:51    ev_bios_ready            BIOS_READY
Jun 29 2022 13:46:10    ev_unmapped_event        STATE_NOT_CHANGED
Jun 29 2022 13:46:10    ev_powered_on            CARD_POWERED_ON
Jun 29 2022 13:46:05    ev_card_reset_done       CARD_RESET
Jun 29 2022 13:46:05    transient_condition      CARD_RESETTING
Jun 29 2022 13:46:05    ev_check_card_down_reaso CHECKING_DOWN_REASON
Jun 29 2022 13:46:05    ev_os_halted             OS_HALTED
Jun 29 2022 13:45:50    ev_factory_reset_done    FACTORY_RESET_DONE
Jun 29 2022 13:34:09    ev_factory_reset_started FACTORY_RESET_IN_PROGRESS
Jun 29 2022 13:33:59    ev_os_halting            OS_HALT_IN_PROGRESS
Jun 29 2022 13:33:58    ev_xr_shut               START_OS_HALT
Jun 29 2022 13:33:56    ev_graceful_reload       CARD_SHUTDOWN_IN_PROGRESS
Jun 29 2022 09:18:43    ev_xr_ready              XR_RUN
Jun 29 2022 09:17:37    ev_card_info_rcvd        CARD_INFO_RCVD
Jun 29 2022 09:17:32    ev_powered_on            CARD_POWERED_ON
Jun 29 2022 09:17:31    init                     CARD_DISCOVERED
```

Excluding Sensitive Information in Show Running Configurations Output

Table 74: Feature History Table

Feature Name	Release Information	Feature Description
Excluding Sensitive Information in Show Running Configurations Command Output	Release 7.5.4	You can now exclude sensitive information such as strings, usernames, passwords, comments, or IP addresses within the **show running-configuration** command output by enabling sanitization on the nonvolatile generation (NVGEN) process. With this feature, you can achieve better data protection to prevent cybersecurity risks compared to regular router algorithms. This feature introduces the **nvgen default-sanitize** command.

The **show running configuration** command uses the nonvolatile generation (NVGEN) process in IOS-XR software to collect configuration information from every system component and construct a running configuration file to create its output. However, this file may contain sensitive information, including usernames, passwords, and IP addresses, which could pose a security threat when obfuscation algorithms in the router are weak compared to modern cryptographic standards.

In this feature, you can mask the following types of sensitive information in the show running configurations:

- Strings
- Usernames

- Passwords
- Comments
- IP Addresses

On enabling the sanitization in show running configurations, the NVGEN process replaces the corresponding information with **<removed>** string. For example, if you enable sanitization for IP Addresses, the show running configuration includes the **<removed>** string in place of all the IP Addresses in the output.

Sanitizing Strings

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize strings
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize strings
!
```

Verification

```
Router# show run int Hu0/2/0/4
interface HundredGigE0/2/0/4
 ! This is comment 1
 description <removed>
 !
```

Sanitizing Usernames

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize usernames
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize usernames
!
```

Verification

```
Router# show run username test
username <removed>
 group root-lr
 password 7 172864HJWBJHBCWH
!
```

Sanitizing Passwords

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize passwords
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize passwords
!
```

Verification

```
Router# show run username test
username test
 group root-lr
 password 7 <removed>
!
```

Sanitizing Comments

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize comments
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize comments
!
```

Verification

```
Router# show run int Hu0/2/0/4
interface HundredGigE0/2/0/4
 ! <comments removed>
 description This is bundle member
 !
```

Sanitizing IP Addresses

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize ipaddrs
Router:(config)# commit
```

Verification

```
Router# show run int Hu0/2/0/4
interface HundredGigE0/2/0/4
 ! This is comment 1
 description This is bundle member
  ipv4 address <removed> <removed>
!
```