CONTENTS

PREFACE

Preface

Changes to this document

Obtaining Documentation and Submitting a Service Request

CHAPTER 1

New and Changed Feature Information

Interface and Hardware Component Features Added or Modified in IOS XR Release 7.1.x

CHAPTER 2

Preconfiguring Physical Interfaces

Physical Interface Preconfiguration Overview

Prerequisites for Preconfiguring Physical Interfaces

Benefits of Interface Preconfiguration

How to Preconfigure Physical Interfaces

Information About Preconfiguring Physical Interfaces

Use of the Interface Preconfigure Command

CHAPTER 3

Advanced Configuration and Modification of the Management Ethernet Interface

Prerequisites for Configuring Management Ethernet Interfaces

How to Perform Advanced Management Ethernet Interface Configuration

Configuring a Management Ethernet Interface

IPv6 Stateless Address Auto Configuration on Management Interface

Modifying the MAC Address for a Management Ethernet Interface

Verifying Management Ethernet Interface Configuration

Information About Configuring Management Ethernet Interfaces

Dense Wavelength Division Multiplexing Tunable Optics

CHAPTER 4

Configuring Ethernet Interfaces
Configure UDLD 99
Y.1731 Performance Monitoring 101
  Two-Way Delay Measurement 102
    Configuring Two-Way Delay Measurement 102
Synthetic Loss Measurement 108
  Configuring Synthetic Loss Measurement 109
Bit Error Rate 114
Configuration Examples for Ethernet OAM 115
  Configuration Examples for EOAM Interfaces 115
    Configuring an Ethernet OAM Profile Globally: Example 115
    Configuring Ethernet OAM Features on an Individual Interface: Example 115
    Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example 116
  Clearing Ethernet OAM Statistics on an Interface: Example 116
  Enabling SNMP Server Traps on a Router: Example 116
Configuration Examples for Ethernet CFM 117
  Ethernet CFM Domain Configuration: Example 117
  Ethernet CFM Service Configuration: Example 117
  Flexible Tagging for an Ethernet CFM Service Configuration: Example 117
  Continuity Check for an Ethernet CFM Service Configuration: Example 117
  MIP Creation for an Ethernet CFM Service Configuration: Example 117
  Cross-check for an Ethernet CFM Service Configuration: Example 118
  Other Ethernet CFM Service Parameter Configuration: Example 118
  MEP Configuration: Example 118
  Ethernet CFM Show Command: Examples 118
  AIS for CFM Configuration: Examples 121
  AIS for CFM Show Commands: Examples 122
    show ethernet cfm interfaces ais Command: Example 122
    show ethernet cfm local meps Command: Examples 122
    show ethernet cfm local meps detail Command: Example 124
CFM Adaptive Bandwidth Notifications 125
  Bandwidth Notification Messages 125
  Restrictions for CFM Bandwidth Notifications 126
  Bandwidth Reporting 127
## Contents

- **Configure ERSPAN Egress Rate Limit** 193
- **SPAN over Pseudo-Wire** 195
  - Limitations 195
  - Configuring SPAN over Pseudo-Wire 195
  - Verifying SPAN over Pseudo-Wire 196
- **SPAN to File** 198
  - Action Commands for SPAN to File 198
  - Configuring SPAN to File 199
- **Introduction to File Mirroring** 201
  - Limitations 201
  - Configure File Mirroring 201

### CHAPTER 9

- **Configuring Virtual Loopback and Null Interfaces** 203
  - Information About Configuring Virtual Interfaces 203
  - Virtual Loopback Interface Overview 203
  - Prerequisites for Configuring Virtual Interfaces 204
  - Configuring Virtual Loopback Interfaces 204
  - Null Interface Overview 206
  - Configuring Null Interfaces 206
  - Configuring Virtual IPv4 Interfaces 208

### CHAPTER 10

- **Configuring GRE Tunnels** 211
  - Configuring GRE Tunnels 211
  - Single Pass GRE Encapsulation Allowing Line Rate Encapsulation 212
    - Configuration 213
    - Running Configuration 216
    - Verification 219

### CHAPTER 11

- **Configuring Generic UDP Encapsulation** 223
  - Understand Generic UDP Encapsulation 223
  - Restrictions 225
  - Configure GUE 225

### CHAPTER 12

- **Configuring Controllers** 229
Preface

The Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers provides information and procedures related to router interface and hardware configuration.

The preface contains the following sections:

- Changes to this document, on page xi
- Obtaining Documentation and Submitting a Service Request, on page xi

Changes to this document

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 2020</td>
<td>Initial release of this document.</td>
</tr>
<tr>
<td>April 2020</td>
<td>Republished for Release 7.1.15.</td>
</tr>
<tr>
<td>August 2020</td>
<td>Republished for Release 7.1.2.</td>
</tr>
</tbody>
</table>

Obtaining Documentation and Submitting a Service Request

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

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

This table summarizes the new and changed feature information for the Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, and tells you where they are documented.

- Interface and Hardware Component Features Added or Modified in IOS XR Release 7.1.x, on page 1

Interface and Hardware Component Features Added or Modified in IOS XR Release 7.1.x

Table 2: New and Changed Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFM Adaptive Bandwidth Notifications</td>
<td>Connectivity Fault Management (CFM) extension is used to send Bandwidth Notifications Messages (BNM) to Maintenance Endpoints (MEPs) on the corresponding interface on the head-end router.</td>
<td>Release 7.1.1</td>
<td>For more information about the feature, see the Configuring Ethernet OAM chapter in the Interface and Hardware Component Command Reference for Cisco NCS 5500 and NCS 540 and NCS 560 Series Routers.</td>
</tr>
<tr>
<td>Generic UDP Encapsulation</td>
<td>Generic UDP Encapsulation (GUE) is a UDP-based network encapsulation protocol that encapsulates IPv4 and IPv6 packets.</td>
<td>Release 7.1.2</td>
<td>Configuring Generic UDP Encapsulation, on page 223</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Introduced in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SPAN to File</td>
<td>SPAN to File feature is an extension of the pre-existing SPAN feature to allow network packets to be mirrored to a file instead of an interface, so that they can be analyzed later. The file format is PCAP, so that it can be easily used with tools such as tcpdump or wireshark.</td>
<td>Release 7.1.2</td>
<td>For more information about the feature, see the <em>Configuring Traffic Mirroring</em> chapter.</td>
</tr>
<tr>
<td>File Mirroring</td>
<td>File mirroring feature enables the router to copy files or directories automatically from <code>/harddisk:/mirror</code> location in active RP to <code>/harddisk:/mirror</code> location in standby RP or RSP without user intervention or EEM scripts.</td>
<td>Release 7.1.2</td>
<td>For more information about the feature, see the <em>Configuring Traffic Mirroring</em> chapter.</td>
</tr>
</tbody>
</table>
Preconfiguring Physical Interfaces

This module describes the preconfiguration of physical interfaces.

Preconfiguration is supported for these types of interfaces and controllers:

- 100-Gigabit Ethernet
- Management Ethernet

Preconfiguration allows you to configure line cards before they are inserted into the router. When the cards are inserted, they are instantly configured. The preconfiguration information is created in a different system database tree, rather than with the regularly configured interfaces. That database tree is known as the preconfiguration directory on the route processor.

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

---

**Note**

One Gigabit Ethernet interface is not supported. Only physical interfaces can be preconfigured.

**Note**

Eight quadrature amplitude modulation (8QAM) requires V2 (or higher) CFP2 version and 5.23 (or higher) firmware.

**Note**

From Cisco IOS XR Release 6.3.2, a six-seconds delay is introduced in error propagation from the driver to DPA for the MACSec line card and Oldcastle platforms. As a result, the BER algorithm on these platforms knows the error with a delay of 6 seconds.

---

- Physical Interface Preconfiguration Overview, on page 4
- Prerequisites for Preconfiguring Physical Interfaces, on page 4
- Benefits of Interface Preconfiguration, on page 4
- How to Preconfigure Physical Interfaces, on page 5
- Information About Preconfiguring Physical Interfaces, on page 6
Physical Interface Preconfiguration Overview

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

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

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

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

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

---

**Prerequisites for Preconfiguring Physical Interfaces**

Before preconfiguring physical interfaces, ensure that this condition is met:

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

---

**Benefits of Interface Preconfiguration**

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

Another advantage of performing a preconfiguration is that during a cards replacement, when the cards is removed, you can still see the previous configuration and make modifications.
How to Preconfigure Physical Interfaces

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

**Step 1**
configure

*Example:*

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

Enters global configuration mode.

**Step 2**
interface preconfigure type interface-path-id

*Example:*

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

Enters interface preconfiguration mode for an interface, where *type* specifies the supported interface type that you want to configure and *interface-path-id* specifies the location where the interface will be located in *rack/slot/module/port* notation.

**Step 3**
Use one of the following commands:

- ipv4 address ip-address subnet-mask
- ipv4 address ip-address/prefix

*Example:*

```
RP/0/RP0/CPU0:router(config-if-pre)#ipv4 address 192.168.1.2/31
```

Assigns an IP address and mask to the interface.

**Step 4**
Configure additional interface parameters, as described in this manual in the configuration chapter that applies to the type of interface that you are configuring.

**Step 5**
end or commit best-effort
Example:

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

or

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

Saves configuration changes.

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

Step 6  show running-config

Example:

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

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

Information About Preconfiguring Physical Interfaces

From Cisco IOS XR Release 7.0.2, the NC57-18DD-SE follows the following port mapping:

- Port number 0-17 (nine pairs) and 24-29 (three pairs): They together drive 400G mode. This means that if the top port is in 400G mode, the bottom port is unusable. These ports are retimer ports.
- Port number 18-23 (six ports): They are direct connected ports and are individually capable of 400G mode.

There’s a limitation for ports 0, 1 and 14, 15. You have to insert modules of similar speed (40G or 100G) into these pairs of ports. For example, if you insert 40G module in port 0, then 40G module must be inserted in port 1.

For 400G-only mode, the ports to be used are 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 19, 20, 21, 22, 23, 24, 26, and 28.

For detailed information on port mapping and usage, see the figure NC57-18DD-SE Line Card in chapter NCS 5500 Series Modular Router Overview of Hardware Installation Guide for Cisco NCS 5500 Series Modular Routers guide.

To control the interfaces which are created, use the `hw-module port-range mode` command with the following modes:

- 40-100: This is the default port mode. Two ports are created in 100G mode by default. Online Insertion and Removal (OIR) to 40G creates the 40G port. Assume both ports to be similar to J/J+ ports.
- 400: The first port created is 400G. No port is created for the bottom port.
- 2x100: For 2x100 mode. This supports QDD-2X100-LR4 optics.

Port range can be in the form of n to n+1. Example: 0, 1 or 6, 7. The port range is valid for ports 0-17 and 24-29. To configure a port with 400G rate:

```
RP/0/RP0/CPU0:router(config)#hw-module port-range 0 1 location 0/3/CPU0 mode 400
RP/0/RP0/CPU0:router(config)#commit
```

```
LC/0/3/CPU0:Feb 6 03:23:12.923 UTC: ifmgr[281]: %PKT_INFRA-LINK-3-UPDOWN : Interface HundredGigE0/3/0/0, changed state to Down
LC/0/3/CPU0:Feb 6 03:23:13.548 UTC: ifmgr[281]: %PKT_INFRA-LINK-3-UPDOWN : Interface HundredGigE0/3/0/1, changed state to Down
RP/0/RP0/CPU0:router(config)#end
RP/0/RP0/CPU0:router#show ipv4 int br location 0/3/CPU0
```

```
Wed Feb 6 03:26:07.935 UTC

Interface     IP-Address        Status      Protocol Vrf-Name
FourHundredGigE0/3/0/0 unassigned Shutdown Down default
HundredGigE0/3/0/2 unassigned Shutdown Down default
HundredGigE0/3/0/3 unassigned Shutdown Down default
HundredGigE0/3/0/4 unassigned Shutdown Down default
HundredGigE0/3/0/5 unassigned Shutdown Down default
HundredGigE0/3/0/6 unassigned Shutdown Down default
```

To change a port mode:

```
RP/0/RP0/CPU0:router#conf
Thu Jan 9 05:13:02.853 UTC
RP/0/RP0/CPU0:router(config)#hw-module port-range 2 3 location 0/3/CPU0 mode 2x100
RP/0/RP0/CPU0:router(config)#commit
Thu Jan 9 05:13:11.411 UTC
LC/0/3/CPU0:Jan 9 05:13:11.469 UTC: optics_driver[196]: %PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR :PORTMODE SPEED MISMATCH :CLEAR :0/3/CPU0: Optics0/3/0/3
```
Use the following commands for the newly configured image:

```
hw-module port-range 0 1 location 0/6/CPU0 mode 400
hw-module port-range 2 3 location 0/6/CPU0 mode 400
hw-module port-range 4 5 location 0/6/CPU0 mode 400
hw-module port-range 6 7 location 0/6/CPU0 mode 400
hw-module port-range 8 9 location 0/6/CPU0 mode 400
hw-module port-range 10 11 location 0/6/CPU0 mode 400
hw-module port-range 12 13 location 0/6/CPU0 mode 400
hw-module port-range 14 15 location 0/6/CPU0 mode 400
hw-module port-range 16 17 location 0/6/CPU0 mode 400
hw-module port-range 24 25 location 0/6/CPU0 mode 400
hw-module port-range 26 27 location 0/6/CPU0 mode 400
hw-module port-range 28 29 location 0/6/CPU0 mode 400
hw-module port-range 0 1 location 0/6/CPU0 mode 2x100
hw-module port-range 2 3 location 0/6/CPU0 mode 2x100
hw-module port-range 4 5 location 0/6/CPU0 mode 2x100
hw-module port-range 6 7 location 0/6/CPU0 mode 2x100
hw-module port-range 10 11 location 0/6/CPU0 mode 2x100
hw-module port-range 12 13 location 0/6/CPU0 mode 2x100
hw-module port-range 14 15 location 0/6/CPU0 mode 2x100
hw-module port-range 16 17 location 0/6/CPU0 mode 2x100
hw-module port-range 24 25 location 0/6/CPU0 mode 2x100
hw-module port-range 26 27 location 0/6/CPU0 mode 2x100
hw-module port-range 28 29 location 0/6/CPU0 mode 2x100
```

To preconfigure interfaces, you must understand these concepts:

**Use of the Interface Preconfigure Command**

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

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

---

**Note**

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

---

**Note**

Specifying an interface name that already exists and is configured (or an abbreviated name like Hu0/3/0/0) is not permitted.

---
Use of the Interface Preconfigure Command
CHAPTER 3

Advanced Configuration and Modification of the Management Ethernet Interface

This module describes the configuration of Management Ethernet interfaces. Before you can use Telnet to access the router through the LAN IP address, you must set up a Management Ethernet interface and enable Telnet servers.

Note

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

- Prerequisites for Configuring Management Ethernet Interfaces, on page 11
- How to Perform Advanced Management Ethernet Interface Configuration, on page 12
- Information About Configuring Management Ethernet Interfaces, on page 18
- Dense Wavelength Division Multiplexing Tunable Optics, on page 18

Prerequisites for Configuring Management Ethernet Interfaces

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

- You have performed the initial configuration of the Management Ethernet interface.

- You know how to apply the generalized interface name specification rack/slot/module/port.

Note

For transparent switchover, both active and standby Management Ethernet interfaces are expected to be physically connected to the same LAN or switch.
How to Perform Advanced Management Ethernet Interface Configuration

This section contains the following procedures:

Configuring a Management Ethernet Interface

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1 configure
Example:

RP/0/RP0/CPU0:router# configure
Enters global configuration mode.

Step 2 interface MgmtEth interface-path-id
Example:

RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
Enters interface configuration mode and specifies the Ethernet interface name and notation rack/slot/module/port.
The example indicates port 0 on the RP card that is installed in slot 0.

Step 3 ipv4 address ip-address mask
Example:

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. The network mask can be specified in either of two ways:
  - The network mask can be a four-part dotted decimal address. For example, 255.255.0.0 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, /16 indicates that the first 16 bits of the mask are ones, and the corresponding bits of the address are network address.

**Step 4**  
 mtu *bytes*  

*Example:*

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

(Optional) Sets the maximum transmission unit (MTU) byte value for the interface. The default is 1514.

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

**Step 5**  
 no shutdown  

*Example:*

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

Removes the shutdown configuration, which removes the forced administrative down on the interface, enabling it to move to an up or down state.

**Step 6**  
 end or commit  

*Example:*

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

or

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

Saves configuration changes.

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

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 7  
show interfaces MgmtEth interface-path-id

Example:

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

(Optional) Displays statistics for interfaces on the router.

Example

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

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:router(config)# ipv4 address 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, 1000Mbit/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 appliance, 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 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
!
```

**IPv6 Stateless Address Auto Configuration on Management Interface**

Perform this task to enable IPv6 stateless auto configuration on Management interface.

**SUMMARY STEPS**

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

**DETAILED STEPS**

**Step 1**  
**configure**

**Example:**

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

Enters global configuration mode.

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

**Example:**

```
RP/0/RP0/CPU0:router(config)# interface MgmtEth0/RP0/CPU0/0
```

Enters interface configuration mode and specifies the Ethernet interface name and notation `rack/slot/module/port`. The example indicates port 0 on the RP card that is installed in slot 0.

**Step 3**  
**ipv6 address autoconfig**
Example:

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

Enable IPv6 stateless address auto configuration on the management port.

Step 4 show ipv6 interfaces interface-path-id

Example:

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

(Optional) Displays statistics for interfaces on the router.

Example

This example displays:

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:router(config)# ipv6 address autoconfig
RP/0/RP0/CPU0:router# show ipv6 interfaces gigabitEthernet 0/2/0/0

Fri Nov 4 16:48:14.372 IST
GigabitEthernet0/2/0/0 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
IPv6 is enabled, link-local address is fe80::d1:1eff:fe2b:baf
Global unicast address(es):
5::d1:1eff:fe2b:baf [AUTO CONFIGURED], subnet is 5::/64 <<<<<< auto configured address

Joined group address(es): ff02::1:ff2b:baf ff02::2 ff02::1
MTU is 1514 (1500 is available to IPv6)
ICMP redirects are disabled
ICMP unreachables are enabled
ND DAD is enabled, number of DAD attempts 1
ND reachable time is 0 milliseconds
ND cache entry limit is 1000000000
ND advertised retransmit interval is 0 milliseconds
Hosts use stateless autoconfig for addresses.
Outgoing access list is not set
Inbound common access list is not set, access list is not set
Table Id is 0xe00000000
Complete protocol adjacency: 0
Complete glean adjacency: 0
Incomplete protocol adjacency: 0
Incomplete glean adjacency: 0
Dropped protocol request: 0
Dropped glean request: 0

Modifying the MAC Address for a Management Ethernet Interface

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

SUMMARY STEPS

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

**DETAILED STEPS**

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

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

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

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

### Verifying Management Ethernet Interface Configuration

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

**Step 1**
`show interfaces MgmtEth interface-path-id`

**Example:**

```
RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/RP0/CPU0/0
```
Displays the Management Ethernet interface configuration.

**Step 2**
`show running-config interface MgmtEth interface-path-id`

**Example:**

```
RP/0/RP0/CPU0:router# show running-config interface MgmtEth 0/RP0/CPU0/0
```
Displays the running configuration.

### Information About Configuring Management Ethernet Interfaces

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

#### Dense Wavelength Division Multiplexing Tunable Optics

The Dense Wavelength-Division Multiplexing (DWDM) wavelengths of the DWDM-SFP10G-C module on the Cisco NCS 5500 Series Aggregation Services Routers is tunable. You can configure the DWDM ITU wavelengths by using the `itu channel` command in the interface configuration mode. The `itu channel` command ensures that the traffic continues to flow.

The following table contains the wavelength mapping information for the DWDM module:
<table>
<thead>
<tr>
<th>Channel</th>
<th>Frequency (THz)</th>
<th>Wavelength (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>191.35</td>
<td>1566.723</td>
</tr>
<tr>
<td>2</td>
<td>191.40</td>
<td>1566.314</td>
</tr>
<tr>
<td>3</td>
<td>191.45</td>
<td>1565.905</td>
</tr>
<tr>
<td>4</td>
<td>191.50</td>
<td>1565.496</td>
</tr>
<tr>
<td>5</td>
<td>191.55</td>
<td>1565.087</td>
</tr>
<tr>
<td>6</td>
<td>191.60</td>
<td>1564.679</td>
</tr>
<tr>
<td>7</td>
<td>191.65</td>
<td>1564.271</td>
</tr>
<tr>
<td>8</td>
<td>191.70</td>
<td>1563.863</td>
</tr>
<tr>
<td>9</td>
<td>191.75</td>
<td>1563.455</td>
</tr>
<tr>
<td>10</td>
<td>191.80</td>
<td>1563.047</td>
</tr>
<tr>
<td>11</td>
<td>191.85</td>
<td>1562.640</td>
</tr>
<tr>
<td>12</td>
<td>191.90</td>
<td>1562.233</td>
</tr>
<tr>
<td>13</td>
<td>191.95</td>
<td>1561.826</td>
</tr>
<tr>
<td>14</td>
<td>192.00</td>
<td>1561.419</td>
</tr>
<tr>
<td>15</td>
<td>192.05</td>
<td>1561.013</td>
</tr>
<tr>
<td>16</td>
<td>192.10</td>
<td>1560.606</td>
</tr>
<tr>
<td>17</td>
<td>192.15</td>
<td>1560.200</td>
</tr>
<tr>
<td>18</td>
<td>192.20</td>
<td>1559.794</td>
</tr>
<tr>
<td>19</td>
<td>192.25</td>
<td>1559.389</td>
</tr>
<tr>
<td>20</td>
<td>192.30</td>
<td>1558.983</td>
</tr>
<tr>
<td>21</td>
<td>192.35</td>
<td>1558.578</td>
</tr>
<tr>
<td>22</td>
<td>192.40</td>
<td>1558.173</td>
</tr>
<tr>
<td>23</td>
<td>192.45</td>
<td>1557.768</td>
</tr>
<tr>
<td>24</td>
<td>192.50</td>
<td>1557.363</td>
</tr>
<tr>
<td>25</td>
<td>192.55</td>
<td>1556.959</td>
</tr>
<tr>
<td>26</td>
<td>192.60</td>
<td>1556.555</td>
</tr>
<tr>
<td>27</td>
<td>192.65</td>
<td>1556.151</td>
</tr>
<tr>
<td>28</td>
<td>192.70</td>
<td>1555.747</td>
</tr>
<tr>
<td>Channel</td>
<td>Frequency (THz)</td>
<td>Wavelength (nm)</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>29</td>
<td>192.75</td>
<td>1555.343</td>
</tr>
<tr>
<td>30</td>
<td>192.80</td>
<td>1554.940</td>
</tr>
<tr>
<td>31</td>
<td>192.85</td>
<td>1554.537</td>
</tr>
<tr>
<td>32</td>
<td>192.90</td>
<td>1554.134</td>
</tr>
<tr>
<td>33</td>
<td>192.95</td>
<td>1553.731</td>
</tr>
<tr>
<td>34</td>
<td>193.00</td>
<td>1553.329</td>
</tr>
<tr>
<td>35</td>
<td>193.05</td>
<td>1552.926</td>
</tr>
<tr>
<td>36</td>
<td>193.10</td>
<td>1552.524</td>
</tr>
<tr>
<td>37</td>
<td>193.15</td>
<td>1552.122</td>
</tr>
<tr>
<td>38</td>
<td>193.20</td>
<td>1551.721</td>
</tr>
<tr>
<td>39</td>
<td>193.25</td>
<td>1551.319</td>
</tr>
<tr>
<td>40</td>
<td>193.30</td>
<td>1550.918</td>
</tr>
<tr>
<td>41</td>
<td>193.35</td>
<td>1550.517</td>
</tr>
<tr>
<td>42</td>
<td>193.40</td>
<td>1550.116</td>
</tr>
<tr>
<td>43</td>
<td>193.45</td>
<td>1549.715</td>
</tr>
<tr>
<td>44</td>
<td>193.50</td>
<td>1549.315</td>
</tr>
<tr>
<td>45</td>
<td>193.55</td>
<td>1548.915</td>
</tr>
<tr>
<td>46</td>
<td>193.60</td>
<td>1548.515</td>
</tr>
<tr>
<td>47</td>
<td>193.65</td>
<td>1548.115</td>
</tr>
<tr>
<td>48</td>
<td>193.70</td>
<td>1547.715</td>
</tr>
<tr>
<td>49</td>
<td>193.75</td>
<td>1547.316</td>
</tr>
<tr>
<td>50</td>
<td>193.80</td>
<td>1546.917</td>
</tr>
<tr>
<td>51</td>
<td>193.85</td>
<td>1546.518</td>
</tr>
<tr>
<td>52</td>
<td>193.90</td>
<td>1546.119</td>
</tr>
<tr>
<td>53</td>
<td>193.95</td>
<td>1545.720</td>
</tr>
<tr>
<td>54</td>
<td>194.00</td>
<td>1545.322</td>
</tr>
<tr>
<td>55</td>
<td>194.05</td>
<td>1544.924</td>
</tr>
<tr>
<td>56</td>
<td>194.10</td>
<td>1544.526</td>
</tr>
<tr>
<td>Channel</td>
<td>Frequency (THz)</td>
<td>Wavelength (nm)</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>57</td>
<td>194.15</td>
<td>1544.128</td>
</tr>
<tr>
<td>58</td>
<td>194.20</td>
<td>1543.730</td>
</tr>
<tr>
<td>59</td>
<td>194.25</td>
<td>1543.333</td>
</tr>
<tr>
<td>60</td>
<td>194.30</td>
<td>1542.936</td>
</tr>
<tr>
<td>61</td>
<td>194.35</td>
<td>1542.539</td>
</tr>
<tr>
<td>62</td>
<td>194.40</td>
<td>1542.142</td>
</tr>
<tr>
<td>63</td>
<td>194.45</td>
<td>1541.746</td>
</tr>
<tr>
<td>64</td>
<td>194.50</td>
<td>1541.349</td>
</tr>
<tr>
<td>65</td>
<td>194.55</td>
<td>1540.953</td>
</tr>
<tr>
<td>66</td>
<td>194.60</td>
<td>1540.557</td>
</tr>
<tr>
<td>67</td>
<td>194.65</td>
<td>1540.162</td>
</tr>
<tr>
<td>68</td>
<td>194.70</td>
<td>1539.766</td>
</tr>
<tr>
<td>69</td>
<td>194.75</td>
<td>1539.371</td>
</tr>
<tr>
<td>70</td>
<td>194.80</td>
<td>1538.976</td>
</tr>
<tr>
<td>71</td>
<td>194.85</td>
<td>1538.581</td>
</tr>
<tr>
<td>72</td>
<td>194.90</td>
<td>1538.186</td>
</tr>
<tr>
<td>73</td>
<td>194.95</td>
<td>1537.792</td>
</tr>
<tr>
<td>74</td>
<td>195.00</td>
<td>1537.397</td>
</tr>
<tr>
<td>75</td>
<td>195.05</td>
<td>1537.003</td>
</tr>
<tr>
<td>76</td>
<td>195.10</td>
<td>1536.609</td>
</tr>
<tr>
<td>77</td>
<td>195.15</td>
<td>1536.216</td>
</tr>
<tr>
<td>78</td>
<td>195.20</td>
<td>1535.822</td>
</tr>
<tr>
<td>79</td>
<td>195.25</td>
<td>1535.429</td>
</tr>
<tr>
<td>80</td>
<td>195.30</td>
<td>1535.036</td>
</tr>
<tr>
<td>81</td>
<td>195.35</td>
<td>1534.643</td>
</tr>
<tr>
<td>82</td>
<td>195.40</td>
<td>1534.250</td>
</tr>
<tr>
<td>83</td>
<td>195.45</td>
<td>1533.858</td>
</tr>
<tr>
<td>84</td>
<td>195.50</td>
<td>1533.465</td>
</tr>
</tbody>
</table>
### Dense Wavelength Division Multiplexing Tunable Optics

<table>
<thead>
<tr>
<th>Channel</th>
<th>Frequency (THz)</th>
<th>Wavelength (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>195.55</td>
<td>1533.073</td>
</tr>
<tr>
<td>86</td>
<td>195.60</td>
<td>1532.681</td>
</tr>
<tr>
<td>87</td>
<td>195.65</td>
<td>1532.290</td>
</tr>
<tr>
<td>88</td>
<td>195.70</td>
<td>1531.898</td>
</tr>
<tr>
<td>89</td>
<td>195.75</td>
<td>1531.507</td>
</tr>
<tr>
<td>90</td>
<td>195.80</td>
<td>1531.116</td>
</tr>
<tr>
<td>91</td>
<td>195.85</td>
<td>1530.725</td>
</tr>
<tr>
<td>92</td>
<td>195.90</td>
<td>1530.334</td>
</tr>
<tr>
<td>93</td>
<td>195.95</td>
<td>1529.944</td>
</tr>
<tr>
<td>94</td>
<td>196.00</td>
<td>1529.553</td>
</tr>
<tr>
<td>95</td>
<td>196.05</td>
<td>1529.163</td>
</tr>
<tr>
<td>96</td>
<td>196.10</td>
<td>1528.773</td>
</tr>
</tbody>
</table>

---

**Note**

CHAPTER 4

Configuring Ethernet Interfaces

This module describes the configuration of Ethernet interfaces.

The following distributed ethernet architecture delivers network scalability and performance, while enabling service providers to offer high-density, high-bandwidth networking solutions.

- 10-Gigabit
- 40-Gigabit
- 100-Gigabit

These solutions are designed to interconnect the router with other systems in point-of-presence (POP)s, including core and edge routers and Layer 2 and Layer 3 switches.

Restrictions

Router does not support configuration of the static mac address.

- Configuring Gigabit Ethernet Interfaces, on page 23
- Information About Configuring Ethernet, on page 27
- Link Layer Discovery Protocol (LLDP), on page 34
- Dense Wavelength Division Multiplexing Tunable Optics, on page 37
- Priority Flow Control (PFC), on page 48
- How to Configure Interfaces in Breakout Mode, on page 51

Configuring Gigabit Ethernet Interfaces

Use this procedure to create a basic Ethernet interface configuration.

SUMMARY STEPS

1. show version
2. show interfaces [GigE | TenGigE | | HundredGigE] interface-path-id
3. configure
4. interface [GigE | TenGigE | | HundredGigE] interface-path-id
5. ipv4 address ip-address mask
6. mtu bytes
7. no shutdown
8. end or commit
9. show interfaces [GigE TenGigE HundredGigE] interface-path-id

DETAILED STEPS

---

**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 [GigE | TenGigE | | | HundredGigE] 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**

`interface [GigE | TenGigE | | | HundredGigE] 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:

- GigE
- 10GigE
- 100GigE

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

**Step 5**

`ipv4 address ip-address mask`

**Example:**

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

Assigns an IP address and subnet mask to the interface.

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

Step 6  mtu bytes
  Example:
  
  RP/0/RP0/CPU0:router(config-if)# mtu 2000
  (Optional) Sets the MTU value for the interface.
  • The configurable range for MTU values is 1514 bytes to 9646 bytes.
  • The default is 1514 bytes for normal frames and 1518 bytes for 802.1Q tagged frames.

Step 7  no shutdown
  Example:
  
  RP/0/RP0/CPU0:router(config-if)# no shutdown
  Removes the shutdown configuration, which forces an interface administratively down.

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

Step 9  show interfaces [GigE TenGigE HundredGigE ] 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 2000
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
Information About Configuring Ethernet

This section provides the following information sections:

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 100-Gigabit Ethernet line card.

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 by entering the no shutdown command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration File Entry</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTU</td>
<td>mtu</td>
<td>• 1514 bytes for normal frames</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1518 bytes for 802.1Q tagged frames.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1522 bytes for Q-in-Q frames.</td>
</tr>
<tr>
<td>MAC address</td>
<td>mac address</td>
<td>Hardware burned-in address (BIA)</td>
</tr>
</tbody>
</table>

Network Interface Speed

1Gig interfaces connected through copper or fiber cable can have interface speed of either 100 Mbps or 1000 Mbps. This is applicable on 1Gig interface with a 1000Base-T module (GLC-TE). By default 1G interface has following capabilities:

- Speed—1000 Mbps for fiber cable and autonegotiate for copper cable
- Duplex—Full
- Pause—Receive Part (RX) and Transmit Part (TX)

The copper and fiber cables have same default values as mentioned above but autonegotiation is default for copper cable.
The speed can either be configured or set to autonegotiate with remote end interface. When in autonegotiation mode, an interface is capable of negotiating the speed of 100 Mbps or 1000 Mbps depending on the speed at the remote end interface; and other parameters such as full duplex and pause are also autonegotiated.

Autonegotiation is an optional function of the Fast Ethernet standard that enables devices to automatically exchange information over a link about speed and duplex abilities. Autonegotiation is very useful for ports where devices with different capabilities are connected and disconnected on a regular basis.

### Configuring Network Interface Speed

You can configure the network interface speed by using one of the following methods:

- Using the `speed` command
- Using the `negotiation auto` command
- Using both `speed` and `negotiation auto` command

---

**Note**

Cisco recommends to configure network interface speed in autonegotiation mode.

---

### Using the `speed` command

When you configure the speed of the network interface (1G) using the `speed` command, the interface speed is forced to the configured speed by limiting the speed value of the auto negotiated parameter to the configured speed.

This sample configuration forces the Gig interface speed to 100Mbps.

---

**Note**

The interface speed at remote end is also set to 100Mbps.

```bash
#configuration
(config)#interface GigabitEthernet 0/0/0/31
(config-if)#speed 100
(config-if)#commit
(config-if)#end
```

Use the `show controller GigE` and `show interface GigE` commands to verify if the speed is configured to 100Mbps and autonegotiation is disabled:

```bash
#show controllers GigabitEthernet 0/0/0/31
Operational data for interface GigabitEthernet0/0/0/31:
    State:
        Administrative state: enabled
        Operational state: Up
    LED state: Green On
    Phy:
        Media type: Four-pair Category 5 UTP PHY, full duplex
        Optics:
            Vendor: CISCO
            Part number: SBCU-5740ARZ-CS1
            Serial number: AVC194525HW
            Wavelength: 0 nm
```
Digital Optical Monitoring:
Transceiver Temp: 0.000 C
Transceiver Voltage: 0.000 V

Alarms key: (H) Alarm high, (h) Warning high
(L) Alarm low, (l) Warning low

<table>
<thead>
<tr>
<th>Lane</th>
<th>Wavelength (nm)</th>
<th>Tx Power (dBm)</th>
<th>Rx Power (dBm)</th>
<th>Laser Bias (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>n/a</td>
<td>0.0</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
</tbody>
</table>

DOM alarms:
No alarms

<table>
<thead>
<tr>
<th>Alarm</th>
<th>Thresholds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>Transceiver Temp (C):</td>
<td>0.000</td>
</tr>
<tr>
<td>Transceiver Voltage (V):</td>
<td>0.000</td>
</tr>
<tr>
<td>Laser Bias (mA):</td>
<td>0.000</td>
</tr>
<tr>
<td>Transmit Power (mW):</td>
<td>1.000</td>
</tr>
<tr>
<td>Transmit Power (dBm):</td>
<td>0.000</td>
</tr>
<tr>
<td>Receive Power (mW):</td>
<td>1.000</td>
</tr>
<tr>
<td>Receive Power (dBm):</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Statistics:
FEC:
Corrected Codeword Count: 0
Uncorrected Codeword Count: 0

MAC address information:
Operational address: 0035.1a00.e67c
Burnt-in address: 0035.1a00.e62c

Autonegotiation disabled.

Operational values:
**Speed: 100Mbps**

*Gig interface speed is set to 100Mbps *
Duplex: Full Duplex
Flowcontrol: None
Loopback: None (or external)
MTU: 1514
MRU: 1514
Forward error correction: Disabled

#show interfaces GigabitEthernet 0/0/0/31

*GigabitEthernet0/0/0/31 is up, line protocol is up*

Interface state transitions: 7
Hardware is GigabitEthernet, address is 0035.1a00.e62c (bia 0035.1a00.e62c)
Internet address is Unknown
MTU 1514 bytes, BW 100000 Kbit (Max: 100000 Kbit)
reliability 255/255, txload 0/255, rxload 0/255
Encapsulation ARPA,
Full-duplex, 100Mb/s, TFD, 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 00:00:30
Last input 00:00:00, output 00:00:00
Last clearing of "show interface" counters never
30 second input rate 1000 bits/sec, 1 packets/sec
30 second output rate 0 bits/sec, 1 packets/sec
90943 packets input, 11680016 bytes, 0 total input drops
0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 90943 multicast packets
In the above show output you will observe that the state of the GigabitEthernet0/0/0/31 is up, and line protocol is up. This is because the speed at both ends is 100Mbps.

Using the negotiation auto command

When you configure the network interface speed using `negotiation auto` command, the speed is autonegotiated with the remote end interface. This command enhances the speed capability to 100M or 1G to be negotiated with the peer.

This sample configuration sets the interface speed to autonegotiate:

```
#configuration
(config)#interface GigabitEthernet 0/0/0/31
(config-if)#negotiation auto
(config-if)#commit
(config-if)#end
```

Use the `show controller GigE` and `show interface GigE` commands to verify if the speed is autonegotiated:

```
#show interfaces GigabitEthernet 0/0/0/31
GigabitEthernet0/0/0/31 is up, line protocol is up
Interface state transitions: 10
Hardware is GigabitEthernet, address is 0035.1a00.e62c (bia 0035.1a00.e62c)
Internet address is Unknown
MTU 1514 bytes, BW 100000 Kbit (Max: 100000 Kbit)
reliability 255/255, txload 0/255, rxload 0/255
Encapsulation ARPA,
Full-duplex, 100Mb/s, TFD, link type is autonegotiation
output flow control is off, input flow control is off
Carrier delay (up) is 10 msec
loopback not set,
Last link flapped 00:00:01
Last input 00:00:00, output 00:00:00
Last clearing of "show interface" counters never
30 second input rate 1000 bits/sec, 1 packets/sec
30 second output rate 0 bits/sec, 0 packets/sec
91005 packets input, 11687850 bytes, 0 total input drops
9 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 91005 multicast packets
0 runts, 0 giants, 0 throttles, 0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
61307 packets output, 4350024 bytes, 0 total output drops
```

Note: The interface speed at remote end is set to 100Mbps.

Note: Default setting for auto-negotiation varies with different platforms under the NCS 5500 family. On NCS 540 and NCS 55A2, 100G auto-negotiation is enabled by default.
Output 0 broadcast packets, 8668 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
15 carrier transitions

In the above show output you see that GigabitEthernet0/0/0/31 is up, and line protocol is up.

```
#show controllers GigabitEthernet 0/0/0/31
Operational data for interface GigabitEthernet0/0/0/31:

State:
  Administrative state: enabled
  Operational state: Up
  LED state: Green On

Phy:
  Media type: Four-pair Category 5 UTP PHY, full duplex
  Optics:
    Vendor: CISCO
    Part number: SBCU-5740ARZ-CS1
    Serial number: AVC194525HW
    Wavelength: 0 nm

Digital Optical Monitoring:
  Transceiver Temp: 0.000 C
  Transceiver Voltage: 0.000 V

Alarms key: (H) Alarm high, (h) Warning high
             (L) Alarm low, (l) Warning low

<table>
<thead>
<tr>
<th>Lane</th>
<th>Wavelength (nm)</th>
<th>Tx Power (dBm)</th>
<th>Rx Power (mW)</th>
<th>Laser Bias (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>n/a</td>
<td>0.0</td>
<td>1.0000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

DOM alarms:
  No alarms

Alarm Thresholds:
  - Transceiver Temp (C): 0.000 0.000 0.000 0.000
  - Transceiver Voltage (V): 0.000 0.000 0.000 0.000
  - Laser Bias (mA): 0.000 0.000 0.000 0.000
  - Transmit Power (mW): 1.000 1.000 1.000 1.000
  - Transmit Power (dBm): 0.000 0.000 0.000 0.000
  - Receive Power (mW): 1.000 1.000 1.000 1.000
  - Receive Power (dBm): 0.000 0.000 0.000 0.000

Statistics:
  FEC:
    Corrected Codeword Count: 0
    Uncorrected Codeword Count: 0

MAC address information:
  Operational address: 0035.1a00.e67c
  Burnt-in address: 0035.1a00.e62c

Auto-negotiation enabled:
  No restricted parameters

Operational values:
  Speed: 100Mbps
  Duplex: Full Duplex
  Flowcontrol: None
  Loopback: None (or external)
  MTU: 1514
Using speed and negotiation auto command

When you configure the speed of the network interface (1G) using the `speed` and `negotiation auto` command, the interface autonegotiates all the parameters (full-duplex and pause) except speed. The speed is forced to the configured value.

This sample shows how to configure Gig interface speed to 100Mbps and autonegotiate other parameters:

```bash
Note

#configuration
(config)#interface GigabitEthernet 0/0/0/31
(config-if)#negotiation auto
(config-if)#speed 100
(config-if)#end
```

Use the `show controller GigE` and `show interface GigE` command to verify if the link is up, speed is forced to 100Mbps and autonegotiation is enabled:

```bash
#show interfaces GigabitEthernet 0/0/0/31
GigabitEthernet0/0/0/31 is up, line protocol is up
  Interface state transitions: 9
  Hardware is GigabitEthernet, address is 0035.1a00.e62c (bia 0035.1a00.e62c)
  Internet address is Unknown
  MTU 1514 bytes, BW 100000 Kbit (Max: 100000 Kbit)
  reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA, Full-duplex, 100Mb/s, TFD, link type is autonegotiation
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  Last link flapped 00:00:03
  Last input 00:00:00, output 00:00:00
  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
  90968 packets input, 11683189 bytes, 0 total input drops
  0 drops for unrecognized upper-level protocol
  Received 0 broadcast packets, 90968 multicast packets
  0 runts, 0 giants, 0 throttles, 0 parity
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  61287 packets output, 4348541 bytes, 0 total output drops
  Output 0 broadcast packets, 8664 multicast packets
  0 output errors, 0 underruns, 0 applique, 0 resets
  0 output buffer failures, 0 output buffers swapped out
  12 carrier transitions
```

In the above show output you will observe that the GigabitEthernet0/0/0/31 is up, and line protocol is up. This is because the speed at both ends is 100Mbps.

```bash
#show controllers GigabitEthernet 0/0/0/31
Operational data for interface GigabitEthernet0/0/0/31:
```
State:
Administrative state: enabled
Operational state: Up
LED state: Green On

Phy:
Media type: Four-pair Category 5 UTP PHY, full duplex

Optics:
Vendor: CISCO
Part number: SBCU-5740ARZ-CS1
Serial number: AVC194525HW
Wavelength: 0 nm

Digital Optical Monitoring:
Transceiver Temp: 0.000 C
Transceiver Voltage: 0.000 V

Alarms key: (H) Alarm high, (h) Warning high
(L) Alarm low, (l) Warning low

<table>
<thead>
<tr>
<th>Wavelength</th>
<th>Tx Power</th>
<th>Rx Power</th>
<th>Laser Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane (nm)</td>
<td>(dBm)</td>
<td>(mW)</td>
<td>(mA)</td>
</tr>
<tr>
<td>0</td>
<td>n/a</td>
<td>0.0</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

DOM alarms:
No alarms

Alarm Thresholds:
Transceiver Temp (C):
Transceiver Voltage (V):
Laser Bias (mA):
Transmit Power (mW):
Receive Power (mW):
Receive Power (dBm):

Statistics:
FEC:
Corrected Codeword Count: 0
Uncorrected Codeword Count: 0

MAC address information:
Operational address: 0035.1a00.e67c
Burnt-in address: 0035.1a00.e62c

Autonegotiation enabled:
Speed restricted to: 100Mbps
* autonegotiation is enabled and speed is forced to 100Mbps*/

Operational values:
Speed: 100Mbps
Duplex: Full Duplex
Flowcontrol: None
Loopback: None (or external)
MTU: 1514
MRU: 1514
Forward error correction: Disabled
Ethernet MTU

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

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

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

Note
IPv6 does not support fragmentation.

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

Jumbo frame support is automatically enabled 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).

Link Layer Discovery Protocol (LLDP)

Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2. Layer 2 is also known as the data link layer that runs on all Cisco-manufactured devices, such as routers, bridges, access servers, and switches. CDP allows the network management applications to automatically discover and learn about other Cisco devices that connect to the network.

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

With LLDP, you can also access the information about a particular physical network connection. If you use a non-Cisco monitoring tool (via SNMP,) LLDP helps you identify the Object Identifiers (OIDs) that the system supports. The following are the supported OIDs:

- 1.0.8802.1.1.2.1.4.1.1.4
- 1.0.8802.1.1.2.1.4.1.1.5
- 1.0.8802.1.1.2.1.4.1.1.6
- 1.0.8802.1.1.2.1.4.1.1.7
- 1.0.8802.1.1.2.1.4.1.1.8
- 1.0.8802.1.1.2.1.4.1.1.9
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.

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

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

To enable LLDP globally, complete the following steps:

1. RP/0/RP0/CPU0:router # configure
2. RP/0/RP0/CPU0:router(config) #lldp
3. end or commit

Running configuration

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

RP/0/RP0/CPU0:router#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:router#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
Enabling LLDP Per Interface

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

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

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

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

GigabitEthernet0/2/0/0:
**Dense Wavelength Division Multiplexing Tunable Optics**

The Dense Wavelength-Division Multiplexing (DWDM) wavelengths of the DWDM-SFP10G-C module on the Cisco NCS 5500 Series Aggregation Services Routers is tunable. You can configure the DWDM ITU wavelengths by using the itu channel command in the interface configuration mode. The itu channel command ensures that the traffic continues to flow.

The following table contains the wavelength mapping information for the DWDM module:

<table>
<thead>
<tr>
<th>Channel</th>
<th>Frequency (THz)</th>
<th>Wavelength (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>191.35</td>
<td>1566.723</td>
</tr>
<tr>
<td>2</td>
<td>191.40</td>
<td>1566.314</td>
</tr>
<tr>
<td>3</td>
<td>191.45</td>
<td>1565.905</td>
</tr>
<tr>
<td>4</td>
<td>191.50</td>
<td>1565.496</td>
</tr>
<tr>
<td>5</td>
<td>191.55</td>
<td>1565.087</td>
</tr>
<tr>
<td>6</td>
<td>191.60</td>
<td>1564.679</td>
</tr>
<tr>
<td>7</td>
<td>191.65</td>
<td>1564.271</td>
</tr>
<tr>
<td>8</td>
<td>191.70</td>
<td>1563.863</td>
</tr>
<tr>
<td>9</td>
<td>191.75</td>
<td>1563.455</td>
</tr>
<tr>
<td>10</td>
<td>191.80</td>
<td>1563.047</td>
</tr>
<tr>
<td>11</td>
<td>191.85</td>
<td>1562.640</td>
</tr>
<tr>
<td>12</td>
<td>191.90</td>
<td>1562.233</td>
</tr>
<tr>
<td>13</td>
<td>191.95</td>
<td>1561.826</td>
</tr>
<tr>
<td>14</td>
<td>192.00</td>
<td>1561.419</td>
</tr>
<tr>
<td>Channel</td>
<td>Frequency (THz)</td>
<td>Wavelength (nm)</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>15</td>
<td>192.05</td>
<td>1561.013</td>
</tr>
<tr>
<td>16</td>
<td>192.10</td>
<td>1560.606</td>
</tr>
<tr>
<td>17</td>
<td>192.15</td>
<td>1560.200</td>
</tr>
<tr>
<td>18</td>
<td>192.20</td>
<td>1559.794</td>
</tr>
<tr>
<td>19</td>
<td>192.25</td>
<td>1559.389</td>
</tr>
<tr>
<td>20</td>
<td>192.30</td>
<td>1558.983</td>
</tr>
<tr>
<td>21</td>
<td>192.35</td>
<td>1558.578</td>
</tr>
<tr>
<td>22</td>
<td>192.40</td>
<td>1558.173</td>
</tr>
<tr>
<td>23</td>
<td>192.45</td>
<td>1557.768</td>
</tr>
<tr>
<td>24</td>
<td>192.50</td>
<td>1557.363</td>
</tr>
<tr>
<td>25</td>
<td>192.55</td>
<td>1556.959</td>
</tr>
<tr>
<td>26</td>
<td>192.60</td>
<td>1556.555</td>
</tr>
<tr>
<td>27</td>
<td>192.65</td>
<td>1556.151</td>
</tr>
<tr>
<td>28</td>
<td>192.70</td>
<td>1555.747</td>
</tr>
<tr>
<td>29</td>
<td>192.75</td>
<td>1555.343</td>
</tr>
<tr>
<td>30</td>
<td>192.80</td>
<td>1554.940</td>
</tr>
<tr>
<td>31</td>
<td>192.85</td>
<td>1554.537</td>
</tr>
<tr>
<td>32</td>
<td>192.90</td>
<td>1554.134</td>
</tr>
<tr>
<td>33</td>
<td>192.95</td>
<td>1553.731</td>
</tr>
<tr>
<td>34</td>
<td>193.00</td>
<td>1553.329</td>
</tr>
<tr>
<td>35</td>
<td>193.05</td>
<td>1552.926</td>
</tr>
<tr>
<td>36</td>
<td>193.10</td>
<td>1552.524</td>
</tr>
<tr>
<td>37</td>
<td>193.15</td>
<td>1552.122</td>
</tr>
<tr>
<td>38</td>
<td>193.20</td>
<td>1551.721</td>
</tr>
<tr>
<td>39</td>
<td>193.25</td>
<td>1551.319</td>
</tr>
<tr>
<td>40</td>
<td>193.30</td>
<td>1550.918</td>
</tr>
<tr>
<td>41</td>
<td>193.35</td>
<td>1550.517</td>
</tr>
<tr>
<td>42</td>
<td>193.40</td>
<td>1550.116</td>
</tr>
<tr>
<td>Channel</td>
<td>Frequency (THz)</td>
<td>Wavelength (nm)</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>43</td>
<td>193.45</td>
<td>1549.715</td>
</tr>
<tr>
<td>44</td>
<td>193.50</td>
<td>1549.315</td>
</tr>
<tr>
<td>45</td>
<td>193.55</td>
<td>1548.915</td>
</tr>
<tr>
<td>46</td>
<td>193.60</td>
<td>1548.515</td>
</tr>
<tr>
<td>47</td>
<td>193.65</td>
<td>1548.115</td>
</tr>
<tr>
<td>48</td>
<td>193.70</td>
<td>1547.715</td>
</tr>
<tr>
<td>49</td>
<td>193.75</td>
<td>1547.316</td>
</tr>
<tr>
<td>50</td>
<td>193.80</td>
<td>1546.917</td>
</tr>
<tr>
<td>51</td>
<td>193.85</td>
<td>1546.518</td>
</tr>
<tr>
<td>52</td>
<td>193.90</td>
<td>1546.119</td>
</tr>
<tr>
<td>53</td>
<td>193.95</td>
<td>1545.720</td>
</tr>
<tr>
<td>54</td>
<td>194.00</td>
<td>1545.322</td>
</tr>
<tr>
<td>55</td>
<td>194.05</td>
<td>1544.924</td>
</tr>
<tr>
<td>56</td>
<td>194.10</td>
<td>1544.526</td>
</tr>
<tr>
<td>57</td>
<td>194.15</td>
<td>1544.128</td>
</tr>
<tr>
<td>58</td>
<td>194.20</td>
<td>1543.730</td>
</tr>
<tr>
<td>59</td>
<td>194.25</td>
<td>1543.333</td>
</tr>
<tr>
<td>60</td>
<td>194.30</td>
<td>1542.936</td>
</tr>
<tr>
<td>61</td>
<td>194.35</td>
<td>1542.539</td>
</tr>
<tr>
<td>62</td>
<td>194.40</td>
<td>1542.142</td>
</tr>
<tr>
<td>63</td>
<td>194.45</td>
<td>1541.746</td>
</tr>
<tr>
<td>64</td>
<td>194.50</td>
<td>1541.349</td>
</tr>
<tr>
<td>65</td>
<td>194.55</td>
<td>1540.953</td>
</tr>
<tr>
<td>66</td>
<td>194.60</td>
<td>1540.557</td>
</tr>
<tr>
<td>67</td>
<td>194.65</td>
<td>1540.162</td>
</tr>
<tr>
<td>68</td>
<td>194.70</td>
<td>1539.766</td>
</tr>
<tr>
<td>69</td>
<td>194.75</td>
<td>1539.371</td>
</tr>
<tr>
<td>70</td>
<td>194.80</td>
<td>1538.976</td>
</tr>
<tr>
<td>Channel</td>
<td>Frequency (THz)</td>
<td>Wavelength (nm)</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>71</td>
<td>194.85</td>
<td>1538.581</td>
</tr>
<tr>
<td>72</td>
<td>194.90</td>
<td>1538.186</td>
</tr>
<tr>
<td>73</td>
<td>194.95</td>
<td>1537.792</td>
</tr>
<tr>
<td>74</td>
<td>195.00</td>
<td>1537.397</td>
</tr>
<tr>
<td>75</td>
<td>195.05</td>
<td>1537.003</td>
</tr>
<tr>
<td>76</td>
<td>195.10</td>
<td>1536.609</td>
</tr>
<tr>
<td>77</td>
<td>195.15</td>
<td>1536.216</td>
</tr>
<tr>
<td>78</td>
<td>195.20</td>
<td>1535.822</td>
</tr>
<tr>
<td>79</td>
<td>195.25</td>
<td>1535.429</td>
</tr>
<tr>
<td>80</td>
<td>195.30</td>
<td>1535.036</td>
</tr>
<tr>
<td>81</td>
<td>195.35</td>
<td>1534.643</td>
</tr>
<tr>
<td>82</td>
<td>195.40</td>
<td>1534.250</td>
</tr>
<tr>
<td>83</td>
<td>195.45</td>
<td>1533.858</td>
</tr>
<tr>
<td>84</td>
<td>195.50</td>
<td>1533.465</td>
</tr>
<tr>
<td>85</td>
<td>195.55</td>
<td>1533.073</td>
</tr>
<tr>
<td>86</td>
<td>195.60</td>
<td>1532.681</td>
</tr>
<tr>
<td>87</td>
<td>195.65</td>
<td>1532.290</td>
</tr>
<tr>
<td>88</td>
<td>195.70</td>
<td>1531.898</td>
</tr>
<tr>
<td>89</td>
<td>195.75</td>
<td>1531.507</td>
</tr>
<tr>
<td>90</td>
<td>195.80</td>
<td>1531.116</td>
</tr>
<tr>
<td>91</td>
<td>195.85</td>
<td>1530.725</td>
</tr>
<tr>
<td>92</td>
<td>195.90</td>
<td>1530.334</td>
</tr>
<tr>
<td>93</td>
<td>195.95</td>
<td>1529.944</td>
</tr>
<tr>
<td>94</td>
<td>196.00</td>
<td>1529.553</td>
</tr>
<tr>
<td>95</td>
<td>196.05</td>
<td>1529.163</td>
</tr>
<tr>
<td>96</td>
<td>196.10</td>
<td>1528.773</td>
</tr>
</tbody>
</table>
Configuring the DWDM Tunable Optics

Perform the following procedure to configure the DWDM Tunable Optics module:

1. Router# enable //Enables the privileged EXEC mode. If prompted, enter your password.
2. Router# configure terminal
3. Router(config)# interface tengigabitethernet 4/11 // Specifies the 10-Gigabit Ethernet interface to be configured. slot/port—Specifies the location of the interface.
4. Router(config-if)# itu channel 28 //Sets the ITU channel. number specifies the ITU channel number. The acceptable values are from 1–96.

Verifying the ITU Configuration

The following example shows how to use the show controller optics command to check an ITU configuration:

RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/16
Tue Sep 5 08:25:54.127 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: Off
LED State: Off
Optics Status
  Optics Type: SFP+ 10G DWDM Tunable
  DWDM carrier Info: C BAND, MSA ITU Channel=49, Frequency=193.75THz, Wavelength=1547.316nm
  Alarm Status: 
    Detected Alarms: LOW-RX0-PWR
  LOS/LOL/Fault Status:
    Laser Bias Current = 0.0 mA
    Actual TX Power = 0.00 dBm
    RX Power = 0.00 dBm
  Performance Monitoring: Enable
  THRESHOLD VALUES
  ------------------------ ---------- --------- ------------ ---------
  Parameter | High Alarm | Low Alarm | High Warning | Low Warning
  ------------------------ ---------- --------- ------------ ---------

### RX Power Threshold (dBm)

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.9</td>
</tr>
<tr>
<td>-30.9</td>
</tr>
<tr>
<td>-7.0</td>
</tr>
<tr>
<td>-26.9</td>
</tr>
</tbody>
</table>

### TX Power Threshold (dBm)

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.9</td>
</tr>
<tr>
<td>-5.0</td>
</tr>
<tr>
<td>2.9</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
</tbody>
</table>

### LBC Threshold (mA)

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>75.00</td>
</tr>
<tr>
<td>25.00</td>
</tr>
<tr>
<td>70.00</td>
</tr>
<tr>
<td>30.00</td>
</tr>
</tbody>
</table>

### Temp. Threshold (Celsius)

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>75.00</td>
</tr>
<tr>
<td>-5.00</td>
</tr>
<tr>
<td>70.00</td>
</tr>
<tr>
<td>0.00</td>
</tr>
</tbody>
</table>

### Voltage Threshold (Volt)

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.63</td>
</tr>
<tr>
<td>2.97</td>
</tr>
<tr>
<td>3.46</td>
</tr>
<tr>
<td>3.13</td>
</tr>
</tbody>
</table>

### Temperature = 38.00 Celsius

### Voltage = 3.28 V

### DWDM Carrier Map table

<table>
<thead>
<tr>
<th>ITU Ch</th>
<th>G.694.1 Frequency (THz)</th>
<th>Wavelength (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -35</td>
<td>191.35</td>
<td>1566.723</td>
</tr>
<tr>
<td>2 -34</td>
<td>191.40</td>
<td>1566.314</td>
</tr>
<tr>
<td>3 -33</td>
<td>191.45</td>
<td>1565.905</td>
</tr>
<tr>
<td>4 -32</td>
<td>191.50</td>
<td>1565.496</td>
</tr>
<tr>
<td>5 -31</td>
<td>191.55</td>
<td>1565.087</td>
</tr>
<tr>
<td>6 -30</td>
<td>191.60</td>
<td>1564.679</td>
</tr>
<tr>
<td>7 -29</td>
<td>191.65</td>
<td>1564.271</td>
</tr>
<tr>
<td>8 -28</td>
<td>191.70</td>
<td>1563.863</td>
</tr>
<tr>
<td>9 -27</td>
<td>191.75</td>
<td>1563.455</td>
</tr>
<tr>
<td>10 -26</td>
<td>191.80</td>
<td>1563.047</td>
</tr>
<tr>
<td>11 -25</td>
<td>191.85</td>
<td>1562.640</td>
</tr>
<tr>
<td>12 -24</td>
<td>191.90</td>
<td>1562.233</td>
</tr>
<tr>
<td>13 -23</td>
<td>191.95</td>
<td>1561.826</td>
</tr>
<tr>
<td>14 -22</td>
<td>192.00</td>
<td>1561.419</td>
</tr>
</tbody>
</table>

### Transceiver Vendor Details

- **Form Factor**: SFP+
- **Vendor Info**
  - **Optics type**: SFP+ 10G DWDM Tunable
  - **Name**: CISCO-OCLARO
  - **OUI Number**: 00.0b.40
  - **Part Number**: TRS7080FNCCA033
  - **Rev Number**: 0000
  - **Serial Number**: ONT203800B
  - **PID**: DWDM-SFP10G-C
  - **VID**: V01

### DWDM Channel to Frequency/Wavelength Mapping

// DWDM Channel to Frequency/Wavelength Mapping
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/16 dwdm-carrrier-map
Tue Sep 5 08:26:31.175 UTC
DWDM Carrier Band:: (null)
MSA ITU channel range supported: 1-96

// DWDM Carrier Map table

---

### Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.1.x

42
Configuring Ethernet Interfaces

Configuring the DWDM Tunable Optics

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>11</td>
<td>193.65</td>
<td>1548.115</td>
</tr>
<tr>
<td>48</td>
<td>12</td>
<td>193.70</td>
<td>1547.715</td>
</tr>
<tr>
<td>49</td>
<td>13</td>
<td>193.75</td>
<td>1547.316</td>
</tr>
<tr>
<td>50</td>
<td>14</td>
<td>193.80</td>
<td>1546.917</td>
</tr>
<tr>
<td>51</td>
<td>15</td>
<td>193.85</td>
<td>1546.518</td>
</tr>
<tr>
<td>52</td>
<td>16</td>
<td>193.90</td>
<td>1546.119</td>
</tr>
<tr>
<td>53</td>
<td>17</td>
<td>193.95</td>
<td>1545.720</td>
</tr>
<tr>
<td>54</td>
<td>18</td>
<td>194.00</td>
<td>1545.322</td>
</tr>
<tr>
<td>55</td>
<td>19</td>
<td>194.05</td>
<td>1544.924</td>
</tr>
<tr>
<td>56</td>
<td>20</td>
<td>194.10</td>
<td>1544.526</td>
</tr>
<tr>
<td>57</td>
<td>21</td>
<td>194.15</td>
<td>1544.128</td>
</tr>
<tr>
<td>58</td>
<td>22</td>
<td>194.20</td>
<td>1543.730</td>
</tr>
<tr>
<td>59</td>
<td>23</td>
<td>194.25</td>
<td>1543.333</td>
</tr>
<tr>
<td>60</td>
<td>24</td>
<td>194.30</td>
<td>1542.936</td>
</tr>
<tr>
<td>61</td>
<td>25</td>
<td>194.35</td>
<td>1542.539</td>
</tr>
<tr>
<td>62</td>
<td>26</td>
<td>194.40</td>
<td>1542.142</td>
</tr>
<tr>
<td>63</td>
<td>27</td>
<td>194.45</td>
<td>1541.746</td>
</tr>
<tr>
<td>64</td>
<td>28</td>
<td>194.50</td>
<td>1541.349</td>
</tr>
<tr>
<td>65</td>
<td>29</td>
<td>194.55</td>
<td>1540.953</td>
</tr>
<tr>
<td>66</td>
<td>30</td>
<td>194.60</td>
<td>1540.557</td>
</tr>
<tr>
<td>67</td>
<td>31</td>
<td>194.65</td>
<td>1540.162</td>
</tr>
<tr>
<td>68</td>
<td>32</td>
<td>194.70</td>
<td>1539.766</td>
</tr>
<tr>
<td>69</td>
<td>33</td>
<td>194.75</td>
<td>1539.371</td>
</tr>
<tr>
<td>70</td>
<td>34</td>
<td>194.80</td>
<td>1538.976</td>
</tr>
<tr>
<td>71</td>
<td>35</td>
<td>194.85</td>
<td>1538.581</td>
</tr>
<tr>
<td>72</td>
<td>36</td>
<td>194.90</td>
<td>1538.186</td>
</tr>
<tr>
<td>73</td>
<td>37</td>
<td>194.95</td>
<td>1537.792</td>
</tr>
<tr>
<td>74</td>
<td>38</td>
<td>195.00</td>
<td>1537.397</td>
</tr>
<tr>
<td>75</td>
<td>39</td>
<td>195.05</td>
<td>1537.003</td>
</tr>
<tr>
<td>76</td>
<td>40</td>
<td>195.10</td>
<td>1536.609</td>
</tr>
<tr>
<td>77</td>
<td>41</td>
<td>195.15</td>
<td>1536.216</td>
</tr>
<tr>
<td>78</td>
<td>42</td>
<td>195.20</td>
<td>1535.822</td>
</tr>
</tbody>
</table>
Configuring Ethernet Interfaces

Configuring the DWDM Tunable Optics

---
| 79 | 43 | 195.25 | 1535.429 |
| 80 | 44 | 195.30 | 1535.036 |
| 81 | 45 | 195.35 | 1534.643 |
| 82 | 46 | 195.40 | 1534.250 |
| 83 | 47 | 195.45 | 1533.858 |
| 84 | 48 | 195.50 | 1533.465 |
| 85 | 49 | 195.55 | 1533.073 |
| 86 | 50 | 195.60 | 1532.681 |
| 87 | 51 | 195.65 | 1532.290 |
| 88 | 52 | 195.70 | 1531.898 |
| 89 | 53 | 195.75 | 1531.507 |
| 90 | 54 | 195.80 | 1531.116 |
| 91 | 55 | 195.85 | 1530.725 |
| 92 | 56 | 195.90 | 1530.334 |
| 93 | 57 | 195.95 | 1529.944 |
| 94 | 58 | 196.00 | 1529.553 |
| 95 | 59 | 196.05 | 1529.163 |
| 96 | 60 | 196.10 | 1528.773 |
---

// Change Frequency

RP/0/RP0/CPU0:ios#conf t
Tue Sep 5 08:34:14.312 UTC
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/16
RP/0/RP0/CPU0:ios(config-Optics)#shutdown
RP/0/RP0/CPU0:ios(config-Optics)#dwdm-carrier 50GHz-grid frequency 19335
RP/0/RP0/CPU0:ios(config-Optics)#commit
Tue Sep 5 08:34:39.943 UTC
RP/0/RP0/CPU0:ios(config-Optics)#end
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/16
Tue Sep 5 08:34:42.824 UTC

Controller State: Administratively Down
Transport Admin State: Out Of Service
Laser State: Off
LED State: Off

Optics Status

Optics Type: SFP+ 10G DWDM Tunable
DWDM carrier Info: C BAND, MSA ITU Channel=41, Frequency=193.35THz,
Wavelength=1550.517nm
Alarm Status:
-------------
Detected Alarms:
LOW-RX0-PWR

LOS/LOL/Fault Status:
Laser Bias Current = 0.0 mA
Actual TX Power = 0.00 dBm
RX Power = 0.00 dBm

Performance Monitoring: Enable

THRESHOLD VALUES
--------------------
<table>
<thead>
<tr>
<th>Parameter</th>
<th>High Alarm</th>
<th>Low Alarm</th>
<th>High Warning</th>
<th>Low Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rx Power Threshold(dBm)</td>
<td>-2.9</td>
<td>-30.9</td>
<td>-7.0</td>
<td>-26.9</td>
</tr>
<tr>
<td>Tx Power Threshold(dBm)</td>
<td>5.9</td>
<td>-5.0</td>
<td>2.9</td>
<td>-1.0</td>
</tr>
<tr>
<td>LBC Threshold(mA)</td>
<td>75.00</td>
<td>25.00</td>
<td>70.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Temp. Threshold(celsius)</td>
<td>75.00</td>
<td>-5.0</td>
<td>70.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Voltage Threshold(volt)</td>
<td>3.63</td>
<td>2.97</td>
<td>3.46</td>
<td>3.13</td>
</tr>
</tbody>
</table>

Polarization parameters not supported by optics

Temperature = 39.00 Celsius
Voltage = 3.28 V

Transceiver Vendor Details

Form Factor : SFP+
Vendor Info
----------
Optics type : SFP+ 10G DWDM Tunable
Name : CISCO-OCLARO
OUI Number : 00.0b.40
Part Number : TRS7080FNCCA033
Rev Number : 0000
Serial Number : ONT2038009B
PID : DWDM-SFP10G-C
VID : V01

// Change Wavelength

RP/0/RP0/CPU0:ios#conf t
Tue Sep 5 11:27:21.614 UTC
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/16
RP/0/RP0/CPU0:ios(config-Optics)#shutdown
RP/0/RP0/CPU0:ios(config-Optics)#dwdm-carrier 50GHz-grid wavelength 1539766
RP/0/RP0/CPU0:ios(config-Optics)#commit
Tue Sep 5 11:28:14.547 UTC
RP/0/RP0/CPU0:ios(config-Optics)#end
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/16
Tue Sep 5 11:28:30.934 UTC

Controller State: Administratively Down
Transport Admin State: Out Of Service
Laser State: Off
LED State: Off
Optics Status

Optics Type: SFP+ 10G DWDM Tunable
DWDM carrier Info: C BAND, MSA ITU Channel=68, Frequency=194.70THz, Wavelength=1539.766nm

Alarm Status:
--------------
Detected Alarms:
LOW-RX0-PWR

LOS/LOL/Fault Status:

Laser Bias Current = 0.0 mA
Actual TX Power = 0.00 dBm
RX Power = 0.00 dBm

Performance Monitoring: Enable

THRESHOLD VALUES
-----------------
<table>
<thead>
<tr>
<th>Parameter</th>
<th>High Alarm</th>
<th>Low Alarm</th>
<th>High Warning</th>
<th>Low Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rx Power Threshold(dBm)</td>
<td>-2.9</td>
<td>-30.9</td>
<td>-7.0</td>
<td>-26.9</td>
</tr>
<tr>
<td>Tx Power Threshold(dBm)</td>
<td>5.9</td>
<td>-5.0</td>
<td>-2.9</td>
<td>-1.0</td>
</tr>
<tr>
<td>LBC Threshold(mA)</td>
<td>75.00</td>
<td>25.00</td>
<td>70.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Temp. Threshold(celsius)</td>
<td>75.00</td>
<td>-5.00</td>
<td>70.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Voltage Threshold(volt)</td>
<td>3.63</td>
<td>2.97</td>
<td>3.46</td>
<td>3.13</td>
</tr>
</tbody>
</table>

Polarization parameters not supported by optics

Temperature = 38.00 Celsius
Voltage = 3.28 V

Transceiver Vendor Details

Form Factor : SFP+
Vendor Info
----------
Optics type : SFP+ 10G DWDM Tunable
Name : CISCO-OCLARO
OUI Number : 00.0b.40
Part Number : TRS7080FNCCA033
Rev Number : 0000
Serial Number : ONT2038009B
PID : DWDM-SFP10G-C
VID : V01

// Change Channel
RP/0/RP0/CP0:ios#conf t
Tue Sep 5 08:29:03.648 UTC
RP/0/RP0/CP0:ios(config)#controller optics 0/0/0/16
RP/0/RP0/CP0:ios(config-Optics)#shutdown
RP/0/RP0/CP0:ios(config-Optics)#dwdm-carrier 50GHz-grid ?
  frequency  Configure Frequency and Map to ITU Channel
  itu-ch     Configure the ITU 50GHz Grid ITU Channel
  wavelength Configure Wavelength and Map to ITU Channel
RP/0/RP0/CP0:ios(config-Optics)#dwdm-carrier 50GHz-grid itu-ch 84
RP/0/RP0/CP0:ios(config-Optics)#commit

RP/0/RP0/CP0:ios#show controllers optics 0/0/0/16
Tue Sep 5 08:29:54.851 UTC
Controller State: Administratively Down
Transport Admin State: Out Of Service
Laser State: Off
LED State: Off

Optics Status

Optics Type: SFP+ 10G DWDM Tunable
DWDM carrier Info: C BAND, MSA ITU Channel=84, Frequency=195.50THz, Wavelength=1533.465nm

Alarm Status:
------------------
Detected Alarms:
LOW-RX0-PWR

LOS/LOL/Fault Status:

Laser Bias Current = 0.0 mA
Actual TX Power = 0.00 dBm
RX Power = 0.00 dBm

Performance Monitoring: Enable

Threshold Values
------------------

<table>
<thead>
<tr>
<th>Parameter</th>
<th>High Alarm</th>
<th>Low Alarm</th>
<th>High Warning</th>
<th>Low Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rx Power Threshold(dBm)</td>
<td>-2.9</td>
<td>-30.9</td>
<td>-7.0</td>
<td>-26.9</td>
</tr>
<tr>
<td>Tx Power Threshold(dBm)</td>
<td>5.9</td>
<td>-5.0</td>
<td>2.9</td>
<td>-1.0</td>
</tr>
<tr>
<td>LBC Threshold(mA)</td>
<td>75.00</td>
<td>25.00</td>
<td>70.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Temp. Threshold(celsius)</td>
<td>75.00</td>
<td>-5.00</td>
<td>70.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Voltage Threshold(volt)</td>
<td>3.63</td>
<td>2.97</td>
<td>3.46</td>
<td>3.13</td>
</tr>
</tbody>
</table>

Polarization parameters not supported by optics

Temperature = 38.00 Celsius
Voltage = 3.28 V

Transceiver Vendor Details

Form Factor : SFP+
Vendor Info
--------------
Optics type : SFP+ 10G DWDM Tunable
Name : CISCO-OCLARO
OUI Number : 00.0b.40
Part Number : TRS7080FNCCA033
Rev Number : 0000
Serial Number : ONT2038009B
PID : DWDM-SFP10G-C
VID : V01

Priority Flow Control (PFC)

Priority flow control (PFC; IEEE 802.1Qbb), which is also referred to as Class-based Flow Control (CBFC) or Per Priority Pause (PPP), is a mechanism that prevents frame loss that is due to congestion. PFC is similar
to 802.3x Flow Control (pause frames) or link-level flow control (LFC). However, PFC functions on a per class-of-service (CoS) basis.

During congestion, PFC sends a pause frame that indicates which CoS value needs to be paused. A PFC pause frame contains a 2-octet timer value for each CoS that indicates the length of time that the traffic needs to be paused. The unit of time for the timer is specified in pause quanta. A quanta is the time that is required for transmitting 512 bits at the speed of the port. The range is from 0 to 65535.

The router sends out the required amount of pause frames or pause-threshold (x-off) messages to achieve lossless queues. It also sends out resume-threshold (x-on) messages.

Note

PFC asks the peer to stop sending frames of a particular CoS value by sending a pause frame to a well-known multicast address. This pause frame is a one-hop frame that is not forwarded when received by the peer. When the congestion is mitigated, the router stops sending the PFC frames to the upstream node.

PFC is supported only on the following line cards (including any MACSec / scale minor variants) of the modular NCS5500 chassis:

- NC55-36X100G-BA
- NC55-24H12F-SB
- NC55-24X100G-SB
- NC55-18H18F-BA
- NC55-36X100G-SB
- NC55-36X100G-U-SB

Restrictions for PFC

PFC has the following restrictions:

- PFC for transmit is not supported for internal traffic (recycle / loopback) and non-unicast traffic (broadcast / multicast).

- PFC for receive impacts all traffic meant to go out of the port. This may cause unintended drops to both unicast and non-unicast traffic because non-unicast traffic may consume buffer descriptors, thus starving unicast traffic. Hence, PFC is incompatible with sustained high rate non-unicast traffic in the system.

- PFC configuration will enable or disable both PFC transmit and receive functionalities. There is no support to enable only transmit or receive functions.

- When the PFC transmit thresholds for a particular CoS value are crossed, PFC pause frames for that CoS value will be transmitted out of all ports on that NP core, including the port that may have received pause frames from a peer. This is required as traffic for that CoS value may enter the router from any of the ports in that NP core.

- There is no option to clear the PFC frame counters displayed on the show controller command. They aggregate until the device reloads, irrespective of software configuration changes.

- There is no MIB support for PFC frame counters displayed on the show controller command.
• Standards compliant PFC receive functionality and watchdog monitoring requires an 8-priority egress queuing policy on all PFC enabled interfaces. Since 8-priority egress queuing policies are only supported in the non-HQoS profile, PFC is also supported only in the non-HQoS profile. For more details on this QoS prerequisite and configuration examples, please refer to Modular QoS Configuration Guide for Cisco NCS 5500 Series Routers.

• PFC is only qualified on 40G and 100G physical interface types. PFC is not supported on breakout ports for these interface types and is not qualified on other interface types.

• Being an Ethernet feature, PFC has to be individually configured on the member interfaces of a bundle instead of the bundle interface. The user is expected to either enable or disable PFC on all members of the bundle, as a mix isn’t supported.

Configuring Priority Flow Control

Use the following steps to configure Priority Flow Control:

**Configuration:**

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(configure)# interface HundredGigE0/0/0/0
RP/0/RP0/CPU0:router(config)# priority-flow-control on
```

**Running configuration:**

```
*Interface Level*
interface HundredGigE0/0/0/0
priority-flow-control mode on
```

**Verification:**

Per-port, per-CoS PFC Rx and Tx frame counters can be checked by the `show controllers <interface>` command:

```
RP/0/RP0/CPU0:router#show controllers hundredGigE 0/0/0/0
Thu Nov 28 11:13:22.829 UTC
Operational data for interface HundredGigE0/0/0/0:

State:
    Administrative state: disabled
    Operational state: Down (Reason: Link is shutdown)
    LED state: Off

Phy:
    Media type: Not known
    No optics present
    Alarms:
        Current: No alarms
    Statistics:
        FEC:
            Corrected Codeword Count: 0
            Uncorrected Codeword Count: 0

MAC address information:
    Operational address: 008a.96ce.6424
    Burnt-in address: 008a.96ce.6424

Autonegotiation disabled.

Priority Flow Control:
```
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.

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
```
Verify a Breakout Configuration

![Image of show int br location 0/1/CPU0 | i Te0/1/0/28]

<table>
<thead>
<tr>
<th>Interface</th>
<th>Status</th>
<th>Protocol</th>
<th>Traffic 10000</th>
<th>Traffic 10000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Te0/1/0/27/0</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
<tr>
<td>Te0/1/0/27/1</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
<tr>
<td>Te0/1/0/27/2</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
<tr>
<td>Te0/1/0/27/3</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
<tr>
<td>Te0/1/0/28/0</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
<tr>
<td>Te0/1/0/28/1</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
<tr>
<td>Te0/1/0/28/2</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
<tr>
<td>Te0/1/0/28/3</td>
<td>up</td>
<td>up</td>
<td>ARPA 10000</td>
<td>10000000</td>
</tr>
</tbody>
</table>
CHAPTER 5

Configuring Ethernet OAM

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

Feature History for Configuring Ethernet OAM

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.1.1</td>
<td>Support for the following features was introduced:</td>
</tr>
<tr>
<td></td>
<td>• Ethernet Link OAM</td>
</tr>
<tr>
<td></td>
<td>• Ethernet CFM</td>
</tr>
<tr>
<td>Release 7.1.1</td>
<td>Support for CFM adaptive bandwidth notifications was introduced.</td>
</tr>
</tbody>
</table>

- Configuring Ethernet OAM, on page 53
- Information About Configuring Ethernet OAM, on page 54
- How to Configure Ethernet OAM, on page 69
- CFM Over Bundles, on page 97
- Unidirectional Link Detection Protocol, on page 97
- Y.1731 Performance Monitoring, on page 101
- Bit Error Rate, on page 114
- Configuration Examples for Ethernet OAM, on page 115
- CFM Adaptive Bandwidth Notifications, on page 125

Configuring Ethernet OAM

This module describes the configuration of Ethernet Operations, Administration, and Maintenance (OAM).
Support for the following features was introduced:

| Release 6.1.1 | Support for CFM adaptive bandwidth notifications was introduced.
| Ethernet Link OAM | Ethernet CFM |

Support for CFM adaptive bandwidth notifications was introduced.

Release 7.1.1

Information About Configuring Ethernet OAM

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

**Ethernet Link OAM**

Ethernet as a Metro Area Network (MAN) or a Wide Area Network (WAN) technology benefits greatly from the implementation of Operations, Administration and Maintenance (OAM) features. Ethernet link OAM features allow Service Providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events. 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.
Ethernet Fault Detection (EFD) is a mechanism that allows Ethernet OAM protocols, such as CFM, to control the line protocol state of an interface.

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

EFD changes this to allow CFM to act as the line protocol for Ethernet interfaces. This allows CFM to control the interface state so that if a CFM defect (such as AIS or loss of continuity) is detected with an expected peer MEP, the interface can be shut down. This not only stops 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.

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

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

Figure 1: CFM 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.

Ethernet CFM

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

**Note**

Enable a maximum of 32 VLAN ranges per NPU. Else, when you reload the device, all CFM sessions over the 802.1Q VLAN interface might go down. Also, the corresponding bundle interface might go down. If more than 32 VLAN ranges exist on an NPU, remove the additional VLAN ranges and reload the device to address the issue.

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.

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

**Maintenance Domains**

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

Scenario A: Touching Domains OK
Scenario B: Nested Domains OK
Scenario C: Intersecting Domains Not Allowed
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 two types of MP:

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

• Maintenance Intermediate Points (MIPs)—Created in the middle of the domain. Unlike MEPS, MIPs do allow CFM frames at their own level to be forwarded.

MIP Creation

Unlike MEPS, MIPs are not explicitly configured on each interface. MIPs are created automatically according to the algorithm specified in the CFM 802.1ag standard. The algorithm, in brief, operates as follows for each interface:

• The bridge-domain or cross-connect for the interface is found, and all services associated with that bridge-domain or cross-connect are considered for MIP auto-creation.

• The level of the highest-level MEP on the interface is found. From among the services considered above, the service in the domain with the lowest level that is higher than the highest MEP level is selected. If there are no MEPs on the interface, the service in the domain with the lowest level is selected.
• The MIP auto-creation configuration (mip auto-create command) for the selected service is examined to determine whether a MIP should be created.

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

MEP and CFM Processing Overview

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

• Down MEPs—Send CFM frames from the interface where they are configured, and process CFM frames received on that interface. Down MEPs transmit AIS messages upward (toward the 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.
This figure shows maintenance points at different levels. Because domains are allowed to nest but not intersect (see ), a MEP at a low level always corresponds with a MEP or MIP at a higher level. In addition, only a single MIP is allowed on any interface—this is generally created in the lowest domain that exists at the interface and that does not have a MEP.

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

MEPs continue to operate normally if the interface they are created on is blocked by the Spanning Tree Protocol (STP); that is, CFM frames at the level of the MEP continue to be sent and received, according to the direction of the MEP. MEPs never allow CFM frames at the level of the MEP to be forwarded, so the STP block is maintained.
MIPs also continue to receive CFM frames at their level if the interface is STP blocked, and can respond to any received frames. However, MIPs do not allow CFM frames at the level of the MIP to be forwarded if the interface is blocked.

---

**Note**

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

---

**CFM Protocol Messages**

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

This section describes the following CFM messages:

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

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

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

**Figure 5: Continuity Check Message Flow**

All the MEPs in a service must transmit CCMs at the same interval. IEEE 802.1ag defines 7 possible intervals that can be used:
AMEP detects a loss of connectivity with one of its peer MEPs when some number of CCMs have been missed. This occurs when sufficient time has passed during which a certain number of CCMs were expected, given the CCM interval. This number is called the loss threshold, and is usually set to 3.

CFM is supported only on interfaces which 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`

The following table summarizes the supported values and parameters for MDID and SMAN. This table only details the MAID restriction on the hardware offload feature. There is no MAID restriction for software offload or non-offloaded MEPs.

For Cisco NCS 5500 series routers, "id null" has to be explicitly configured for the domain ID, for hardware offloaded sessions.

<table>
<thead>
<tr>
<th>Format</th>
<th>MDID</th>
<th>SMAN</th>
<th>Support</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>2 byte integer</td>
<td>Yes</td>
<td>Up to 2000 entries</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>13 bytes ICCCode (6 bytes) and UMC (7 bytes)</td>
<td>Yes</td>
<td>Up to 15 unique ICC</td>
<td></td>
</tr>
<tr>
<td>48 bytes string based</td>
<td>1-48 bytes of MDID and SMAN</td>
<td>No</td>
<td>Most commonly used</td>
<td></td>
</tr>
</tbody>
</table>

- 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 than 1 minute intervals.
- Sender TLV and Cisco Proprietary TLVs are not supported for MEPs with less than 1 min 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 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 MEP's own level.

• Loop—A CCM is received with the source MAC address equal to the MAC address of the interface where the MEP is operating.

• Configuration error—A CCM is received with the same MEP ID as the MEP ID configured for the receiving MEP.

• Cross-connect—A CCM is received with an MAID that does not match the locally configured MAID. This generally indicates a VLAN misconfiguration within the network, such that CCMs from one service are leaking into a different service.

• Peer interface down—A CCM is received that indicates the interface on the peer is down.

• Remote defect indication—A CCM is received carrying a remote defect indication.

---

**Note**

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

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

---

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

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

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

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

**Flexible VLAN Tagging for CFM**

The Flexible VLAN Tagging for CFM feature ensures that CFM packets are sent with the right VLAN tags so that they are appropriately handled as a CFM packet by the remote device. When packets are received by an edge router, they are treated as either CFM packets or data packets, depending on the number of tags in the header. The system differentiates between CFM packets and data packets based on the number of tags in the packet, and forwards the packets to the appropriate paths based on the number of tags in the packet.

CFM frames are normally sent with the same VLAN tags as the corresponding customer data traffic on the interface, as defined by the configured encapsulation and tag rewrite operations. Likewise, received frames are treated as CFM frames if they have the correct number of tags as defined by the configured encapsulation and tag rewrite configuration, and are treated as data frames (that is, they are forwarded transparently) if they have more than this number of tags.

In most cases, this behavior is as desired, since the CFM frames are then treated in exactly the same way as the data traffic flowing through the same service. However, in a scenario where multiple customer VLANs are multiplexed over a single multipoint provider service (for example, N:1 bundling), a different behavior might be desirable.

This figure shows an example of a network with multiple VLANS using CFM.
Figure 8: Service Provider Network With Multiple VLANs and CFM

This figure shows a provider’s access network, where the S-VLAN tag is used as the service delimiter. PE1 faces the customer, and PE2 is at the edge of the access network facing the core. N:1 bundling is used, so the interface encapsulation matches a range of C-VLAN tags. This could potentially be the full range, resulting in all:1 bundling. There is also a use case where only a single C-VLAN is matched, but the S-VLAN is nevertheless used as the service delimiter—this is more in keeping with the IEEE model, but limits the provider to 4094 services.

CFM is used in this network with a MEP at each end of the access network, and MIPs on the boxes within the network (if it is native Ethernet). In the normal case, CFM frames are sent by the up MEP on PE1 with two VLAN tags, matching the customer data traffic. This means that at the core interfaces and at the MEP on PE2, the CFM frames are forwarded as if they were customer data traffic, since these interfaces match only on the S-VLAN tag. So, the CFM frames sent by the MEP on PE1 are not seen by any of the other MPs.

Flexible VLAN tagging changes the encapsulation for CFM frames that are sent and received at Up MEPs. Flexible VLAN tagging allows the frames to be sent from the MEP on PE1 with just the S-VLAN tag that represents the provider service. If this is done, the core interfaces will treat the frames as CFM frames and they will be seen by the MIPs and by the MEP on PE2. Likewise, the MEP on PE1 should handle received frames with only one tag, as this is what it will receive from the MEP on PE2.

To ensure that CFM packets from Up MEPs are routed to the appropriate paths successfully, tags may be set to a specific number in a domain service, using the `tags` command. Currently, tags can only be set to one (1).

How to Configure Ethernet OAM

This section provides these configuration procedures:

Configuring Ethernet Link OAM

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

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

Perform these steps to configure an Ethernet OAM profile.

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 2**

ethernet oam profile *profile-name*

Example:

RP/0/RP0/CPU0:router(config)# ethernet oam profile Profile_1

Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.

| **Step 3**

link-monitor

Example:

RP/0/RP0/CPU0:router(config-eoam)# link-monitor

Enters the Ethernet OAM link monitor configuration mode.

| **Step 4**

symbol-period window *window*

Example:

RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period window 60000

(Optional) Configures the window size (in milliseconds) for an Ethernet OAM symbol-period error event. The IEEE 802.3 standard defines the window size as a number of symbols rather than a time duration. These two formats can be converted either way by using a knowledge of the interface speed and encoding.

The range is 1000 to 60000.

The default value is 1000.

| **Step 5**

symbol-period threshold low *threshold* high *threshold*

Example:

RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period threshold ppm low 1000000 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**

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 7**

frame threshold low *threshold* high *threshold*

Example:

RP/0/RP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000

(Optional) Configures the thresholds (in symbols) that triggers an Ethernet OAM frame error event. The high threshold is optional and is configurable only in conjunction with the low threshold.

The range is from 0 to 60000000.

The default low threshold is 1.

| **Step 8**

frame-period window *window*

Example:

RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window 60000

(Optional) Configures the window size (in milliseconds) for an Ethernet OAM frame-period error event. The IEEE 802.3 standard defines the window size as number of frames rather than a time duration. These two formats can be converted either way by using a knowledge of the interface speed. Note that the conversion assumes that all frames are of the minimum size.
### Configuring Ethernet OAM Profile

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window milliseconds 60000</td>
<td>The range is from 100 to 60000.</td>
</tr>
<tr>
<td></td>
<td>The default value is 1000.</td>
</tr>
<tr>
<td>Note</td>
<td>The only accepted values are multiples of the line card-specific polling interval, that is, 1000 milliseconds for most line cards.</td>
</tr>
<tr>
<td><strong>Step 9</strong> frame-period threshold low threshold high threshold</td>
<td>(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.</td>
</tr>
<tr>
<td>Example:</td>
<td>The range is from 0 to 1000000.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period threshold ppm low 100 high 1000000</td>
<td>The default low threshold is 1.</td>
</tr>
<tr>
<td></td>
<td>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 ofErrored Frame Window.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Step 10</strong> frame-seconds window window</td>
<td>(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event.</td>
</tr>
<tr>
<td>Example:</td>
<td>The range is 10000 to 900000.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds window 900000</td>
<td>The default value is 6000.</td>
</tr>
<tr>
<td></td>
<td>The only accepted values are multiples of the line card-specific polling interval, that is, 1000 milliseconds for most line cards.</td>
</tr>
<tr>
<td><strong>Step 11</strong> frame-seconds threshold low threshold high threshold</td>
<td>(Optional) Configures the thresholds (in seconds) that trigger a frame-seconds error event. The high threshold value can be configured only in conjunction with the low threshold value.</td>
</tr>
<tr>
<td>Example:</td>
<td>The range is 1 to 900</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 threshold high 900</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>exit</td>
<td>The default value is 1.</td>
</tr>
<tr>
<td>Example:</td>
<td>Exits back to Ethernet OAM mode.</td>
</tr>
</tbody>
</table>

**Step 12**
```
RP/0/RP0/CPU0:router(config-eoam-lm)# exit
```

<table>
<thead>
<tr>
<th>mib-retrieval</th>
<th>Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>

**Step 13**
```
RP/0/RP0/CPU0:router(config-eoam)# mib-retrieval
```

<table>
<thead>
<tr>
<th>connection timeout &lt;timeout&gt;</th>
<th>Configures the connection timeout period for an Ethernet OAM session. as a multiple of the hello interval.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>The range is 2 to 30.</td>
</tr>
<tr>
<td></td>
<td>The default value is 5.</td>
</tr>
</tbody>
</table>

**Step 14**
```
RP/0/RP0/CPU0:router(config-eoam)# connection timeout 30
```

| hello-interval {100ms|1s} | Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (1s). |
|------------------------|-----------------------------------------------------------------------------------------------------------|
| Example:               |                                                                                                          |

**Step 15**
```
RP/0/RP0/CPU0:router(config-eoam)# hello-interval 100ms
```

| mode {active|passive} | Configures the Ethernet OAM mode. The default is active. |
|-------------|-----------------------------------------------------------|
| Example:    |                                                            |

**Step 16**
```
RP/0/RP0/CPU0:router(config-eoam)# mode passive
```

| require-remote mode {active|passive} | Requires that active mode or passive mode is configured on the remote end before the OAM session becomes active. |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Example:                                  |                                                                                                            |

**Step 17**
```
RP/0/RP0/CPU0:router(config-eoam)# require-remote mode active
```

<table>
<thead>
<tr>
<th>require-remote mib-retrieval</th>
<th>Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>

**Step 18**
```
RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval
```

| action capabilities-conflict {disable | efd | error-disable-interface} | Specifies the action that is taken on an interface when a capabilities-conflict event occurs. The default action is to create a syslog entry. |
|-----------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Example:                    |                                                                                   | 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. |

```
RP/0/RP0/CPU0:router(config-eoam)# action capabilities-conflict efd
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 20** action critical-event \{disable | 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.  
Example:  
RP/0/RP0/CPU0:router(config-eoam)# action critical-event error-disable-interface  
**Note** • If you change the default, the log keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs. |
| **Step 21** action discovery-timeout \{disable | efd | error-disable-interface\} | Specifies the action that is taken on an interface when a connection timeout occurs. The default action is to create a syslog entry.  
Example:  
RP/0/RP0/CPU0:router(config-eoam)# action discovery-timeout efd  
**Note** • If you change the default, the log keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs. |
| **Step 22** action dying-gasp \{disable | error-disable-interface\} | 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.  
Example:  
RP/0/RP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface  
**Note** • If you change the default, the log keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs. |
| **Step 23** action high-threshold \{error-disable-interface | log\} | 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.  
Example:  
RP/0/RP0/CPU0:router(config-eoam)# action high-threshold error-disable-interface  
**Note** • If you change the default, the disable keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and take no action at the interface when the event occurs. |
| **Step 24** action session-down \{disable | efd | error-disable-interface\} | Specifies the action that is taken on an interface when an Ethernet OAM session goes down.  
Example: |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-eoam)# action session-down efd</code></td>
<td><strong>Note</strong>&lt;br&gt;If you change the default, the <code>log</code> keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.</td>
</tr>
<tr>
<td><strong>Step 25</strong>&lt;br&gt;<strong>action session-up disable</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config-eoam)# action session-up disable</code></td>
<td>Specifies that no action is taken on an interface when an Ethernet OAM session is established. The default action is to create a syslog entry.&lt;br&gt;&lt;br&gt;<strong>Note</strong>&lt;br&gt;If you change the default, the <code>log</code> keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.</td>
</tr>
<tr>
<td><strong>Step 26</strong>&lt;br&gt;**action uni-directional link-fault {disable</td>
<td>efd</td>
</tr>
<tr>
<td><strong>Step 27</strong>&lt;br&gt;**action wiring-conflict {disable</td>
<td>efd</td>
</tr>
<tr>
<td><strong>Step 28</strong>&lt;br&gt;<strong>uni-directional link-fault detection</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config-eoam)# uni-directional link-fault detection</code></td>
<td>Enables detection of a local, unidirectional link fault and sends notification of that fault to an Ethernet OAM peer.</td>
</tr>
<tr>
<td><strong>Step 29</strong>&lt;br&gt;<strong>commit</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config-if)# commit</code></td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
</tbody>
</table>
## Attaching an Ethernet OAM Profile to an Interface

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

### SUMMARY STEPS

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

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>`interface [FastEthernet</td>
<td>HundredGigE</td>
</tr>
<tr>
<td></td>
<td>Example: <code>interface TenGigE 0/1/0/0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>ethernet oam</code></td>
<td>Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>ethernet oam</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>profile profile-name</code></td>
<td>Attaches the specified Ethernet OAM profile (<code>profile-name</code>), and all of its configuration, to the interface.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>profile profile-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>commit</code></td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
</tbody>
</table>

### Purpose

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.

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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
| configure
| Example: RP/0/RP0/CPU0:router# configure terminal |
| **Step 2**
| interface [HundredGigE | TenGigE] interface-path-id
| Example: |

Ends the configuration session and exits to the EXEC mode.

Example:

RP/0/RP0/CPU0:router(config-if)# end
Purpose
Command or Action | Purpose | Note
--- | --- | ---
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0 | **The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.**

**Step 3**

**Command or Action**
```plaintext
RP/0/RP0/CPU0:router(config-if)# ethernet oam
```

**Note**
- Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.

**Step 4**

**Command or Action**
```plaintext
RP/0/RP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface
```

**Note**
- 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**

**Command or Action**
```plaintext
RP/0/RP0/CPU0:router(config-if)# commit
```

**Note**
- Saves the configuration changes to the running configuration file and remains within the configuration session.

**Step 6**

**Command or Action**
```plaintext
RP/0/RP0/CPU0:router(config-if)# end
```

**Note**
- 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:

```plaintext
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
```
Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:

Note

CFM is not supported for the following:

• L3 Interfaces and Sub-Interfaces
• Bundle Member Ports
• EVPN-FXC
• Bridge Domain
• VPLS

Configuring a CFM Maintenance Domain

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

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 ethernet cfm</td>
<td>Enters Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RP/0/RP0/CPU0:router(config)# ethernet cfm</strong></td>
</tr>
</tbody>
</table>

### 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 32000 CFM services for a maintenance domain. To configure services for a CFM maintenance domain, perform the following steps:

#### SUMMARY STEPS

1. configure
2. ethernet cfm
### Configuring Services for a CFM Maintenance Domain

#### 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure&lt;br&gt;Example: RP/0/RP0/CPU0:router# configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>ethernet cfm&lt;br&gt;Example: RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td>Enters Ethernet CFM configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td>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.</td>
</tr>
<tr>
<td>Step 4</td>
<td>service service-name {down-meps</td>
<td>xconnect-group xconnect-group-name p2p xconnect-name} [id [icc-based icc-string umc-string]] [ [number number]]&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1</td>
</tr>
<tr>
<td>Step 5</td>
<td>end or commit&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

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

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

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

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Entering <code>cancel</code> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>`service service-name {down-meps</td>
<td>xconnect group xconnect-group-name p2p xconnect-name}[id [icc-based icc-string umc-string]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>continuity-check interval time [loss-threshold threshold]</code></td>
<td>(Optional) Enables Continuity Check and specifies the time interval at which CCMs are transmitted or to set the threshold limit for when a MEP is declared down.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>continuity-check archive hold-time minutes</code></td>
<td>(Optional) Configures how long information about peer MEps is stored after they have timed out.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><code>continuity-check loss auto-traceroute</code></td>
<td>(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>
| **Example:** | RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit | - When you use the `end` command, the system prompts you to commit changes:  
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
  (cancel)  
  
  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.  
  
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.  
  
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.  
  
  - Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session. |
## Configuring Automatic MIP Creation for a CFM Service

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

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

### SUMMARY STEPS

1. `configure`
2. `ethernet cfm`
3. `domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]`
4. `service service-name (down-meps | xconnect-group xconnect-group-name p2p xconnect-name)[id [icc-basedicc-string umc-string] | [number number]]`
5. `mip auto-create {all | lower-mep-only} {ccm-learning}`
6. `end` or `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong>&lt;br&gt;Example:&lt;br&gt;RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>ethernet cfm</strong>&lt;br&gt;Example:&lt;br&gt;RP/0/RP0/CPU0:router# ethernet cfm</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]</strong>&lt;br&gt;Example:&lt;br&gt;RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**service service-name (down-meps</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>**mip auto-create {all</td>
</tr>
</tbody>
</table>

(Optional) Enables the automatic creation of MIPs in a bridge domain. `ccm-learning` option enables CCM learning for MIPs created in this service. This must be used only in...
Purpose

Command or Action | Purpose
-----------------|---------------------------------------------------
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all ccm-learning | services with a relatively long CCM interval of at least 100 ms. CCM learning at MIPs is disabled by default.

**Step 6**

Example:

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

**Example:**

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

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

**SUMMARY STEPS**

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet cfm</td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]</td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service service-name {bridge group bridge-domain-group bridge-domain bridge-domain-name</td>
<td>Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPS, or associate the service with a bridge domain or xconnect where MIPs and up MEPs will be created.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> mep crosscheck</td>
<td>Enters CFM MEP crosscheck configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> mep-id mep-id-number [mac-address mac-address]</td>
<td>Enables cross-check on a MEP.</td>
</tr>
</tbody>
</table>
| Example: RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10 | **Note**
| | • Repeat this command for every MEP that you want included in the expected set of MEPS for cross-check. |
| **Step 7** end or commit | Saves configuration changes. |
| Example: RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit | • When you use the end command, the system prompts you to commit changes: |
| | Uncommitted changes found, commit them before |
### Configuring Other Options for a CFM Service

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

#### SUMMARY STEPS

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
| `configure`
| Example:
| RP/0/RP0/CPU0:router# configure |
| **Step 2**
| `ethernet cfm`
| Example:
| RP/0/RP0/CPU0:router# ethernet cfm |

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.
### Configuring Ethernet OAM

#### Command or Action

**Step 3**

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

**Step 4**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`service service-name {bridge group bridge-domain-group bridge-domain bridge-domain-name</td>
<td>down-meps</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>The <code>id</code> sets the short MA name.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
</tbody>
</table>

**Step 5**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>maximum-meps number</code></td>
<td>(Optional) Configures the maximum number (2 to 8190) of MEPS across the network, which limits the number of peer MEPS recorded in the database.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000</td>
<td></td>
</tr>
</tbody>
</table>

**Step 6**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`log {ais</td>
<td>continuity-check errors</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors</td>
<td></td>
</tr>
</tbody>
</table>

**Step 7**

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

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

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

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

**Command or Action**

- **cancel**
- **commit**

**Purpose**

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

---

### Configuring CFM MEPs

#### SUMMARY STEPS

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

#### DETAILED STEPS

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

| **Step 2** | Type of Ethernet interface on which you want to create a MEP. Enter **HundredGigE** or **TenGigE** and the physical interface or virtual interface. |
| **interface** {HundredGigE | TenGigE} interface-path-id | |
| **Example:** | |
| RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1 | |

| **Step 3** | 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 subinterface path ID. Naming convention is **interface-path-id.subinterface**. The period in front of the subinterface value is required as part of the notation. |
| **interface** {HundredGigE | TenGigE | Bundle-Ether} interface-path-id.subinterface | |
| **Example:** | |
| RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1 | |

**Note**

- Use the **show interfaces** command to see a list of all interfaces currently configured on the router.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>vrf vrf-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if)# vrf vrf_A</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>**interface {HundredGigE</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>ethernet cfm</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if)# ethernet cfm</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>mep domain domain-name service service-name mep-id id-number</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>cos cos</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>end or commit</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>• Entering <code>no</code> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Entering <code>cancel</code> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>

## Configuring Y.1731 AIS

This section has the following step procedures:

### Configuring AIS in a CFM Domain Service

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

### SUMMARY STEPS

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

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters Ethernet CFM global configuration mode.</td>
</tr>
<tr>
<td><code>ethernet cfm</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies the domain and domain level.</td>
</tr>
<tr>
<td><code>domain name level level</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

```
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
```

**Step 4**

**service** *name* **bridge group** *name* **bridge-domain** *name*  

**Example:**

```
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
```

**Step 5**

**service** *name* **xconnect group** *xconnect-group-name* **p2p**  

**Example:**

```
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 xconnect group XG1 p2p X2
```

**Step 6**

```
ais transmission [interval {1s|1m}][cos cos]
```

**Example:**

```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```

**Step 7**

```
log ais
```

**Example:**

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

**Step 8**

```
end or commit
```

**Example:**

```
RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit
```

---

**Saves configuration changes.**

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

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

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
| `configure`
| Example:
| RP/0/RP0/CPU0:router# configure
| Enters global configuration mode. |
| **Step 2**
| `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:
| RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7
| Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface. |
| **Step 5**
| `end` or `commit`
| Example:
| RP/0/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. |
Purpose
Command or Action | Purpose
--- | ---
Enters global configuration mode. | configure
Example: 
RP/0/RP0/CPU0:router# configure

Enters Ethernet CFM global configuration mode. | ethernet cfm
Example: 
RP/0/RP0/CPU0:router(config)# ethernet cfm

Specifies the domain and domain level. | domain name level level
Example: 
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1

Specifies the service, bridge group, and bridge domain. | service name bridge group name bridge-domain name
Example: 

---

**Configuring Flexible VLAN Tagging for CFM**

Use this procedure to set the number of tags in CFM packets in a CFM domain service.

**SUMMARY STEPS**

1. configure
2. ethernet cfm
3. domain name level level
4. service name bridge group name bridge-domain name
5. tags number
6. end or commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| Example: 
RP/0/RP0/CPU0:router# configure |
| **Step 2** ethernet cfm | Enters Ethernet CFM global configuration mode. |
| Example: 
RP/0/RP0/CPU0:router(config)# ethernet cfm |
| **Step 3** domain name level level | Specifies the domain and domain level. |
| Example: 
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1 |
| **Step 4** service name bridge group name bridge-domain name | Specifies the service, bridge group, and bridge domain. |

---

Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.1.x
### Configuring Ethernet OAM

#### Verifying the CFM Configuration

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-cfm-dmn)# service S2 bridge group BG1 bridge-domain BD2</code></td>
<td>Specifies the number of tags in CFM packets. Currently, the only valid value is 1.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>tags number</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# tags 1</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>end or commit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit</code></td>
</tr>
</tbody>
</table>

- **Step 5**: `tags number`<br>  Specifies the number of tags in CFM packets. Currently, the only valid value is 1.<br><br> **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# tags 1`

- **Step 6**: `end` or `commit`<br>  Saves configuration changes.<br><br> **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`

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

---

**Verifying the CFM Configuration**

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

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

---

**Note**<br>  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.
Troubleshooting Tips

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

SUMMARY STEPS

1. To verify connectivity to a problematic MEP, use the `ping ethernet cfm` command as shown in this example:

   ```
   RP/0/RP0/CPU0:router# ping ethernet cfm domain D1 service S1 mep-id 16 source interface TenGigE 0/0/0/1
   Type escape sequence to abort.
   Sending 5 CFM Loopbacks, timeout is 2 seconds -
   Domain foo (level 2), Service foo
   Source: MEP ID 1, interface TenGigE0/0/0/1
   Target: 0001.0002.0003 (MEP ID 16):
   Running (5s) ...
   Success rate is 60.0 percent (3/5), round-trip min/avg/max = 1251/1349/1402 ms
   Out-of-sequence: 0.0 percent (0/3)
   Bad data: 0.0 percent (0/3)
   Received packet rate: 1.4 pps
   ```

2. If the results of the `ping ethernet cfm` command show a problem with connectivity to the peer MEP, use the `traceroute ethernet cfm` command to help further isolate the location of the problem as shown in the following example:

   ```
   RP/0/RP0/CPU0:router# traceroute ethernet cfm domain D1 service S1 mep-id 16 source interface TenGigE 0/0/0/2
   Traceroutes in domain D1 (level 4), service S1
   Source: MEP-ID 1, interface TenGigE0/0/0/2
   Hop Hostname/Last Ingress MAC/name Egress MAC/Name Relay
   --- --------------- ---------------------- ----- ----- ---- ---- -------
   1 ios 0000-0001.0203.0400 [Down] 0001.0203.0400 [Down] FDB
   2 abc TenGigE0/0/0/2 0001.0203.0401 [Ok] FDB
   3 bcd 0001.0203.0402 [Ok] Not present Hit
   abc TenGigE0/0
   Replies dropped: 0
   ```

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

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

---

**CFM Over Bundles**

CFM over bundle supports the following:

- CFM Maintenance Points — UP MEP, Down MEP, and MIP, which includes L2 bundle main and sub-interfaces.
- CCM interval of 100ms, 1s, 10s, 1min, and 10mins.
- RP OIR/VM reload without impacting learnt CFM peer MEPs.
- Process restart without impacting CFM sessions.
- CFM MEPs on bundle interfaces as software offloaded MEPs, with all possible rewrite and encapsulation combinations supported by L2 sub-interfaces.
- CCM learning on MIP over bundle interfaces. CCM database learning supports investigation of one CCM out of 50 that goes over MIP.
- Static and dynamic 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 except for those matching VLAN tag ‘any’.
- 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.

**Unidirectional Link Detection Protocol**

Unidirectional Link Detection (UDLD) is a single-hop physical link protocol for monitoring an ethernet link, including both point-to-point and shared media links. This is a Cisco-proprietary protocol to detect link problems, which are not detected at the physical link layer. This protocol is specifically targeted at possible wiring errors, when using unbundled fiber links, where there can be a mismatch between the transmitting and receiving connections of a port.
Limitations

- UDLD is not tunneled through L2VPN like other slow protocols.
- UDLD must not be enabled on a Switched Port Analyzer (SPAN) source or a destination port.
- UDLD must not be enabled on a port that acts as a source or destination port for SPAN.

Types of Fault Detection

UDLD can detect these types of faults:

- Transmit faults — These are cases where there is a failure in transmitting packets from the local port to the peer device, but packets are being received from the peer. These faults are caused by failure of the physical link (where notification at layer 1 of unidirectional link faults is not supported by the media) as well as packet path faults on the local or peer device.
- Miswiring faults — These are cases where the receiving and transmitting sides of a port on the local device are connected to different peer ports (on the same device or on different devices). This can occur when using unbundled fibers to connect fiber optic ports.
- Loopback faults — These are cases where the receiving and transmitting sides of a port are connected to each other, creating a loopback condition. This can be an intentional mode of operation, for certain types of testing, but UDLD must not be used in these cases.
- Receive faults — The protocol includes a heartbeat signal that is transmitted at a negotiated periodic interval to the peer device. Missed heartbeats can therefore be used to detect failures on the receiving side of the link (where they do not result in interface state changes). These could be caused by a unidirectional link with a failure only affecting the receiving side, or by a link which has developed a bidirectional fault. This detection depends on reliable, regular packet transmission by the peer device. For this reason, the UDLD protocol has two (configurable) modes of operation which determine the behavior on a heartbeat timeout. These modes are described in the section UDLD Modes of Operation, on page 98.

UDLD Modes of Operation

UDLD can operate in these modes:

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

Note

The difference of behavior between normal and aggressive modes is only seen in case of neighbor timeout. In all other cases, irrespective of the normal or aggressive mode, the system error disables a link once a unidirectional link is detected.
Configure UDLD

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

Perform the following steps to configure UDLD protocol on an interface:

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0

The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.

Running Configuration

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

aggressive  Run UDLD in aggressive mode
normal  Run UDLD in normal mode

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

RP/0/RSP0/CPU0:router(config-if-udld)# message-time?
<7-90>  'Mslow' message time (in seconds) to use for the UDLD protocol

RP/0/RSP0/CPU0:router(config-if-udld)# message-time 50

RP/0/RSP0/CPU0:router(config-if-udld)# destination mac-address?
H.H.H  A valid multicast MAC address

cisco-l2cp  Use the Cisco L2CP MAC address (used by CDP)

RP/0/RSP0/CPU0:router(config-if-udld)# destination mac-address 0100.5e01.0101

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

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

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

RP/0/RSP0/CPU0:router# sh run interface TenGigE 0/0/0/0
interface TenGigE0/0/0/0
ethernet udld
mode aggressive
message-time 50
destination mac-address 0100.5e01.0101
logging disable

!

Verification

RP/0/RP0/CPU0:IOS#sh ethernet udld interfaces

Device ID: 00:8a:96:e1:20:d8
Device name: IOS
Interface TenGigE0/0/0/0
Port state: Up
Main FSM state: Advertising
Detection FSM state: Unknown
Message interval: 7 seconds
Timeout interval: 5 seconds
Destination MAC: 01:00:5e:01:01:01

RP/0/RP0/CPU0:IOS#sh ethernet udld statistics tenGigE 0/0/0/0

Interface TenGigE0/0/0/0
Counters last cleared: 00:01:18 ago
Main FSM transitions (to each state)
Link up: 1
Detection: 0
Advertise: 1
Port shutdown: 0
UDLD inactive: 0
Detection FSM transitions (to each state)
Unknown: 0
Bidirectional: 0
Unidirectional: 0
Neighbor mismatch: 0
Loopback: 0
Rx packet counts
Probe: 0
Echo: 0
Flush: 0
Invalid packets (dropped): 0
Tx packet counts
Probe: 19
Echo: 0
Flush: 0
Unable to send (dropped): 0

RP/0/RP0/CPU0:IOS#
RP/0/RP0/CPU0:IOS#sh ethernet udld daemon database

Interface TenGigE0/0/0/0

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface handle</td>
<td>Te0/0/0/0 (0x00000200)</td>
</tr>
<tr>
<td>Name</td>
<td>Te0/0/0/0</td>
</tr>
<tr>
<td>Name (long internal format)</td>
<td>TenGigE0_0_0_0</td>
</tr>
<tr>
<td>Configured ?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Caps add in progress ?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Caps remove in progress ?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Caps added ?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Protocol start pending ?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Protocol running ?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Registered for packet I/O ?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Aggressive mode ?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Logging enabled ?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Error disabled on start ?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Error disabled during ISSU ?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Attributes read ?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Pending state down nfn ?</td>
<td>FALSE</td>
</tr>
<tr>
<td>Message time</td>
<td>50</td>
</tr>
</tbody>
</table>

**Y.1731 Performance Monitoring**

Y.1731 Performance Monitoring (PM) provides a standard Ethernet PM function that includes measurement of Ethernet frame delay, frame delay variation, frame loss, and frame throughput measurements. This is specified by the ITU-T Y-1731 standard and interpreted by the Metro Ethernet Forum (MEF) standards group.

The router supports the following:

- Two-Way Delay Measurement (DM)
- Synthetic Loss Measurement (SLM)
Two-Way Delay Measurement

Use the Ethernet frame delay measurement to measure frame delay and frame delay variations. The system measures the Ethernet frame delay by using the Delay Measurement Message (DMM) method.

Restrictions for Configuring Two-Way Delay Measurement

Follow the guidelines and restrictions listed here when you configure two-way delay measurement:

- Y.1731 PM does not support One-Way DMM since PTP support is not available in the Release 6.3.1 for NCS 5500.
- System supports software-based timestamping for Two-Way DMM for NCS5502 and NCS5508 routers. The restriction is only applicable to UP MEP (Maintenance association End Point), which requires core NPU (Network Processor) and access NPU to have ToD (Time of Day) in sync to support 64-bit hardware-based timestamping. After you enable PTP (Precision Time Protocol) and sync all NPUs, the restriction is removed.

Configuring Two-Way Delay Measurement

Perform the following steps to configure two-way delay measurement:

RP/0/RP0/CPU0:router (config) # ethernet sla

profile DMM type cfm-delay-measurement

probe
   send burst every 5 seconds packet count 5 interval 1 seconds
   schedule
   every 1 minutes for 40 seconds
   !

statistics
   measure round-trip-delay
   buckets size 1 probes
   buckets archive 5
   !

measure round-trip-jitter
   buckets size 1 probes
   buckets archive 1
   !
   !

interface TenGigE0/0/0/10.1 l2transport
encapsulation dot1q 1
ethernet cfm
   mep domain DOWN0 service s10 mep-id 2001
   sla operation profile DMM target mep-id 6001
   !

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

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

RP/0/RP0/CPU0:router #

ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mac-address 2.3.4
Running Configuration

RP/0/RP0/CPU0:ios#show ethernet cfm peer meps
Mon Sep 11 12:09:44.534 UTC
Flags:
  > - Ok
  I - Wrong interval
  R - Remote Defect received
  V - Wrong level
  L - Loop (our MAC received)
  T - Timed out
  C - Config (our ID received)
  M - Missing (cross-check)
  X - Cross-connect (wrong MAID)
  U - Unexpected (cross-check)
  * - Multiple errors received
  S - Standby

Domain UP6 (level 6), Service s6
Up MEP on FortyGigE0/0/1/2.1 MEP-ID 1
================================================================================
<table>
<thead>
<tr>
<th>St</th>
<th>ID</th>
<th>MAC Address</th>
<th>Port</th>
<th>Up/Downtime</th>
<th>CcmRcvd</th>
<th>SeqErr</th>
<th>RDI Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4001</td>
<td>70e4.227c.2865</td>
<td>Up</td>
<td>00:01:27</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Domain DOWN0 (level 0), Service s10
Down MEP on TenGigE0/0/0/10.1 MEP-ID 2001
================================================================================
<table>
<thead>
<tr>
<th>St</th>
<th>ID</th>
<th>MAC Address</th>
<th>Port</th>
<th>Up/Downtime</th>
<th>CcmRcvd</th>
<th>SeqErr</th>
<th>RDI Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6001</td>
<td>70e4.227c.287a</td>
<td>Up</td>
<td>00:02:11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios#show running-config
Mon Sep 11 12:10:18.467 UTC
Building configuration...
!! IOS XR Configuration version = 6.4.1.14
!! Last configuration change at Mon Sep 11 12:08:16 2017 by root
!
logging console disable
telnet vrf default ipv4 server max-servers 10
username root
group root-lr
group cisco-support
secret 5 $1QJ3$94M5/wK5J0v/lpAu/wz31/
!
line console
eexec-timeout 0 0
!
eternet cfm
domain UP6 level 6 id null
  service s6 xconnect group g1 p2p pl id number 6
  mip auto-create all ccm-learning
  continuity-check interval 1s
  mep crosscheck
  mep-id 4001
  !
!
domain DOWN0 level 0 id null
  service s10 down-meps id number 10
  continuity-check interval 1s
  mep crosscheck
  mep-id 6001
  !
!
profile DMM type cfm-delay-measurement
  probe
    send burst every 5 seconds packet count 5 interval 1 seconds
! schedule
every 1 minutes for 40 seconds
!
statistics
measure round-trip-delay
  buckets size 1 probes
  buckets archive 5
!
measure round-trip-jitter
  buckets size 1 probes
  buckets archive 1
!
interface MgmtEth0/RP0/CPU0/0
  shutdown
!
interface TenGigE0/0/0/0
  shutdown
!
interface TenGigE0/0/0/1
  shutdown
!
interface TenGigE0/0/0/2
  shutdown
!
interface TenGigE0/0/0/3
  shutdown
!
interface TenGigE0/0/0/4
  shutdown
!
interface TenGigE0/0/0/5
  shutdown
!
interface TenGigE0/0/0/6
  shutdown
!
interface TenGigE0/0/0/7
  shutdown
!
interface TenGigE0/0/0/8
  shutdown
!
interface TenGigE0/0/0/9
  shutdown
!
interface TenGigE0/0/0/10
  l2transport
  encapsulation dot1q 1
  ethernet cfm
    mep domain DOWN0 service s10 mep-id 2001
    sla operation profile DMM target mep-id 6001
    sla operation profile test-slm target mep-id 6001
!
!
interface TenGigE0/0/0/11
  shutdown
!
interface TenGigE0/0/0/12
  shutdown
!
interface TenGigE0/0/0/13
  shutdown
!
interface TenGigE0/0/0/14
shutdown
!
interface TenGigE0/0/0/15
shutdown
!
interface TenGigE0/0/0/16
shutdown
!
interface TenGigE0/0/0/17
shutdown
!
interface TenGigE0/0/0/18
shutdown
!
interface TenGigE0/0/0/19
shutdown
!
interface TenGigE0/0/0/20
shutdown
!
interface TenGigE0/0/0/21
shutdown
!
interface TenGigE0/0/0/22
shutdown
!
interface TenGigE0/0/0/23
shutdown
!
interface TenGigE0/0/0/24
shutdown
!
interface TenGigE0/0/0/25
shutdown
!
interface TenGigE0/0/0/26
shutdown
!
interface TenGigE0/0/0/27
shutdown
!
interface TenGigE0/0/0/28
shutdown
!
interface TenGigE0/0/0/29
shutdown
!
interface TenGigE0/0/0/30
shutdown
!
interface TenGigE0/0/0/31
shutdown
!
interface TenGigE0/0/0/32
shutdown
!
interface TenGigE0/0/0/33
shutdown
!
interface TenGigE0/0/0/34
shutdown
!
interface TenGigE0/0/0/35
  shutdown
!
interface TenGigE0/0/0/36
  shutdown
!
interface TenGigE0/0/0/37
  shutdown
!
interface TenGigE0/0/0/38
  shutdown
!
interface TenGigE0/0/0/39
  shutdown
!
interface TenGigE0/0/1/0/1
  shutdown
!
interface TenGigE0/0/1/0/2
  shutdown
!
interface TenGigE0/0/1/0/3
  shutdown
!
controller Optics0/0/1/0
  breakout 4x10
!
interface HundredGigE0/0/1/1
  shutdown
!
interface FortyGigE0/0/1/2.1
  l2transport
  encapsulation dot1q 1
  ethernet cfm
    mep domain UP6 service s6 mep-id 1
      sla operation profile DMM target mep-id 6001
      sla operation profile test-slm target mep-id 6001
!
!
l2vpn
  xconnect group g1
    p2p p1
      interface TenGigE0/0/0/10.1
      interface FortyGigE0/0/1/2.1
!
!
end

Verification

One-way Delay (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: 1912765.961ms; Max: 1912765.961ms; Mean: 1912765.961ms; StdDev: -2147483.648ms

One-way Delay (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Verification

One-way Delay (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: 1912765.961ms; Max: 1912765.961ms; Mean: 1912765.961ms; StdDev: -2147483.648ms
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: -1912765.952ms; Max: -1912765.951ms; Mean: -1912765.951ms; StdDev: -2147483.648ms

Round Trip Jitter
~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
Result count: 9
Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms

One-way Jitter (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
Result count: 9
Min: 0.000ms; Max: 0.000ms; Mean: 0.000ms; StdDev: 0.000ms

One-way Jitter (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
Result count: 9
Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms

RP/0/RP0/CPU0:ios#ethernet sla on-demand operation type cfm-syn probe domain DOWN0 source interface tenGigE 0/0/0/10.1 target mep-id 6001
Mon Sep 11 12:12:39.259 UTC
Warning: Burst configuration is present and so this profile cannot be represented in the MEF-SOAM-PM-MIB configuration tables. However, the statistics are still collected
On-demand operation 2 succesfully created / - Completed - statistics will be displayed shortly.
RP/0/RP0/CPU0:ios#show ethernet sla statistics on-demand id 2
Mon Sep 11 12:13:24.825 UTC
Source: Interface TenGigE0/0/0/10.1, Domain DOWN0
Destination: Target MEP-ID 6001
On-demand operation ID #2, packet type 'cfm-synthetic-loss-measurement'
Started at 12:12:41 UTC Mon 11 September 2017, runs once for 10s
Frame Loss Ratio calculated every 10s
One-way Frame Loss (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:12:41 UTC Mon 11 September 2017 lasting 10s
Synthetic Loss Measurement

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

This alternative mechanism does not measure the loss of the actual data traffic, but instead injects synthetic CFM frames and measures the loss of these synthetic frames. You can perform a statistical analysis to give
an approximation of the loss of data traffic. This technique is called Synthetic Loss Measurement (SLM). SLM has been included in the latest version of the Y.1731 standard. Use SLA to perform the following measurements:

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

SLM supports the following:

- All Layer 2 transport interfaces, such as physical, bundle interfaces, Layer2 sub-interfaces, pseudowire Head-end interfaces or attachment circuits. Transport network can be EVPN or BGP-MPLS.
- Up and Down MEPs.
- Transparent passing of the SLM packets through the MIP without punting it to the software.
- 100 concurrent SLM sessions.
- 1000 pps of SLM/SLR traffic.

Configuring Synthetic Loss Measurement

The following section describes how you can configure Synthetic Loss Measurement:

```
RP/0/RP0/CPU0:router (config)ethernet sla
profile test-slm type cfm-synthetic-loss-measurement
  probe
    send packet every 1 seconds
    synthetic loss calculation packets 24
  !
  schedule
    every 3 minutes for 120 seconds
  !
  statistics
    measure one-way-loss-sd
    buckets size 1 probes
    buckets archive 5
  !
    measure one-way-loss-ds
    buckets size 1 probes
    buckets archive 5
  !

interface TenGigE0/0/0/10.1 l2transport
  encapsulation dot1q 1
  ethernet cfm
    mep domain DOWN0 service s10 mep-id 2001
    sla operation profile test-slm target mep-id 6001
```

Configuring an On-Demand Ethernet SLA Operation for CFM Synthetic Loss Measurement

To configure an on-demand Ethernet SLA operation for CFM synthetic loss measurement, use this command in privileged EXEC configuration mode:
RP/0/RP0/CPU0:router# ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mac-address 2.3.4

Running Configuration

RP/0/RP0/CPU0:ios# show ethernet sla statistics on-demand id 1
Mon Sep 11 12:12:00.699 UTC
Source: Interface TenGigE/0/0/10.1, Domain DOWN0
Destination: Target MEP-ID 6001
================================================================================
On-demand operation ID #1, packet type 'cfm-delay-measurement'
RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios# show running-config
Mon Sep 11 12:10:18.467 UTC
Building configuration...
!! IOS XR Configuration version = 6.4.1.14I
!! Last configuration change at Mon Sep 11 12:08:16 2017 by root
!
logging console disable
telnet vrf default ipv4 server max-servers 10
username root
group root-lr
group cisco-support
secret 5 $1$QJT3$94M5/wK5J0v/lpAu/wz31/
!
line console
exec-timeout 0 0
!
ethernet cfm
domain UP6 level 6 id null
  service s6 xconnect group g1 p2p p1 id number 6
    mip auto-create all ccm-learning
    continuity-check interval 1s
    mep crosscheck
    mep-id 4001
  
!

domain DOWN0 level 0 id null
  service s10 down-meps id number 10
    continuity-check interval 1s
    mep crosscheck
    mep-id 6001
  
!
!
profile test-slm type cfm-synthetic-loss-measurement
  probe
    send packet every 1 seconds
    synthetic loss calculation packets 24
  
!
statistics
  measure one-way-loss-sd
    buckets size 1 probes
    buckets archive 5
  measure one-way-loss-ds
    buckets size 1 probes
buckets archive 5
!
interface MgmtEth0/RP0/CPU0/0
shutdown
!
interface TenGigE0/0/0/0
shutdown
!
interface TenGigE0/0/0/1
shutdown
!
interface TenGigE0/0/0/2
shutdown
!
interface TenGigE0/0/0/3
shutdown
!
interface TenGigE0/0/0/4
shutdown
!
interface TenGigE0/0/0/5
shutdown
!
interface TenGigE0/0/0/6
shutdown
!
interface TenGigE0/0/0/7
shutdown
!
interface TenGigE0/0/0/8
shutdown
!
interface TenGigE0/0/0/9
shutdown
!
interface TenGigE0/0/0/10.1 l2transport
encapsulation dot1q 1
ethernet cfm
  mep domain DOWN0 service s10 mep-id 2001
  sla operation profile DMM target mep-id 6001
  sla operation profile test-slm target mep-id 6001
!
!
interface TenGigE0/0/0/11
shutdown
!
interface TenGigE0/0/0/12
shutdown
!
interface TenGigE0/0/0/13
shutdown
!
interface TenGigE0/0/0/14
shutdown
!
interface TenGigE0/0/0/15
shutdown
!
interface TenGigE0/0/0/16
shutdown
!
interface TenGigE0/0/0/17
shutdown
Interface TenGigE0/0/0/18
shutdown
!
Interface TenGigE0/0/0/19
shutdown
!
Interface TenGigE0/0/0/20
shutdown
!
Interface TenGigE0/0/0/21
shutdown
!
Interface TenGigE0/0/0/22
shutdown
!
Interface TenGigE0/0/0/23
shutdown
!
Interface TenGigE0/0/0/24
shutdown
!
Interface TenGigE0/0/0/25
shutdown
!
Interface TenGigE0/0/0/26
shutdown
!
Interface TenGigE0/0/0/27
shutdown
!
Interface TenGigE0/0/0/28
shutdown
!
Interface TenGigE0/0/0/29
shutdown
!
Interface TenGigE0/0/0/30
shutdown
!
!
Interface TenGigE0/0/0/31
shutdown
!
Interface TenGigE0/0/0/32
shutdown
!
Interface TenGigE0/0/0/33
shutdown
!
Interface TenGigE0/0/0/34
shutdown
!
Interface TenGigE0/0/0/35
shutdown
!
Interface TenGigE0/0/0/36
shutdown
!
Interface TenGigE0/0/0/37
shutdown
!
Interface TenGigE0/0/0/38
shutdown
interface TenGigE0/0/0/39
  shutdown
interface TenGigE0/0/1/0/1
  shutdown
interface TenGigE0/0/1/0/2
  shutdown
interface TenGigE0/0/1/0/3
  shutdown
controller Optics0/0/1/0
  breakout 4x10
interface HundredGigE0/0/1/1
  shutdown
interface FortyGigE0/0/1/2.1 l2transport
  encapsulation dot1q 1
  ethernet cfm
    mep domain UP6 service s6 mep-id 1
    sla operation profile DMM target mep-id 6001
    sla operation profile test-slm target mep-id 6001

l2vpn
  xconnect group g1
    p2p p1
      interface TenGigE0/0/0/10.1
      interface FortyGigE0/0/1/2.1

end

Verification

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

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: 0.009ms; Max: 0.010ms; Mean: 0.009ms; StdDev: 0.000ms

One-way Delay (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: 1912765.961ms; Max: 1912765.961ms; Mean: 1912765.961ms; StdDev: -2147483.648ms

One-way Delay (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Bit Error Rate

Bit Error Rate (BER) is a number of bit errors per unit time that determines the reliability of a link. The system supports BER on 10/40/100 GE interfaces. BER allows you to determine if a link is reliable or not.

Following are the two error conditions in BER:

- Signal Degradation (SD)
- Signal Failure (SF)

The SD/SF alarm clearance is automatic once the error value goes down below a certain threshold level. The system uses an error threshold value to measure the errors. In case the error threshold is reached, the system generates IOS messages.

In order to clear an alarm, the system waits till the last poll of window deployment before clearing the alarm. This happens as soon as the error value is below the calculated threshold value. Hence, ensuring that no new errors are accumulated during the last poll of the complete window that might keep the error count above the threshold.
Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

Configuration Examples for EOAM Interfaces

This section provides the following configuration examples:

Configuring an Ethernet OAM Profile Globally: Example

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

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

Configuring Ethernet OAM Features on an Individual Interface: Example

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

```
configure terminal
interface TenGigE 0/1/0/0
ethernet oam
link-monitor
   symbol-period window 60000
   symbol-period threshold ppm low 10000000 high 60000000
   frame window 60
   frame threshold ppm low 10000000 high 60000000
   frame-period window 60000
   frame-period threshold ppm low 100 high 12000000
   frame-seconds window 900000
   frame-seconds threshold low 3 high 900
exit
mib-retrieval
```
Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example

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

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

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

Clearing Ethernet OAM Statistics on an Interface: Example

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

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

Enabling SNMP Server Traps on a Router: Example

This example shows how to enable SNMP server traps on a router:
configure terminal
snmp-server traps ethernet oam events

Configuration Examples for Ethernet CFM

This section includes the following examples:

Ethernet CFM Domain Configuration: Example

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

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

Ethernet CFM Service Configuration: Example

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

```
service Bridge_Service bridge group BD1 bridge-domain B1
service Cross_Connect_1 xconnect group XG1 p2p X1
commit
```

Flexible Tagging for an Ethernet CFM Service Configuration: Example

This example shows how to set the number of tags in CFM packets from down MEPs in a CFM domain service:

```
configure
  ethernet cfm
  domain D1 level 1
  service S2 bridge group BG1 bridge-domain BD2
  tags 1
commit
```

Continuity Check for an Ethernet CFM Service Configuration: Example

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

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

MIP Creation for an Ethernet CFM Service Configuration: Example

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

```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit
```
Cross-check for an Ethernet CFM Service Configuration: Example

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

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

Other Ethernet CFM Service Parameter Configuration: Example

This example shows how to configure other Ethernet CFM service options:

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

MEP Configuration: Example

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

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

Ethernet CFM Show Command: Examples

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

Example 1

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

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

<table>
<thead>
<tr>
<th>Domain/Level</th>
<th>Service</th>
<th>Interface</th>
<th>Type</th>
<th>ID</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>fig/5</td>
<td>bay</td>
<td>Gi0/10/0/12</td>
<td>Dn MEP</td>
<td>2</td>
<td>44:55:66</td>
</tr>
<tr>
<td>fig/5</td>
<td>bay</td>
<td>Gi0/0/1/0</td>
<td>MIP</td>
<td>55:66:77</td>
<td></td>
</tr>
<tr>
<td>fred/3</td>
<td>barney</td>
<td>Gi0/1/0/0</td>
<td>Dn MEP</td>
<td>66:77:88</td>
<td></td>
</tr>
</tbody>
</table>

Example 2

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

```
RP/0/RP0/CPU0:router# show ethernet cfm configuration-errors
```

Domain fig (level 5), Service bay
* MIP creation configured using bridge-domain blort, but bridge-domain blort does not exist.

* An Up MEP is configured for this domain on interface TenGigE0/0/0/3 and an Up MEP is also configured for domain blort, which is at the same level (5).
A MEP is configured on interface TenGigE0/0/0/1 for this domain/service, which has CC interval 100ms, but the lowest interval supported on that interface is 1s

Example 3

This example shows how to display operational state for local maintenance endpoints (MEPs):

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps
```

A - AIS received  I - Wrong interval
R - Remote Defect received  V - Wrong Level
L - Loop (our MAC received)  T - Timed out (archived)
C - Config (our ID received)  M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
P - Peer port down

Domain foo (level 6), Service bar

```
ID Interface (State)  Dir MEPs/Err  RD Defects  AIS
----- ------------------------ --- -------- -- ------- ---
100 Gi1/1/0/1 (Up)  Up  0/0 N A  L7
```

Domain fred (level 5), Service barney

```
ID Interface (State)  Dir MEPs/Err  RD Defects  AIS
----- ------------------------ --- -------- -- ------- ---
2 Gi0/1/0/0 (Up)  Up  3/2 Y RPC  L6
```

Example 4

This example shows how to display operational state of other maintenance endpoints (MEPs) detected by a local MEP:

```
RP/0/RP0/CPU0:router# show ethernet cfm peer meps
```

Flags:
> - Ok  I - Wrong interval
R - Remote Defect received  V - Wrong level
L - Loop (our MAC received)  T - Timed out
C - Config (our ID received)  M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)

Domain fred (level 7), Service barney

```
Down MEP on TenGigE0/0/0/1, MEP-ID 2
```

```
St ID MAC address Port Up/Downtime CcmRcvd SeqErr RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- ----- 
> 1 0011.2233.4455 Up 00:00:01 1234 0 0 0
R> 4 4455.6677.8899 Up 1d 03:04 3456 0 234 0
L 2 1122.3344.5566 Up 3w 1d 6h 3254 0 0 3254
C 2 7788.9900.1122 Test 00:13 2345 6 20 2345
X 3 2233.4455.6677 Up 00:23 30 0 0 30
I 3 3344.5566.7788 Down 00:34 12345 0 300 1234
V 3 8899.0011.2233 Blocked 00:35 45 0 0 45
```
Domain fred (level 7), Service fig
Down MEP on TenGigE0/0/12, MEP-ID 3

<table>
<thead>
<tr>
<th>St</th>
<th>ID MAC address</th>
<th>Port</th>
<th>Up/Downtime</th>
<th>CcmRcvd</th>
<th>SeqErr</th>
<th>RDI Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>1 9900.1122.3344 Up 03:45 4321 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 5

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

```
RP/0/RP0/CPU0:router# show ethernet cfm peer meps detail
```

Domain dom3 (level 5), Service ser3
Down MEP on TenGigE0/0/1 MEP-ID 1

Peer MEP-ID 10, MAC 0001.0203.0403
CFM state: Wrong level, for 00:01:34
Port state: Up
CCM defects detected: V - Wrong Level
CCMs received: 5
Out-of-sequence: 0
Remote Defect received: 5
Wrong Level: 0
Cross-connect (wrong MAID): 0
Wrong Interval: 5
Loop (our MAC received): 0
Config (our ID received): 0
Last CCM received 00:00:06 ago:
  Level: 4, Version: 0, Interval: 1min
  Sequence number: 5, MEP-ID: 10
  MAID: String: dom3, String: ser3
  Port status: Up, Interface status: Up

Domain dom4 (level 2), Service ser4
Down MEP on TenGigE0/0/2 MEP-ID 1

Peer MEP-ID 20, MAC 0001.0203.0402
CFM state: Ok, for 00:00:04
Port state: Up
CCMs received: 7
Out-of-sequence: 1
Remote Defect received: 0
Wrong Level: 0
Cross-connect (wrong MAID): 0
Wrong Interval: 0
Loop (our MAC received): 0
Config (our ID received): 0
Last CCM received 00:00:04 ago:
  Level: 2, Version: 0, Interval: 10s
  Sequence number: 1, MEP-ID: 20
  MAID: String: dom4, String: ser4
  Chassis ID: Local: ios; Management address: 'Not specified'
  Port status: Up, Interface status: Up

Peer MEP-ID 21, MAC 0001.0203.0403
CFM state: Ok, for 00:00:05
Port state: Up
AIS for CFM Configuration: Examples

Example 1

This example shows how to configure Alarm Indication Signal (AIS) transmission for a CFM domain service:

```
RP/0/RP0/CPU0# configure
RP/0/RP0/CPU0(config)# ethernet cfm
RP/0/RP0/CPU0(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
RP/0/RP0/CPU0(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```

Example 2

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

```
RP/0/RP0/CPU0# configure
RP/0/RP0/CPU0(config)# ethernet cfm
RP/0/RP0/CPU0(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p X1
RP/0/RP0/CPU0(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```
AIS for CFM Show Commands: Examples

This section includes the following examples:

show ethernet cfm interfaces ais Command: Example

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

show ethernet cfm local meps Command: Examples

Example 1: Default

This example shows how to display statistics for local maintenance end points (MEPs):
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
P - Peer port down

Domain foo (level 6), Service bar
<table>
<thead>
<tr>
<th>ID</th>
<th>Interface (State)</th>
<th>Dir MEPs/Err RD Defects AIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Gi1/1/0/1 (Up)</td>
<td>Up 0/0 N A 7</td>
</tr>
</tbody>
</table>

Domain fred (level 5), Service barney
<table>
<thead>
<tr>
<th>ID</th>
<th>Interface (State)</th>
<th>Dir MEPs/Err RD Defects AIS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gi0/1/0/0 (Up)</td>
<td>Up 3/2 Y RPC 6</td>
</tr>
</tbody>
</table>

Example 2: Domain Service
This example shows how to display statistics for MEPs in a domain service:

RP/0/RP0/CPU0:router# show ethernet cfm local meps domain foo service bar detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100

<table>
<thead>
<tr>
<th>Interface state: Up</th>
<th>MAC address: 1122.3344.5566</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)</td>
<td></td>
</tr>
<tr>
<td>CCM generation enabled: No</td>
<td></td>
</tr>
<tr>
<td>AIS generation enabled: Yes (level: 7, interval: 1s)</td>
<td></td>
</tr>
<tr>
<td>Sending AIS: Yes (started 01:32:56 ago)</td>
<td></td>
</tr>
<tr>
<td>Receiving AIS: Yes (from lower MEP, started 01:32:56 ago)</td>
<td></td>
</tr>
</tbody>
</table>

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2

<table>
<thead>
<tr>
<th>Interface state: Up</th>
<th>MAC address: 1122.3344.5566</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)</td>
<td></td>
</tr>
<tr>
<td>Cross-check defects: 0 missing, 0 unexpected</td>
<td></td>
</tr>
<tr>
<td>CCM generation enabled: Yes (Remote Defect detected: Yes)</td>
<td></td>
</tr>
<tr>
<td>CCM defects detected: R - Remote Defect received</td>
<td></td>
</tr>
<tr>
<td>P - Peer port down</td>
<td></td>
</tr>
<tr>
<td>C - Config (our ID received)</td>
<td></td>
</tr>
<tr>
<td>AIS generation enabled: Yes (level: 6, interval: 1s)</td>
<td></td>
</tr>
<tr>
<td>Sending AIS: Yes (to higher MEP, started 01:32:56 ago)</td>
<td></td>
</tr>
<tr>
<td>Receiving AIS: No</td>
<td></td>
</tr>
</tbody>
</table>

Example 4: Detail
This example shows how to display detailed statistics for MEPs in a domain service:

RP/0/RP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100

<table>
<thead>
<tr>
<th>Interface state: Up</th>
<th>MAC address: 1122.3344.5566</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)</td>
<td></td>
</tr>
<tr>
<td>CCM generation enabled: No</td>
<td></td>
</tr>
<tr>
<td>AIS generation enabled: Yes (level: 7, interval: 1s)</td>
<td></td>
</tr>
<tr>
<td>Sending AIS: Yes (started 01:32:56 ago)</td>
<td></td>
</tr>
<tr>
<td>Receiving AIS: Yes (from lower MEP, started 01:32:56 ago)</td>
<td></td>
</tr>
</tbody>
</table>
Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
==================================================================================================
Interface state: Up  MAC address: 1122.3344.5566
Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
Cross-check defects: 0 missing, 0 unexpected

CCM generation enabled: Yes (Remote Defect detected: Yes)
CCM defects detected:  R - Remote Defect received
                      P - Peer port down
                      C - Config (our ID received)
AIS generation enabled: Yes (level: 6, interval: 1s)
Sending AIS: Yes (to higher MEP, started 01:32:56 ago)
Receiving AIS: No

show ethernet cfm local meps detail Command: Example

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

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps detail
Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100
==================================================================================================
Interface state: Up  MAC address: 1122.3344.5566
Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)
Cross-check errors: 2 missing, 0 unexpected

CCM generation enabled: No
AIS generation enabled: Yes (level: 7, interval: 1s)
Sending AIS: Yes (started 01:32:56 ago)
Receiving AIS: Yes (from lower MEP, started 01:32:56 ago)
EFD triggered: Yes

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
==================================================================================================
Interface state: Up  MAC address: 1122.3344.5566
Peer MEPs: 3 up, 0 with errors, 0 timed out (archived)
Cross-check errors: 0 missing, 0 unexpected

CCM generation enabled: Yes (Remote Defect detected: No)
AIS generation enabled: Yes (level: 6, interval: 1s)
Sending AIS: No
Receiving AIS: No
EFD triggered: No
```

You can also verify that EFD has been triggered on an interface using the `show interfaces` and `show interfaces brief` commands. When an EFD trigger has occurred, these commands will show the interface status as `up` and the line protocol state as `down`.  

---

Note

You can also verify that EFD has been triggered on an interface using the `show interfaces` and `show interfaces brief` commands. When an EFD trigger has occurred, these commands will show the interface status as `up` and the line protocol state as `down`.  

---

Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.1.x
CFM Adaptive Bandwidth Notifications

Microwave links are used in carrier ethernet networks, because they are cheaper than laying fibre either in dense metro areas or rural locations. However, the disadvantage of microwave links is that the signal is affected by atmospheric conditions and can degrade.

Modern microwave devices support adaptive modulation schemes to prevent a complete loss of signal. This allows them to continue to operate during periods of degradation, but at a reduced bandwidth. However, to fully take advantage of this, it's necessary to convey the decrease in bandwidth to the head-end router so that appropriate actions can be taken. Otherwise, the link may become saturated and traffic dropped arbitrarily as shown in the following figure:

A generic solution to this is a Connectivity Fault Management (CFM) extension to send Bandwidth Notifications Messages (BNM) to Maintenance Endpoints (MEPs) on the corresponding interface on the head-end router. To be flexible in the actions taken, the choice of solution uses Embedded Event Manager (EEM) to invoke operator written TCL scripts. For information on EEM, see Embedded Event Manager, on page 129.

Bandwidth Notification Messages

The two types of messages used to notify the head-end router are:

- G.8013 Bandwidth Notification Messages (G.8013 BNM)
- Cisco proprietary Bandwidth Vendor-Specific Messages (Cisco BW-VSM)

Both the message types contain the following information:

- Source MAC
- Port ID
- Maintenance Domain (MD) Level
- Transmission period
- Nominal Bandwidth
- Current Bandwidth
During signal degradation, periodic BNMs are sent to the head-end router containing the current bandwidth (sampled over a period of time) and nominal bandwidth (full bandwidth when there is no degradation). This allows the router to reduce the bandwidth directed to the link as shown in the figure below:

When degradation in bandwidth is detected, depending on the topology, the degradation may affect one or more paths in the network. Therefore, in more complex topologies, the head-end router may need information about links in each affected path. The BNM transmission period and a Link ID are used to differentiate between messages from the same source MAC address which refer to different links.

**Restrictions for CFM Bandwidth Notifications**

The list of restrictions for CFM Bandwidth Notifications is:

- Up to 200 unique BNM enabled links learnt from BNMs are supported per line card. Any BNMs for links over this limit will be discarded.

  To reset CFM BNM enabled links for the specified interfaces, use the `clear ethernet cfm interface [ <interface> ] bandwidth-notifications { all | state <state> } | location { all | <node> } ]` command. An archive timer is used to clean up any BNM enabled links whose loss timer expired at least 24 hours ago.

- Over process restart:
  - Loss threshold, wait-to-restore, and hold-off timers are restarted. This may cause links to take longer to transition between states than they would have otherwise.
  - Archive timers are restarted. This may cause historical statistics for links to persist longer than they would have otherwise.
  - Queued events for EEM scripts which have been rate-limited are not preserved. Scripts with at least one link in DEGRADED state, or BNMs have changed over process restart, and are invoked. Rate-limit timers are restarted. This may cause scripts to be invoked when they would otherwise have been filtered by the damping or conformance-testing algorithms. If the last link returns to its nominal bandwidth within the rate-limit period but before the process restart, then the script will not be invoked after the process restart. Thus, actions taken by the script may not reflect the (increased) latest bandwidths of any links which returned to their nominal bandwidths within the rate-limit period.
Bandwidth Reporting

Received BNMss are used to identify BNM enabled links within a Maintenance Entity Group (MEG), and should be uniquely identifiable within the MEG by Port-ID or MAC address. Each link has an associated nominal bandwidth, and a Reported Bandwidth (RBW), which are notified to the operator. The link is considered to be in OK state when the RBW is equal to the nominal bandwidth and DEGRADED if RBW is less than nominal.

Devices sending BNMs can detect changes in bandwidth many times a second. For example, changes caused by an object passing through a microwave link’s line of sight. The protocol for sending BNMs is designed to mitigate fluctuating current bandwidth by sampling across a ‘monitoring-interval’ and applying basic damping to degradation events. To help mitigate this further, a damping algorithm is used. This algorithm is applied on the receiving device, and is distinct from any damping performed by the sender. For more information on this, see Damping Algorithm, on page 127.

An operator may be interested in more than one BNM enabled link, and needs the ability to register on a set of BNM enabled links which affect the path to a node in the network. To do this, the state and RBW for each link of interest are put into a conformance testing algorithm, which both filters and rate-limits changes to publish events notifying the operator only of significant changes. For more information on this, see Conformance Testing Algorithm, on page 129.

The following diagram shows how a received BNM flows through the damping and conformance testing algorithm to invoke operator scripts:

---

**Note**

- Port ID takes precedence over MAC address. This means that BNMs with same port ID but different MAC addresses are counted as same BNMs.
- If BNM reported bandwidth is equal to the threshold, then EEM will not be invoked.
- If a degraded link having bandwidth higher than the threshold receives BNM with bandwidth less than the threshold, it doesn't wait for the hold-off timer and instantly changes the bandwidth by invoking EEM script.

---

**Damping Algorithm**

A damping algorithm is applied to each unique BNM enabled link for which BNMs are received. The table below describes the timers used for this purpose:
### Timers

<table>
<thead>
<tr>
<th><strong>Timers</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>loss threshold (in packet numbers)</td>
<td>This timer handles the case when BNMs stop being received. This timer is (re)started whenever any BNM is received for the link. The value is equal to the expected period between BNMs (as indicated by the last BNM) multiplied by the configured loss threshold. When the loss threshold timer expires, as the link may have changed or been removed entirely, bandwidth information is no longer available, therefore the link is considered to have been restored to its previously notified nominal bandwidth.</td>
</tr>
<tr>
<td>hold-off (in seconds)</td>
<td>This timer is used to damp transient transitions from OK to DEGRADED state. It is started when the first degraded BNM is received, and is stopped if the loss threshold timer expires or the current bandwidth returns to the nominal bandwidth. If the timer expires, then the BNM enabled link enters DEGRADED state. The value of this timer is configurable. If it is zero, then the link immediately enters degraded state and the timer is not started.</td>
</tr>
<tr>
<td>wait-to-restore (WTR, in seconds)</td>
<td>This timer is used to damp transient transitions from DEGRADED to OK state. It is started when a BNM Enabled Link is in DEGRADED state and either the loss threshold timer expires or a BNM is received that indicates the current bandwidth has returned to the nominal bandwidth. If a degraded BNM is received while the timer is running then it is stopped and the BNM Enabled Link remains in DEGRADED state. If this timer expires then the link returns to OK state.</td>
</tr>
</tbody>
</table>

The following internal state transition diagram shows how damping algorithm takes place:
**Conformance Testing Algorithm**

The conformance testing algorithm comprises of two parts:

1. Filtering bandwidth changes.

   Filtering is done so that events are published whenever either:

   • Any link which was in OK state or had a RBW more than or equal to the specified threshold, has transitioned to DEGRADED state and has a RBW less than the specified threshold.

   • Any link which was in DEGRADED state and had a RBW less than the specified threshold, is still in DEGRADED state and has a RBW less than the specified threshold, but the old and new RBWs are different.

   • Any link which was in DEGRADED state and had a RBW less than the specified threshold, has transitioned to OK state or has a RBW more than or equal to the specified threshold.

2. Rate-limiting bandwidth changes

   Rate-limiting is done by only publishing events at most once within any rate-limit period. If there is a change in bandwidth (which passes the filter) within this rate-limit period, a timer is started to expire at the end of the period. Upon timer expiry, an event is published which reflects the latest state and bandwidth of all links of interest which are in DEGRADED state.

**Embedded Event Manager**

The Embedded Event Manager (EEM) consists of an EEM server that monitors various real-time events in the system using programs called Event Detectors (EDs) and triggers registered policies (for example, TCL scripts) to run. The EEM supports at least 200 script registrations.
Typical actions taken in response to signal degradation events include:

- Signaling to G.8032 to switch some flows to alternative paths
- Modifying QoS configuration to adjust traffic shaping to the new bandwidth
- Adjusting IGP metrics to switch some traffic to an alternative path

The following variables can be queried within the TCL script:

<table>
<thead>
<tr>
<th>EEM Variables</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface, level, direction</td>
<td>Identify the MEP in the registration</td>
</tr>
<tr>
<td>min_reported_bandwidth</td>
<td>Minimum reported bandwidth across all links in the registration that are currently in DEGRADED state, and below the specified threshold</td>
</tr>
</tbody>
</table>
| bnm_enabled_links [ { MAC address | Port ID | } ] | Array of BNM enabled links, with each one containing the following elements:  
  * reported_bw: Reported Bandwidth  
  * nominal_bw: Nominal BW in last BNM |
| event_type | Either ‘DEGRADED’ or ‘OK’  
DEGRADED indicates that at least one BNM enabled link in the registration is in DEGRADED state with a reported bandwidth less than the threshold.  
This means that the event_type could be ‘OK’ if all BNM enabled links in the registration which are in DEGRADED state have a reported bandwidth greater than or equal to the threshold. |

The command for EEM TCL scripts registering for CFM Bandwidth Notification events is:  
```
interface <interface name> level <level> direction <direction> {mac-addresses { <addr1> [, ..., <addr20>] } | port-ids { <id1> [, ..., <id20>] } } threshold <bandwidth> [ ratelimit <time> ].
```

To configure EEM, use the following commands:

```
event manager directory user policy disk0:/
event manager directory user library disk0: /
event manager policy EEMscript7.tcl username root persist-time 3600
aaa authorization eventmanager default local
```

Individual scripts located in the specified directory can then be configured with:

```
event manager policy <script_name> username lab persist-time <time>
```

**Event Publishing**

CFM publishes events for a given EEM registration after applying the damping and conformance testing algorithms as described in Damping Algorithm, on page 127 and Conformance Testing Algorithm, on page 129 respectively. The set of BNM Enabled Links published in an event are those in DEGRADED state and whose RBW is less than the specified threshold.
Configure CFM Bandwidth Notifications

Use the following steps to configure CFM bandwidth notifications:

- Configure a CFM domain at the level BNM are expected to be received at, and a CFM service in the direction (either up or down-MEPs) the BNM are expected to be received.

- Configure a CFM MEP on the interface expected to receive BNM in the domain and service above.

Configuration consists of two parts:

- Configuring global CFM. This is similar to Continuity Check Message (CCM) and other CFM configurations.

  **Global CFM configuration:**

  ```
  ethernet cfm
  domain DM1 level 2 id null
  service SR1 down-meps
  
  domain dom1 level 1
  service ser1 down-meps
  
  
  
  
  
  
  
  
  
  ```

- Configuration related to CFM-BNM under interfaces. This is optional and used for changing default values.

  **Interface configuration:**

  Interface TenGigE0/0/1/1

  ```
  ethernet cfm
  mep domain DM1 service SR1 mep-id 3001
  
  bandwidth-notifications
  hold-off 0
  wait-to-restore 60
  loss-threshold 10
  log changes
  
  ```

  Interface TenGigE0/0/0/3

  ```
  ethernet cfm
  mep domain dom1 service ser1 mep-id 11
  
  bandwidth-notifications
  hold-off 10
  wait-to-restore 40
  log changes
  
  ```

**Running Configuration**

```
RP/0/RP0/CPU0:router#show running-configuration
!! IOS XR Configuration 7.1.1.104I
```
!! Last configuration change at Mon Jun 24 21:26:46 2019 by root
!
hostname R2_cXR
logging console debugging
logging buffered 125000000
event manager directory user policy harddisk:/tcl/
event manager directory user library harddisk:/tcl/
event manager policy EEMmac_lev1.tcl username root persist-time 3600
event manager policy EEMport_lev1.tcl username root persist-time 3600
aaa authorization exec default local group tacacs+
aaa authorization eventmanager default local
!
eternet cfm
domain DM0 level 1 id null
  service SR0 down-meps
    continuity-check interval 1m
    mep crosscheck
    mep-id 1003
  !
  ais transmission interval 1s cos 4
  log ais
  log continuity-check errors
  log crosscheck errors
  log continuity-check mep changes
  !
domain DM1 level 2 id null
  service SR1 down-meps id number 1
    continuity-check interval 1m
    mep crosscheck
    mep-id 431
  !
  ais transmission interval 1m
  log ais
  log continuity-check errors
  log crosscheck errors
  log continuity-check mep changes
  !
domain dom1 level 3 id string domain3
  service ser1 xconnect group XG1 p2p XC1 id number 2300
    mip auto-create all
    continuity-check interval 1m
    mep crosscheck
    mep-id 2030
  !
interface Loopback0
  ipv4 address 30.30.30.30 255.255.255.255
  !
interface MgmtEth0/RSP0/CPU0/0
  ipv4 address 5.18.9.102 255.255.0.0
  !
interface MgmtEth0/RSP0/CPU0/1
  shutdown
  !
interface TenGigE0/0/0/0
  shutdown
  !
interface TenGigE0/0/0/3.1 l2transport
  encapsulation dot1q 6
ethernet cfm
  mep domain DM1 service SR1 mep-id 231
  !
  bandwidth-notifications
  hold-off 50
configure ethernet oam

wait-to-restore 50
loss-threshold 100
log changes
!

Verification

RP/0/RP0/CPU0:router#show ethernet cfm interfaces bandwidth-notifications detail
BNM Enabled Links at Level 3 (Down MEP) for GigabitEthernet/1
MAC Address 000a.000a.000a
State (OK):
Nominal Bandwidth: 3000 Mbps
Reported Bandwidth: 1000 Mbps
Elapsed time in this state: 00:00:13.000
Transitions into degraded state: 5000
Hold-off: 111s remaining
Last BNM received 00:00:10 ago
Nominal Bandwidth: 1000 Mbps
Current Bandwidth: 2000 Mbps
Interval: 10s
Packet-type: Cisco BW-VSM
Packets received: 20000

Port ID 7 (MAC Address 000c.000c.000c)
State (DEGRADED):
Nominal Bandwidth: 6000 Mbps
Reported Bandwidth: 2000 Mbps
Elapsed time in this state: 00:00:39.000
Transitions into degraded state: 10000
Wait-to-restore: 111s remaining
Last BNM received 00:00:33 ago
Nominal Bandwidth: 2000 Mbps
Current Bandwidth: 4000 Mbps
Interval: 1min
Packet-type: Cisco BW-VSM
Packets received: 40000

Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 7.1.x
Configure CFM Bandwidth Notifications
Configuring Integrated Routing and Bridging

This module describes the configuration of Integrated Routing and Bridging (IRB). IRB provides the ability to exchange traffic between bridging services and a routed interface using a Bridge-Group Virtual Interface (BVI).

Feature History for IRB

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.1.1</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- IRB Introduction, on page 135
- Bridge-Group Virtual Interface, on page 136
- Supported Features on a BVI, on page 136
- BVI Interface and Line Protocol States, on page 137
- Prerequisites for Configuring IRB, on page 137
- Restrictions for Configuring IRB, on page 138
- How to Configure IRB, on page 139
- Additional Information on IRB, on page 146
- Packet Flows Using IRB, on page 146
- Configuration Examples for IRB, on page 147

IRB Introduction

IRB provides the ability to route between a bridge group and a routed interface using a BVI. The BVI is a virtual interface within the router that acts like a normal routed interface. A BVI is associated with a single bridge domain and represents the link between the bridging and the routing domains on the router. To support receipt of packets from a bridged interface that are destined to a routed interface, the BVI must be configured with the appropriate IP addresses and relevant Layer 3 attributes.
Bridge-Group Virtual Interface

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.

BVI supports only Layer 3 attributes, and has the following characteristics:

- Uses a MAC address taken from the local chassis MAC address pool, unless overridden at the BVI interface.
- Is configured as an interface type using the interface bvi command and uses an IPv4 address that is in the same subnet as the hosts on the segments of the bridged domain. The BVI also supports secondary addresses.
- The BVI identifier is independent of the bridge-domain identifier. These identifiers do not need to correlate like they do in Cisco IOS software.
- Is associated to a bridge group using the routed interface bvi command.
- BVI interfaces support a number range of 1 to 4294967295.

Supported Features on a BVI

- These interface commands are supported on a BVI:
  - arp purge-delay
• arptimeout
• bandwidth (The default is 10 Gbps and is used as the cost metric for routing protocols for the BVI)
• ipv4
• ipv6
• mac-address
• mtu (The default is 1500 bytes)

Note  MTU is not supported on Release 6.3.x and Release 6.5.x.

• shutdown

• The BVI supports IP helper addressing and secondary IP addressing.

### BVI Interface and Line Protocol States

Like typical interface states on the router, a BVI has both an Interface and Line Protocol state.

• The BVI interface state is Up when the following occurs:
  • The BVI interface is created.
  • The bridge-domain that is configured with the `routed interface bvi` command has at least one available active bridge port (Attachment circuit [AC] or pseudowire [PW]).

Note  A BVI will be moved to the Down state if all of the bridge ports (Ethernet flow points [EFPs]) associated with the bridge domain for that BVI are down. However, the BVI will remain up if at least one bridge port is up, even if all EFPs are down.

• These characteristics determine when the the BVI line protocol state is up:
  • The bridge-domain is in Up state.
  • The BVI IP address is not in conflict with any other IP address on another active interface in the router.

### Prerequisites for Configuring IRB

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 IRB, be sure that these tasks and conditions are met:
• Know the IP addressing and other Layer 3 information to be configured on the bridge virtual interface (BVI).

• Complete MAC address planning if you decide to override the common global MAC address for all BVIs.

• Be sure that the BVI network address is being advertised by running static or dynamic routing on the BVI interface.

Restrictions for Configuring IRB

Before configuring IRB, consider these restrictions:

• Only one BVI can be configured in any bridge domain.

• The same BVI can not be configured in multiple bridge domains.

• The following areas are not supported on the BVI:
  • Access Control Lists (ACLs). However, Layer 2 ACLs can be configured on each Layer 2 port of the bridge domain.
  • IP fast reroute (FRR)
  • TI-LFA
  • SR
  • LDP
  • NetFlow
  • MoFRR
  • Quality of Service (QoS)
  • Traffic mirroring
  • Unnumbered interface for BVI
  • Video monitoring (Vidmon)
  • IRB with 802.1ah (BVI and Provider Backbone Bridge (PBB) should not be configured in the same bridge domain).
  • PIM snooping. (Need to use selective flood.)
  • VRF-aware DHCP relay

• The following areas are not supported on the Layer2 bridging (with BVI):
  • Static mac entry configuration in Bridge.
  • Mac ageing configuration at global config mode.
  • MAC Learning Disable.
  • Vlan rewrite.
• QOS configuration on BVI interface is not supported for egress.
• Label allocation mode per-CE with BVI is not supported in an access network along with PE-CE protocols enabled.

How to Configure IRB

This section includes the following configuration tasks:

Configuring the Bridge Group Virtual Interface

To configure a BVI, complete the following steps.

Configuration Guidelines

Consider the following guidelines when configuring the BVI:

• The BVI must be assigned an IPv4 or IPv6 address that is in the same subnet as the hosts in the bridged segments.
• If the bridged network has multiple IP networks, then the BVI must be assigned secondary IP addresses for each network.

SUMMARY STEPS

1. configure
2. interface bvi identifier
3. ipv4 address ipv4-address mask [secondary] ipv6 address ipv6-prefix/prefix-length [eui-64] [route-tag route-tag value]
4. arp purge-delay seconds
5. arp timeout seconds
6. bandwidth rate
7. mtu bytes
8. end or commit

DETAILED STEPS

Step 1 configure
Example:

Router# configure
Enters the global configuration mode.

Step 2 interface bvi identifier
Example:
Router(config)# interface bvi 1

Specifies or creates a BVI, where identifier is a number from 1 to 65535.

**Step 3** ipv4 address ipv4-address mask [secondary] ipv6 address ipv6-prefix/prefix-length [eui-64] [route-tag route-tag value]

*Example:*

Router(config-if)# ipv4 address 10.10.0.4 255.255.255.0

Specifies a primary or secondary IPv4 address or an IPv6 address for an interface.

**Step 4** arp purge-delay seconds

*Example:*

Router(config-if)#arp purge-delay 120

(Optional) Specifies the amount of time (in seconds) to delay purging of Address Resolution Protocol (ARP) table entries when the interface goes down.

The range is 1 to 65535. By default purge delay is not configured.

**Step 5** arp timeout seconds

*Example:*

Router(config-if)# arp timeout 12200

(Optional) Specifies how long dynamic entries learned on the interface remain in the ARP cache.

The range is 30 to 2144448000 seconds. The default is 14,400 seconds (4 hours).

**Step 6** bandwidth rate

*Example:*

Router(config-if)# bandwidth 1000000

(Optional) Specifies the amount of bandwidth (in kilobits per second) to be allocated on the interface. This number is used as the cost metric in routing protocols for the BVI.

The range is 0 to 4294967295. The default is 10000000 (10 Gbps).

**Step 7** mtu bytes

*Example:*

Router(config-if)# mtu 2000

(Optional) Specifies the maximum transmission unit (MTU) size for packets on the interface. The range is 64 to 65535. The default is 1514.

*Note* MTU is not supported on Release 6.3.x and Release 6.5.x.

**Step 8** end or commit

*Example:*

Router(config-if)# end
or

Router(config-if)# commit

Saves configuration changes.

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

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

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

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

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

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

---

**Configuring the Layer 2 AC Interfaces**

To configure the Layer 2 AC interfaces for routing by a BVI, complete the following steps.

**SUMMARY STEPS**

1. **configure**
2. **interface** [HundredGigE | TenGigE] l2transport
3. **end** or **commit**

**DETAILED STEPS**

**Step 1**

**configure**

**Example:**

RP/0/RP0/CPU0:router# configure

Enters global configuration mode.

**Step 2**

**interface** [HundredGigE | TenGigE] l2transport

**Example:**

RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0.1 l2transport

Enables Layer 2 transport mode on a Gigabit Ethernet or 10-Gigabit Ethernet interface or subinterface and enters interface or subinterface configuration mode.
Step 3  end or commit  

Example:  

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

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

Saves configuration changes.  

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

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

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

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

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

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

---

Configuring a Bridge Group and Assigning Interfaces to a Bridge Domain  

To configure a bridge group and assign interfaces to a bridge domain, complete the following steps.  

SUMMARY STEPS  

1. configure  
2. l2vpn  
3. bridge group bridge-group-name  
4. bridge-domain bridge-domain-name  
5. interface [HundredGigE | TenGigE]  
6. end or commit  

DETAILED STEPS  

Step 1  configure  

Example:  

RP/0/RP0/CPU0:router# configure  

Enters global configuration mode.
Step 2  l2vpn
Example:

RP/0/RP0/CPU0:router(config)# l2vpn
Enters L2VPN configuration mode.

Step 3  bridge group **bridge-group-name**
Example:

RP/0/RP0/CPU0:router(config-l2vpn)# bridge group 10
Creates a bridge group and enters L2VPN bridge group configuration mode.

Step 4  bridge-domain **bridge-domain-name**
Example:

RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain BD_1
Creates a bridge domain and enters L2VPN bridge group bridge domain configuration mode.

Step 5  interface [HundredGigE | TenGigE]
Example:

RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# interface HundredGigE 0/1/0/0.1
Associates the 100-Gigabit Ethernet or 10-Gigabit Ethernet interface with the specified bridge domain and enters L2VPN bridge group bridge domain attachment circuit configuration mode.

Repeat this step for as many interfaces as you want to associate with the bridge domain.

Step 6  end or commit
Example:

RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-ac)# end
or

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

### Associating the BVI as the Routed Interface on a Bridge Domain

To associate the BVI as the routed interface on a bridge domain, complete the following steps.

**SUMMARY STEPS**

1. `configure`
2. `l2vpn`
3. `bridge group bridge-group-name`
4. `bridge-domain bridge-domain-name`
5. `routed interface bvi identifier`
6. `end` or `commit`

**DETAILED STEPS**

**Step 1**  
**configure**  
**Example:**

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

Enters global configuration mode.

**Step 2**  
**l2vpn**  
**Example:**

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

Enters L2VPN configuration mode.

**Step 3**  
**bridge group bridge-group-name**  
**Example:**

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

Creates a bridge group and enters L2VPN bridge group configuration mode.

**Step 4**  
**bridge-domain bridge-domain-name**  
**Example:**

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain 1
```

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

**Step 5**  
**routed interface bvi identifier**
Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# routed interface bvi 1
```
Associates the specified BVI as the routed interface for the interfaces assigned to the bridge domain.

**Step 6**  
end or commit

Example:

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

or

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

---

**Displaying Information About a BVI**

To display information about BVI status and packet counters, use the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`show interfaces bvi identifier [accounting</td>
<td>brief</td>
</tr>
<tr>
<td>`show adjacency bvi identifier [detail</td>
<td>remote]`</td>
</tr>
<tr>
<td><code>show l2vpn bridge-domain detail</code></td>
<td>Displays the reason that a BVI is down.</td>
</tr>
</tbody>
</table>
Additional Information on IRB

Packet Flows Using IRB

This figure shows a simplified functional diagram of an IRB implementation to describe different packet flows between Host A, B, and C. In this example, Host C is on a network with a connection to the same router. In reality, another router could be between Host C and the router shown.

**Figure 10: IRB Packet Flows Between Hosts**

When IRB is configured on a router, the following processing happens:

- ARP requests are resolved between the hosts and BVI that are part of the bridge domain.
- All packets from a host on a bridged interface go to the BVI if the destination MAC address matches the BVI MAC address. Otherwise, the packets are bridged.
- For packets destined for a host on a routed network, the BVI forwards the packets to the routing engine before sending them out a routed interface.
- All packets either from or destined to a host on a bridged interface go to the BVI first (unless the packet is destined for a host on the bridge domain).
- For packets that are destined for a host on a segment in the bridge domain that come in to the router on a routed interface, the BVI forwards the packet to the bridging engine, which forwards it through the appropriate bridged interface.

Packet Flows When Host A Sends to Host B on the Bridge Domain

When Host A sends data to Host B in the bridge domain on the 10.10.0.0 network, no routing occurs. The hosts are on the same subnet and the packets are bridged between their segment interfaces on the router.
Packet Flows When Host A Sends to Host C From the Bridge Domain to a Routed Interface

Using host information from this figure, the following occurs when Host A sends data to Host C from the IRB bridging domain to the routing domain:

- Host A sends the packet to the BVI (as long any ARP request the is resolved between the host and the BVI). The packet has the following information:
  - Source MAC address of host A.
  - Destination MAC address of the BVI.

- Since Host C is on another network and needs to be routed, the BVI forwards the packet to the routed interface with the following information:
  - IP source MAC address of Host A (10.10.0.2) is changed to the MAC address of the BVI (10.10.0.4).
  - IP destination address is the IP address of Host C (10.20.0.3).

- Interface 10.20.0.2 sees receipt of a packet from the routed BVI 10.10.0.4. The packet is then routed through interface 10.20.0.2 to Host C.

Packet Flows When Host C Sends to Host B From a Routed Interface to the Bridge Domain

Using host information from this figure, the following occurs when Host C sends data to Host B from the IRB routing domain to the bridging domain:

- The packet comes into the routing domain with the following information:
  - MAC source address—MAC of Host C.
  - MAC destination address—MAC of the 10.20.0.2 ingress interface.
  - IP source address—IP address of Host C (10.20.0.3).
  - IP destination address—IP address of Host B (10.10.0.3).

- When interface 10.20.0.2 receives the packet, it looks in the routing table and determines that the packet needs to be forwarded to the BVI at 10.10.0.4.

- The routing engine captures the packet that is destined for the BVI and forwards it to the BVI’s corresponding bridge domain. The packet is then bridged through the appropriate interface if the destination MAC address for Host B appears in the bridging table, or is flooded on all interfaces in the bridge group if the address is not in the bridging table.

Configuration Examples for IRB

This section provides the following configuration examples:
Basic IRB Configuration: Example

The following example shows how to perform the most basic IRB configuration:

! Configure the BVI and its IPv4 address

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)#interface bvi 1
RP/0/RP0/CPU0:router(config-if)#ipv4 address 10.10.0.4 255.255.255.0
RP/0/RP0/CPU0:router(config-if)# exit
!
! Configure the Layer 2 AC interface
!
RP/0/RP0/CPU0:router(config)#interface HundredGigE 0/1/0/0 l2transport
RP/0/RP0/CPU0:router(config-if)# exit
!
! Configure the L2VPN bridge group and bridge domain and assign interfaces
!
RP/0/RP0/CPU0:router(config)#l2vpn
RP/0/RP0/CPU0:router(config-l2vpn)#bridge group 10
RP/0/RP0/CPU0:router(config-l2vpn-bd)#bridge-domain 1
RP/0/RP0/CPU0:router(config-l2vpn-bd-bd)#interface HundredGigE 0/1/0/0
RP/0/RP0/CPU0:router(config-l2vpn-bd-bd-if)# exit
!
! Associate a BVI to the bridge domain
!
RP/0/RP0/CPU0:router(config-l2vpn-bd-bd)# routed interface bvi 1
RP/0/RP0/CPU0:router(config-l2vpn-bd-bd)# commit

IPv4 Addressing on a BVI Supporting Multiple IP Networks: Example

The following example shows how to configure secondary IPv4 addresses on a BVI that supports bridge domains for the 10.10.10.0/24, 10.20.20.0/24, and 10.30.30.0/24 networks. In this example, the BVI must have an address on each of the bridge domain networks:

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)#interface bvi 1
RP/0/RP0/CPU0:router(config-if)#ipv4 address 10.10.0.4 255.255.255.0
RP/0/RP0/CPU0:router(config-if)#ipv4 address 10.20.0.4 255.255.255.0 secondary
RP/0/RP0/CPU0:router(config-if)#ipv4 address 10.30.0.4 255.255.255.0 secondary
RP/0/RP0/CPU0:router(config-if)# commit

IRB With BVI and VRRP Configuration: Example

This example shows a partial router configuration for the relevant configuration areas for IRB support of a BVI and VRRP:

VRRPv6 is also supported.

l2vpn
bridge group IRB
bridge-domain IRB-EDGE
interface TenGigE0/0/8
! routed interface BVI 100
!
interface TenGigE0/0/0/8
  12transport
!
interface BVI 100
  ipv4 address 10.21.1.1 255.255.255.0
!
router vrrp
  interface BVI 100
  address-family ipv4
  vrrp 1
    address 10.21.1.100
    priority 100
!
Configuring Link Bundling

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

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

A link bundle is simply a group of ports that are bundled together and act as a single link. The advantages of link bundles are as follows:

- Multiple links can span several line cards to form a single interface. Thus, the failure of a single link does not cause a loss of connectivity.
- Bundled interfaces increase bandwidth availability, because traffic is forwarded over all available members of the bundle. Therefore, traffic can flow on the available links if one of the links within a bundle fails. Bandwidth can be added without interrupting packet flow.

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

Cisco IOS XR software supports the following method of forming bundles of Ethernet interfaces:

- IEEE 802.3ad—Standard technology that employs a Link Aggregation Control Protocol (LACP) to ensure that all the member links in a bundle are compatible. Links that are incompatible or have failed are automatically removed from a bundle.

- Limitations and Compatible Characteristics of Ethernet Link Bundles, on page 152
- Configuring Ethernet Link Bundles, on page 153
- Configuring LACP Fallback, on page 157
- Configuring EFP Load Balancing on an Ethernet Link Bundle, on page 158
- VLANs on an Ethernet Link Bundle, on page 161
- Configuring VLAN over Bundles, on page 162
- LACP Short Period Time Intervals, on page 166
- Configuring the Default LACP Short Period Time Interval, on page 167
- Configuring Custom LACP Short Period Time Intervals, on page 169
- Information About Configuring Link Bundling, on page 174
Limitations and Compatible Characteristics of Ethernet Link Bundles

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

- The router supports mixed speed bundles. Mixed speed bundles allow member links of different bandwidth to be configured as active members in a single bundle. The ratio of the bandwidth for bundle members must not exceed 10. Also, the total weight of the bundle must not exceed 64. For example, 100Gbps link and 10Gbps links can be active members in a bundle and load-balancing on member links is based on bandwidth weightage.

- 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 LACP (Link Aggregation Control Protocol).

- A single router can support a maximum of 256 bundle interfaces. Link bundles of only physical interfaces are supported.

- If the HQoS profile is enabled, the maximum available trunks by default (physical+sub-interfaces) is 256. If more trunks are required, you can configure the `hw-module profile qos max-trunks <256/512/1024>` command. With HQoS enabled on bundle interfaces, a maximum of 4 priority levels are supported.

- The available trunks by default is 256. If more trunks are required, you can configure the `hw-module profile bundle-scale <256/512/1024>` command. With HQoS enabled on bundle sub-interfaces, a maximum of 4 priority levels are supported.

- The following limitations apply to the number of supported bundle members with HQoS profile:
  - Maximum of 1024 trunks (128 physical interfaces + 896 sub-interfaces) and 16 bundle members.
  - Maximum of 256 trunks (128 physical interfaces + 128 sub-interfaces) and 64 bundle members.
  - Maximum of 512 trunks (128 physical interfaces + 384 sub-interfaces) and 32 bundle members.

- The following limitations apply to bundle sub-interfaces and the number of members per bundle:
  - Maximum of 1024 bundle sub-interfaces, each containing up to 16 member-links.
  - Maximum of 256 bundle sub-interfaces, each containing up to 64 member-links.
  - Maximum of 512 bundle sub-interfaces, each containing up to 32 member-links.

- 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.

• Multicast traffic is load balanced over the members of a bundle. For a given flow, internal processes select the member link and all traffic for that flow is sent over that member.

**Configuring Ethernet Link Bundles**

This section describes how to configure an Ethernet link bundle.

---

**Note**

In order for an Ethernet bundle to be active, you must perform the same configuration on both connection endpoints of the bundle.

---

**SUMMARY STEPS**

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

**DETAILED STEPS**

**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*  
• 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**  
**exit**  
**Example:**

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

Exits interface configuration submode for the Ethernet link bundle.

**Step 8**  
**interface HundredGigE interface-path-id**  
**Example:**

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

Enters interface configuration mode for the specified interface.

Enter the **HundredGigE** keyword to specify the interface type. Replace the **interface-path-id** argument with the node-id in the **rack/slot/module** format.

**Step 9**  
**bundle id bundle-id [mode {active | on | passive}]**  
**Example:**

```
RP/0/RP0/CPU0:router(config-if)# bundle id 3 mode on
```

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 10**  
**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 11**  
**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 12  
**exit**

Example:

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

Exits interface configuration submode for the Ethernet interface.

Step 13  
**bundle id bundle-id [mode {active | passive | on}] no shutdown exit**

Example:

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/1
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 TenGigE 0/1/0/1
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 14  
**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 15
exit

Example:

RP/0/RP0/CPU0:router(config-if)# exit
Exits interface configuration mode.

Step 16
exit

Example:

RP/0/RP0/CPU0:router(config)# exit
Exits global configuration mode.

Step 17
Perform Step 1 through Step 15 on the remote end of the connection.
Brings up the other end of the link bundle.

Step 18
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 19
show lacp Bundle-Ether bundle-id

Example:

RP/0/RP0/CPU0:router# show lacp Bundle-Ether 3
(Optional) Shows detailed information about LACP ports and their peers.

---

Configuring LACP Fallback

This section describes how to configure the LACP Fallback feature.

**SUMMARY STEPS**

1. configure
2. interface Bundle-Ether bundle-id
3. ipv4 address ipv4-address mask
4. end or commit
5. show bundle infrastructure database ma bdl-info Bundle-e1010 | inc text
6. show bundle infrastructure database ma bdl-info Bundle-e1015 | inc text

**DETAILED STEPS**

Step 1 configure
Configuring EFP Load Balancing on an Ethernet Link Bundle

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

Example:

```plaintext
RP/0/RP0/CPU0:router# configure
Enters global configuration mode.
```

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

**Example:**

```plaintext
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3
Creates and names a new Ethernet link bundle.

The `interface Bundle-Ether` command enters 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 return to global configuration mode.
```

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

**Example:**

```plaintext
RP/0/RP0/CPU0:router(config-if)# bundle lacp-fallback timeout 4
Enables the LACP Fallback feature.
```

**Step 4** end or commit

**Example:**

```plaintext
RP/0/RP0/CPU0:router(config-subif)# commit
Saves configuration changes.
```

**Step 5** show bundle infrastructure database ma bdl-info Bundle-e1010 | inc text

**Example:**

```plaintext
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 6** show bundle infrastructure database ma bdl-info Bundle-e1015 | inc text

**Example:**

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

**Step 1**

configure

Example:

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

Enters global configuration mode.

**Step 2**

interface Bundle-Ether bundle-id l2transport

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3.1 l2transport
```

Creates a new Ethernet link bundle with the specified `bundle-id` and with Layer 2 transport enabled. The range is 1 to 65535.

**Step 3**

bundle load-balance hash hash-value [auto]

Example:

```
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1
```

or

```
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto
```

Configures all egressing traffic on the fixed members of a bundle to flow through the same physical member link.

- `hash-value`—Numeric value that specifies the physical member link through which all egressing traffic in this bundle will flow. The values are 1 through 8.
- `auto`—The physical member link through which all egressing traffic on this bundle will flow is automatically chosen.

**Step 4**

end or commit

Example:

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

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

Saves configuration changes.

- When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)?
- Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
- Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
- 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 all egress traffic on the fixed members of a bundle to flow through the same physical member link automatically.

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

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

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

This example shows output for show bundle load-balancing detail location command:

```
RP/0/RP0/CPU0:router# show bundle load-balancing detail location 0/0/cpu0
Fri Nov 1 17:54:20.417 UTC

Bundle-Ether1
Type: Ether (L3)
Members <current/max>: 1/64
Total Weighting: 1
Load balance: Default
Locality threshold: 65
Avoid rebalancing? False
Sub-interfaces: 2

Member Information:
```
VLANs on an Ethernet Link Bundle

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

- There is no separate limit defined for Layer 3 sub-interfaces on a bundle. However, an overall system limit of 4000 is applicable for NCS5001 and NCS5002, while a limit of 2000 is applicable for NCS5011.

Note

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

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

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

VLAN subinterfaces can support multiple Layer 2 frame types and services, such as Ethernet Flow Points (EFPs) and Layer 3 services.

Layer 2 EFPs are configured as follows:

```
interface bundle-ether instance.subinterface l2transport. encapsulation dot1q xxxxx
```

Layer 3 VLAN subinterfaces are configured as follows:

```
interface bundle-ether instance.subinterface, encapsulation dot1q xxxxx
```

---

**Note**

The difference between the Layer 2 and Layer 3 interfaces is the **l2transport** keyword. Both types of interfaces use **dot1q encapsulation**.

---

### Configuring VLAN over Bundles

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

**Step 1** Create an Ethernet bundle.

**Step 2** Create VLAN subinterfaces and assign them to the Ethernet bundle.

**Step 3** Assign Ethernet links to the Ethernet bundle.

These tasks are describe in detail in the procedure that follows.

---

**Note**

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

---

**SUMMARY STEPS**

1. configure
2. `interface Bundle-Ether bundle-id`
3. `ipv4 address ipv4-address mask`
4. `bundle minimum-active bandwidth kbps`
5. `bundle minimum-active links links`
6. `bundle maximum-active links links [hot-standby]`
7. `exit`
8. `interface Bundle-Ether bundle-id.vlan-id`
9. `encapsulation dot1qvlan-id`
10. `ipv4 address ipv4-address mask`
11. `no shutdown`
12. `exit`
13. Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.
14. `end` or `commit`
15. `exit`
16. `exit`
17. `configure`
18. `interface {TenGigE | FortyGigE | HundredGigE}interface-path-id`

**DETAILED STEPS**

**Step 1**
configure

Example:

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

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

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3
```
Creates and names a new Ethernet link bundle.

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

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

Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```
Assigns an IP address and subnet mask to the virtual interface using the `ipv4 address` configuration subcommand.

**Step 4**
`bundle minimum-active bandwidth kbps`

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000
```
(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.

**Step 5**  
**bundle minimum-active links** `links`  
**Example:**

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2
```

(Optional) Sets the number of active links required before you can bring up a specific bundle.

**Step 6**  
**bundle maximum-active links** `links` [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**  
**exit**  
**Example:**

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

Exits the interface configuration submode.

**Step 8**  
**interface Bundle-Ether** `bundle-id.vlan-id`  
**Example:**

```
RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3.1
```

Creates a new VLAN, and assigns the VLAN to the Ethernet bundle you created in Step 2.  
Replace the `bundle-id` argument with the `bundle-id` you created in Step 2.  
Replace the `vlan-id` with a subinterface identifier.  
Range is from 1 to 4093 inclusive (0, 4094, and 4095 are reserved).  
**Note**  
When you include the `.vlan-id` argument with the `interface Bundle-Ether bundle-id` command, you enter subinterface configuration mode.

**Step 9**  
**encapsulation dot1qvlan-id**  
**Example:**

```
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
```

Sets the Layer 2 encapsulation of an interface.

**Step 10**  
**ipv4 address ipv4-address mask**  
**Example:**

```
RP/0/RP0/CPU0:router#(config-subif)# ipv4 address 10.1.2.3/24
```
Assigns an IP address and subnet mask to the subinterface.

**Step 11**  
no shutdown  
**Example:**  
RP/0/RP0/CPU0:router(config-subif)# no shutdown  
(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.

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

**Step 13**  
Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.  
(Optional) Adds more subinterfaces to the bundle.

**Step 14**  
end or commit  
**Example:**  
RP/0/RP0/CPU0:router(config-subif)# end  
or  
RP/0/RP0/CPU0:router(config-subif)# commit  
Saves configuration changes.  
- When you issue the **end** command, the system prompts you to commit changes:

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

**Step 15**  
exit  
**Example:**  
RP/0/RP0/CPU0:router(config-subif)# end  
Exits interface configuration mode.
Step 16  exit

Example:

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

Exits global configuration mode.

Step 17  configure

Example:

RP/0/RP0/CPU0:router # configure

Enters global configuration mode.

Step 18  interface {'TenGigE | FortyGigE | HundredGigE'} interface-path-id

Example:

RP/0/RP0/CPU0:router(config)# interface TenGigE 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.

---

**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.

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.

---

## Configuring the Default LACP Short Period Time Interval

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

**SUMMARY STEPS**

1. configure
2. interface HundredGigEinterface-path
3. bundle id number mode active
4. lACP period short
5. end or commit

**DETAILED STEPS**

**Step 1**

configure

Example:

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

Enters global configuration mode.

**Step 2**

interface HundredGigEinterface-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
```

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

```
config
```
Configuring Custom LACP Short Period Time Intervals

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

**Note**

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

**SUMMARY STEPS**

1. configure
2. interface Bundle-Ether *bundle-id*
3. ipv4 address *ipv4-address* mask
4. bundle minimum-active bandwidth *kbps*
5. bundle minimum-active links *links*
6. bundle maximum-active links *links*
7. exit
8. interface Bundle-Ether *bundle-id*.vlan-*vlan-id*
9. dot1q vlan *vlan-id*
10. ipv4 address *ipv4-address* mask
11. no shutdown
12. exit
13. Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.
14. end or commit
15. exit
16. exit
17. show ethernet trunk bundle-ether *instance*
18. configure
19. `interface {HundredGigE} interface-path-id`
20. `bundle id bundle-id [mode {active | on | passive}]`
21. `no shutdown`
22. Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.
23. `end` or `commit`
24. Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.
25. `show bundle Bundle-Ether bundle-id [reasons]`
26. `show ethernet trunk bundle-ether instance`

**DETAILED STEPS**

**Step 1**
`configure`

Example:

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

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

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3
```
Creates and names a new Ethernet link bundle. This `interface Bundle-Ether` command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the `exit` command to exit from the interface configuration submode back to the normal global configuration mode.

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

Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```
Assigns an IP address and subnet mask to the virtual interface using the `ipv4 address` configuration subcommand.

**Step 4**
`bundle minimum-active bandwidth kbps`

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000
```
(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.

**Step 5**
`bundle minimum-active links links`

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2
```
(Optional) Sets the number of active links required before you can bring up a specific bundle.

**Step 6**
`bundle maximum-active links links`
Configuring Link Bundling

Example:

RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1

(Optional) Designates one active link and one link in standby mode that can take over immediately for a bundle if the active link fails (1:1 protection).

Note

- The default number of active links allowed in a single bundle is 8.
- If the `bundle maximum-active` command is issued, then only the highest-priority link within the bundle is active. The priority is based on the value from the `bundle port-priority` command, where a lower value is a higher priority. Therefore, we recommend that you configure a higher priority on the link that you want to be the active link.

Step 7

exit

Example:

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

Exits the interface configuration submode.

Step 8

interface Bundle-Ether bundle-id.vlan-id

Example:

RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3.1

Creates a new VLAN, and assigns the VLAN to the Ethernet bundle you created in Step 2.

Replace the `bundle-id` argument with the `bundle-id` you created in Step 2.

Replace the `vlan-id` with a subinterface identifier. Range is from 1 to 4093 inclusive (0, 4094, and 4095 are reserved).

Note

- When you include the `vlan-id` argument with the `interface Bundle-Ether bundle-id` command, you enter subinterface configuration mode.

Step 9

dot1q vlan vlan-id

Example:

RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 10

Assigns a VLAN to the subinterface.

Replace the `vlan-id` argument with a subinterface identifier. Range is from 1 to 4093 inclusive (0, 4094, and 4095 are reserved).

Step 10

ipv4 address ipv4-address mask

Example:

RP/0/RP0/CPU0:router(config-subif)# ipv4 address 10.1.2.3/24

Assigns an IP address and subnet mask to the subinterface.

Step 11

no shutdown

Example:
Step 12  exit
Example:

```
RP/0/RP0/CPU0:router(config-subif)# exit
```
Exits subinterface configuration mode for the VLAN subinterface.

Step 13  Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2. (Optional) Adds more subinterfaces to the bundle.

Step 14  end or commit
Example:

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

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

Step 15  exit
Example:

```
RP/0/RP0/CPU0:router(config-subif)# exit
```
Exits interface configuration mode.

Step 16  exit
Example:

```
RP/0/RP0/CPU0:router(config)# exit
```
Exits global configuration mode.
**Step 17**

show ethernet trunk bundle-ether instance

Example:

```
RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5
```

(Optional) Displays the interface configuration.

The Ethernet bundle instance range is from 1 through 65535.

**Step 18**

configure

Example:

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

Enters global configuration mode.

**Step 19**

interface {HundredGigE} interface-path-id

Example:

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

Enters the interface configuration mode for the Ethernet interface you want to add to the Bundle.

Enter the HundredGigE keyword to specify the interface type. Replace the interface-path-id argument with the node-id in the rack/slot/module format.

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

**Step 20**

bundle id bundle-id [mode {active | on | passive}] 

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle-id 3
```

Adds an Ethernet interface to the bundle you configured in Step 2 through Step 13.

To enable active or passive LACP on the bundle, include the optional mode active or mode passive keywords in the command string.

To add the interface to the bundle without LACP support, include the optional mode on keywords with the command string.

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

**Step 21**

no shutdown

Example:

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

(Optional) If a link is in the down state, bring it up. The no shutdown command returns the link to an up or down state depending on the configuration and state of the link.

**Step 22**

Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.
Step 23  
end or commit  

Example:  

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

Saves configuration changes.  

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

Step 24  
Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.  
Brings up the other end of the link bundle.  

Step 25  
show bundle Bundle-Ether bundle-id [reasons]  

Example:  

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

(Optional) Shows information about the specified Ethernet link bundle.  
The show bundle Bundle-Ether command displays information about the specified bundle. If your bundle has been configured properly and is carrying traffic, the State field in the show bundle Bundle-Ether command output will show the number “4,” which means the specified VLAN bundle port is “distributing.”  

Step 26  
show ethernet trunk bundle-ether instance  

Example:  

RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5  

(Optional) Displays the interface configuration.  
The Ethernet bundle instance range is from 1 through 65535.  

Information About Configuring Link Bundling  
To configure link bundling, you must understand the following concepts:
IEEE 802.3ad Standard

The IEEE 802.3ad standard typically defines a method of forming Ethernet link bundles. For each link configured as bundle member, the following information is exchanged between the systems that host each end of the link bundle:

- A globally unique local system identifier
- An identifier (operational key) for the bundle of which the link is a member
- An identifier (port ID) for the link
- The current aggregation status of the link

This information is used to form the link aggregation group identifier (LAG ID). Links that share a common LAG ID can be aggregated. Individual links have unique LAG IDs.

The system identifier distinguishes one router from another, and its uniqueness is guaranteed through the use of a MAC address from the system. The bundle and link identifiers have significance only to the router assigning them, which must guarantee that no two links have the same identifier, and that no two bundles have the same identifier.

The information from the peer system is combined with the information from the local system to determine the compatibility of the links configured to be members of a bundle.

The MAC address of the first link attached to a bundle becomes the MAC address of the bundle itself. The bundle uses this MAC address until that link (the first link attached to the bundle) is detached from the bundle, or until the user configures a different MAC address. The bundle MAC address is used by all member links when passing bundle traffic. Any unicast or multicast addresses set on the bundle are also set on all the member links.

**Note**

We recommend that you avoid modifying the MAC address, because changes in the MAC address can affect packet forwarding.

Link Bundle Configuration Overview

The following steps provide a general overview of the link bundle configuration. Keep in mind that a link must be cleared of all previous network layer configuration before it can be added to a bundle:

1. In global configuration mode, create a link bundle. To create an Ethernet link bundle, enter the `interface Bundle-Ether` command.
2. Assign an IP address and subnet mask to the virtual interface using the `ipv4 address` command.
3. Add interfaces to the bundle you created in Step 1 with the `bundle id` command in the interface configuration submode.
   You can add up to 32 links to a single bundle.
4. You can optionally implement 1:1 link protection for the bundle by setting the `bundle maximum-active links` command to 1. Performing this configuration causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. (The link priority is based on the value
of the **bundle port-priority** command.) If the active link fails, the standby link immediately becomes the active link.

---

**Note**

A link is configured as a member of a bundle from the interface configuration submode for that link.

---

**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 fast-switchover** 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.
CHAPTER 8

Configuring Traffic Mirroring

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

Feature History for Traffic Mirroring

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.2</td>
<td>SPAN over Pseudo-Wire was introduced.</td>
</tr>
<tr>
<td>Release 7.1.2</td>
<td>SPAN over File was introduced.</td>
</tr>
<tr>
<td>Release 7.1.2</td>
<td>File Mirroring was introduced.</td>
</tr>
</tbody>
</table>

- Introduction to Traffic Mirroring, on page 177
- Configure Traffic Mirroring, on page 181
- Traffic Mirroring Configuration Examples, on page 184
- Troubleshooting Traffic Mirroring, on page 189
- Introduction to ERSPAN Egress Rate Limit, on page 192
- SPAN over Pseudo-Wire, on page 195
- SPAN to File, on page 198
- Introduction to File Mirroring, on page 201

Introduction to Traffic Mirroring

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

Traffic mirroring copies traffic from one or more source ports and sends the copied traffic to one or more destinations for analysis by a network analyzer or other monitoring device. Traffic mirroring does not affect the flow of traffic on the source interfaces or sub-interfaces. It allows the mirrored traffic to be sent to a destination interface or sub-interface.

For example, you can attach a traffic analyzer to the router and capture Ethernet traffic that is sent by host A to host B.
Figure 11: Traffic Mirroring Operation

When local traffic mirroring is enabled, the traffic analyzer gets directly attached to the port that is configured to receive a copy of every packet that host A sends. This port is called a traffic mirroring port.

Traffic Mirroring Types

The following types of traffic mirroring are supported:

- **Local traffic mirroring**: This is the most basic form of traffic mirroring. The network analyzer or sniffer is attached directly to the destination interface. In other words, all monitored ports are located on the same router as the destination port.

- **Remote traffic mirroring**: The network analyzer is reached through a GRE tunnel over an IP network. A copy of every packet includes the Layer 2 header if the ethernet keyword is configured. As this renders the mirrored packets unroutable, the end point of the GRE tunnel must be the network analyzer.

  **Note**

  A copy of every packet includes the Layer 2 header if the ethernet keyword is configured. As this renders the mirrored packets unroutable, the end point of the GRE tunnel must be the network analyzer.

- **ACL-based traffic mirroring**: Traffic is mirrored based on the configuration of the interface ACL.

  You can mirror traffic based on the definition of an interface access control list. When you are mirroring Layer 3 traffic, the ACL is configured using the `ipv4 access-list` or the `ipv6 access-list` command with the **capture** option. The **permit** and **deny** commands determine the behavior of regular traffic. The **capture** option designates the packet is to be mirrored to the destination port, and it is supported only on permit type of access control entries (ACEs).

  **Note**

  Prior to Release 6.5.1, ACL-based traffic mirroring required the use of UDK (User-Defined TCAM Key) with the **enable-capture** option so that the **capture** option can be configured in the ACL.

Traffic Mirroring Terminology

- **Ingress Traffic** — Traffic that comes into the router.

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

**Characteristics of Source Port**

A source port, also called a monitored port, is a routed port that you monitor for network traffic analysis. In a single traffic mirroring session, you can monitor source port traffic. The NCS 5500 Series Router supports a maximum of up to 800 source ports.

A source port has these characteristics:

- It can be any data port type, such as Bundle Interface, 100 Gigabit Ethernet, or 10 Gigabit Ethernet.

  ![Note]

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

- Each source port can be monitored in only one traffic mirroring session.
- When a port is used as a source port, the same port cannot be used as a destination port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor local traffic mirroring. Remote traffic mirroring is supported both in the ingress and egress directions. For bundles, the monitored direction applies to all physical ports in the group.

**Characteristics of 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 are 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 router can have a maximum of four monitor sessions.
- A single monitor session can have only one destination port.
- A single destination port can belong to only one monitor session.
- A monitor session can have a maximum of 800 source ports, as long as the maximum number of source ports from all monitoring sessions does not exceed 800.
### Characteristics of Destination Port

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

A destination port has these characteristics:

- A destination port must reside on the same router as the source port for local traffic mirroring. For remote mirroring, the destination is always a GRE tunnel.

- A destination port for local mirroring can be any Ethernet physical port, EFP, or GRE tunnel interface, but not a bundle interface. It can be a Layer 2 or Layer 3 transport interface.

- A destination port on NCS5500 cannot be a VLAN subinterface.

- At any one time, a destination port can participate in only one traffic mirroring session. A destination port in one traffic mirroring session cannot be a destination port for a second traffic mirroring session. In other words, no two monitor sessions can have the same destination port.

- A destination port cannot also be a source port.

### Restrictions

The following are the generic restriction(s):

- Partial mirroring and sampled mirroring are not supported.

The following general restrictions apply to traffic mirroring using ACLs:

- Traffic mirroring counters are not supported.

- ACL-based traffic mirroring is not supported with Layer 2 (ethernet-services) ACLs.

- Configure ACL(s) on the source interface or any interface on the same network processing unit as the source interface, to avoid default mirroring of traffic. If a Bundle interface is a source interface, configure the ACL(s) on any interface on the same network processing unit as all active bundle-members. Bundle members can be on multiple NPUs. Also ensure that the ACL(s) configured are of the same protocol type and direction as the SPAN configuration. For example, if you configure SPAN with ACL for IPv4 or IPv6, configure an ingress IPv4 or IPv6 ACL on that network processing unit respectively.

The following general restrictions apply to SPAN:

- SPAN only supports port-level source interfaces.

The following restrictions apply to ERSPAN and SPAN ACL:

- SPAN counters are not supported.

- Both SPAN and ER-SPAN features cannot be configured on a router simultaneously. Either SPAN or ERSPAN feature can be configured on the same router.

- The value of ERSPAN session-ID is always zero.
  
  - IOS XR Command for configuring ERPAN is not available.

- ERSPAN next-hop must have ARP resolved.
  
  - Any other traffic or protocol will trigger ARP.
Configure Traffic Mirroring

These tasks describe how to configure traffic mirroring:

Configure Remote Traffic Mirroring

Step 1 configure
Example:

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

Step 2 monitor-session session-name
Example:

```
RP/0/RP0/CPU0:router(config)# monitor-session mon1 ethernet
RP/0/RP0/CPU0:router(config-mon)#
```
Defines a monitor session and enters monitor session configuration mode.

Step 3 destination interface tunnel-ip
Example:

```
RP/0/RP0/CPU0:router(config-mon)# destination interface tunnelip3
```
Specifies the destination subinterface to which traffic is replicated.

Step 4 exit
Example:

```
RP/0/RP0/CPU0:router(config-mon)# exit
RP/0/RP0/CPU0:router(config)#
```
Exits monitor session configuration mode and returns to global configuration mode.
**Step 5**  
**interface** type number  
**Example:**  
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/1/0/1  
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 6**  
**monitor-session** session-name ethernet direction rx-only port-only  
**Example:**  
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only port-only  
Specifies the monitor session to be used on this interface. Use the **direction** keyword to specify that only ingress or egress traffic is mirrored.

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

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:router# show monitor-session
Displays information about the traffic mirroring session.

## Attaching the Configurable Source Interface

**Step 1**  configure

*Example:*

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

Enters global configuration mode.

**Step 2**  interface  type  number

*Example:*

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

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:router(config-if)# ipv4 access-group acl1 ingress
```

Controls access to an interface.

**Step 4**  monitor-session  session-name  ethernet  direction  rx-only  port-level  acl

*Example:*

```
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only port-level acl
```

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:router(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 step overrides any ACL that may be configured on the interface.

**Step 6**  exit

*Example:*

```
Step 7  **end or commit**

**Example:**

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

or

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

Saves configuration changes.

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

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

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

Step 8  **show monitor-session [session-name] status [detail] [error]**

**Example:**

RP/0/RP0/CPU0:router# show monitor-session status

Displays information about the monitor session.

---

**Traffic Mirroring Configuration Examples**

This section contains examples of how to configure traffic mirroring:

**Configuring ACLs for Traffic Mirroring**

This section describes the configuration for creating ACLs for traffic mirroring.

**Configuration**

Use the following configuration to enable traffic mirroring with ACLs.
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 10.1.1.0 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
Router(config-ipv4-acl)# exit
Router(config)# commit

/* Validate the configuration */

Router(config)# show run
Thu May 17 11:17:49.968 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu May 17 11:17:47 2018 by user
...
    hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port proto frag-bit enable-capture
ipv4 access-list TM-ACL
  10 permit udp 10.1.1.0 0.0.0.255 eq 10 any capture
  20 permit udp 10.1.1.0 0.0.0.255 eq 20 any
!
...

You have successfully configured an IPv4 ACL for traffic mirroring.

**Configuring UDF-Based ACL for Traffic Mirroring**

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Configures individual UDF definitions. You can specify the name of the UDF, the networking header from which offset, and the length of data to be extracted.</td>
</tr>
<tr>
<td>udf udf-name header {inner</td>
<td>outer} {l2</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0# udf udf3 header outer l4 offset 0 length 1</td>
<td></td>
</tr>
<tr>
<td>(config-mon)#</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0# udf udf3 header inner l4 offset 10 length 2</td>
<td></td>
</tr>
<tr>
<td>(config-mon)#</td>
<td></td>
</tr>
</tbody>
</table>

The `inner` or `outer` keywords indicate the start of the offset from the unencapsulated Layer 3 or Layer 4 headers, or if there is an encapsulated packet, they indicate the start of offset from the inner L3/L4.

**Note** The maximum offset allowed, from the start of any header, is 63 bytes.

The `length` keyword specifies, in bytes, the length from the offset. The range is from 1 to 4.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 offset 50 length 1 (config-mon)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> hw-module profile tcam format access-list {ipv4</td>
<td>ipv6} [acl-qualifiers] [ udf1 udf-name1 ... udf8 udf-name8] enable-capture</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port dst-port proto tcp-flags packet-length frag-bit udf1 udf-test1 udf2 udf-test2 enable-capture</td>
<td>Note A reload of the line card is required for the new TCAM profile to take effect.</td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv4 access-list acl-name</td>
<td>Creates ACL and enters IP ACL configuration mode. The length of the acl-name argument can be up to 64 characters.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# ipv4 access-list acl1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> permit regular-ace-match-criteria udf udf-name1 value1 ... udf-name8 value8</td>
<td>Configures ACL with UDF match.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any udf udf1 0x1234 0xffff udf3 0x56 0xff capture RP/0/RP0/CPU0:router(config-ipv4-acl)# 30 permit ipv4 any any dscp af11 udf udf5 0x22 0x22 capture</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits IP ACL configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-ipv4-acl)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> interface type number</td>
<td>Configures interface and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> ipv4 access-group acl-name ingress</td>
<td>Applies access list to an interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-if)# ipv4 access-group acl1 ingress</td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>Command or Action</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>commit</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
</tbody>
</table>

### Verifying UDF-based ACL

Use the `show monitor-session status detail` command to verify the configuration of UDF on ACL.

```
RP/0/RP0/CPU0:leaf1# show monitor-session 1 status detail
Fri May 12 19:40:39.429 UTC
Monitor-session 1
  Destination interface tunnel-ip3
  Source Interfaces
    TenGigE0/0/0/15
    Direction: Rx-only
    Port level: True
    ACL match: Enabled
    Portion: Full packet
    Interval: Mirror all packets
    Status: Not operational (destination not active)
```

### Traffic Mirroring with Physical Interfaces (Local): Example

This example shows the basic configuration for traffic mirroring with physical interfaces.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# monitor-session ms1
RP/0/RP0/CPU0:router(config-mon)# destination interface HundredGigE0/2/0/15
RP/0/RP0/CPU0:router(config-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE0/2/0/19
RP/0/RP0/CPU0:router(config-if)# monitor-session ms1 port-level direction rx-only
RP/0/RP0/CPU0:router(config-if)# commit
```

### Viewing Monitor Session Status: Example

This example shows sample output of the `show monitor-session` command with the `status` keyword:

```
RP/0/RSP0/CPU0:router# show monitor-session status
Monitor-session cisco-rtp1
  Destination interface HundredGigE 0/5/0/38
  Source Interface  Dir  Status
  TenGigE0/5/0/4    Both  Operational
  TenGigE0/5/0/17   Both  Operational

RP/0/RSP0/CPU0:router# show monitor-session status detail
```

---

Configuring Traffic Mirroring

Verifying UDF-based ACL
Monitoring Traffic Mirroring on a Layer 2 Interface

This section describes the configuration for monitoring traffic on a Layer 2 interface.

Configuration

To monitor traffic mirroring on a Layer 2 interface, configure the monitor under l2transport sub-config of the interface:
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:

```
# show monitor-session status
Monitor-session ms1
```

In the preceding example, the line marked as `<Session status>` can indicate one of these configuration errors:

<table>
<thead>
<tr>
<th>Session Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session is not configured globally</td>
<td>The session does not exist in global configuration. Review the <code>show run</code> command output and ensure that a session with a correct name has been configured.</td>
</tr>
<tr>
<td>Destination interface <code>&lt;intf&gt;</code> <code>&lt;(down-state)&gt;</code></td>
<td>The destination interface is not in Up state in the Interface Manager. You can verify the state using the <code>show interfaces</code> command. Check the configuration to determine what might be keeping the interface from coming up (for example, a sub-interface needs to have an appropriate encapsulation configured).</td>
</tr>
</tbody>
</table>

The `<Source interface status>` can report these messages:
### Source Interface Status

<table>
<thead>
<tr>
<th>Source Interface Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational</td>
<td>Everything appears to be working correctly in traffic mirroring PI. Please follow up with the platform teams in the first instance, if mirroring is not operating as expected.</td>
</tr>
<tr>
<td>Not operational (Session is not configured globally)</td>
<td>The session does not exist in global configuration. Check the <code>show run</code> command output to ensure that a session with the right name has been configured.</td>
</tr>
<tr>
<td>Not operational (destination not known)</td>
<td>The session exists, but it either does not have a destination interface specified, or the destination interface named for the session does not exist. For example, if the destination is a sub-interface that has not been created.</td>
</tr>
<tr>
<td>Not operational (source same as destination)</td>
<td>The session exists, but the destination and source are the same interface, so traffic mirroring does not work.</td>
</tr>
<tr>
<td>Not operational (destination not active)</td>
<td>The destination interface or pseudowire is not in the Up state. See the corresponding Session status error messages for suggested resolution.</td>
</tr>
<tr>
<td>Not operational (source state &lt;down-state&gt;)</td>
<td>The source interface is not in the Up state. You can verify the state using the <code>show interfaces</code> command. Check the configuration to see what might be keeping the interface from coming up (for example, a sub-interface needs to have an appropriate encapsulation configured).</td>
</tr>
<tr>
<td>Error: see detailed output for explanation</td>
<td>Traffic mirroring has encountered an error. Run the <code>show monitor-session status detail</code> command to display more information.</td>
</tr>
</tbody>
</table>

The `show monitor-session status detail` command displays full details of the configuration parameters and any errors encountered. For example:

```
RP/0/RP0/CPU0:router# show monitor-session status detail
Monitor-session sess1
      Destination interface is not configured
      Source Interfaces
      -----------------
      TenGigE0/0/0/1
      Direction: Both
      ACL match: Disabled
      Portion: Full packet
      Status: Not operational (destination interface not known)
      TenGigE0/0/0/2
      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 TenGigE 0/0/0/0
Source Interfaces
-----------------
TenGigE 0/1/0/0.100:
  Direction: Both
  Status: Operating
TenGigE 0/2/0/0.200:
  Direction: Tx
  Status: Error: <blah>

Monitor session bar
No destination configured
Source Interfaces
-----------------
TenGigE 0/3/0/0.100:
  Direction: Rx
  Status: Not operational (no destination)

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 process all
RP/0/RP0/CPU0:router# debug monitor-session process ea
RP/0/RP0/CPU0:router# debug monitor-session process ma
RP/0/RP0/CPU0:router# show monitor-session process mgr

  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
Introduction to ERSPAN Egress Rate Limit

With ERSPAN egress rate limit feature, you can monitor traffic flow through any IP network. This includes third-party switches and routers.

ERSAPN 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.

This feature provides rate limiting of the mirroring traffic or the egress traffic. With rate limiting, you can limit the amount of egress traffic to a specific rate, which prevents the network and remote ERSPAN destination traffic overloading. Be informed, if the egress rate-limit exceeds then the system may cap or drop the monitored traffic.

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.

- The Discard Class (0 through 2):
  - The default is 0.
  - The discard class configuration is used in WRED.

Benefits

With ERSPAN Egress rate limit feature, you can limit the egress traffic or the mirrored and use the mirrored traffic for data analysis.

Topology

Figure 12: Topology for ERSPAN Egress 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 egress rate limit feature is applied on the router egress 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 Egress Rate Limit

Use the following steps to configure ERSPAN egress 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

```
!! Policy-map to be used with the ERSPAN Destination (egress interface)
!! Traffic class is set to 5. For packets in this class, apply shaping
!! as well as WRED.
class-map match-any TC5
  match traffic-class 5
discard-class
!
policy-map shape-foo
class TC5
  random-detect discard-class 0 10000 bytes 40000 bytes
  random-detect discard-class 1 40000 bytes 80000 bytes
  random-detect discard-class 2 80000 bytes 200000 bytes
  shape average percent 15
!
class class-default
!
end-policy-map
!
!!GRE Tunnel Interface
interface Loopback49
  ipv4 address 49.49.49.49 255.255.255.255
!
interface tunnel-ip100
  ipv4 address 130.100.1.1 255.255.255.0
tunnel mode gre ipv4
tunnel source 49.49.49.49
tunnel destination 10.8.1.2
!
!!ERSPAN Monitor Session with GRE tunnel as the Destination Interface, and with QoS configuration
monitor-session FOO ethernet
destination interface tunnel-ip100
traffic-class 5
discard-class 1
!
```
!!ERSPAN Source Interface
interface TenGigE0/6/0/4/0
description connected to TGEN 9/5
ipv4 address 10.4.90.1 255.255.255.0
monitor-session FOO ethernet port-level
!
!!ERSPAN Destination ip-tunnel00’s underlying interface, with egress policy-map shape-foo attached
interface TenGigE0/6/0/9/0
  service-policy output shape-foo
ipv4 address 10.8.1.1 255.255.255.0

Verification
RP/0/RP0/CP00: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
    Portion: Full packet
    Interval: Mirror all packets
    Status: Operational
RP/0/RP0/CP00:ios#show monitor-session <sess-id> status internal
RP/0/RP0/CP00:ios#show monitor-session FOO status internal
Wed May 2 15:13:06.063 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session FOO (ID 0x00000001) (Ethernet)
  SPAN Mgr: Destination interface tunnel-ip100 (0x08000001) (Ethernet)
    Last error: Success
    Tunnel data:
      Mode: GREoIPv4
      Source IP: 49.49.49.49
      Dest IP: 10.8.1.2
      VRF:
      ToS: 0 (copied)
      TTL: 255
      DFbit: Not set
0/6/CPU0: Destination interface tunnel-ip100 (0x08000001c)
    Tunnel data:
      Mode: GREoIPv4
      Source IP: 49.49.49.49
      Dest IP: 10.8.1.2
      VRF:
      ToS: 0 (copied)
      TTL: 255
      DFbit: Not set

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/6/CPU0: Name 'FOO', destination interface tunnel-ip100 (0x08000001c)
Platform, 0/6/CPU0:
  Dest Port: 0xe7d
ERSPAN Encap:
  Tunnel ID: 0x4001380b
ERSPAN Tunnel ID: 0x4001380c
IP-NH Grp key: 0x3140000cc5
IP-NH hdl: 0x308a5fa5e0
IP-NH IFH: 0x30002a0
IP-NH IPAddr: 10.4.91.2

NPU MirrorRx MirrorTx
00 0x00000003 0x00000004
01 0x00000003 0x00000004
02 0x00000003 0x00000004
03 0x00000003 0x00000004
04 0x00000003 0x00000004
05 0x00000003 0x00000004
RP/0/RP0/CPU0:ios#

SPAN over Pseudo-Wire

Pseudo-wire traffic mirroring (known as PW-SPAN) is an extra functionality on the existing SPAN solutions. The existing SPAN solutions are monitored on a destination interface or through a GRE tunnel or RSPAN. In PW-SPAN, the traffic mirroring destination port is configured to be a pseudo-wire rather than a physical port. Here, the designated traffic on the source port is mirrored over the pseudo-wire to a central location. This allows the centralization of expensive network traffic analysis tools.

Because the pseudo-wire carries only mirrored traffic, this traffic is unidirectional. Incoming traffic from the remote provider edge is not allowed. Typically, a monitor session should be created with a destination pseudo-wire. This monitor session is one of the L2VPN xconnect segments. The other segment of the L2VPN VPWS is a pseudowire.

Note

- A single router can have a maximum of four monitor sessions.
- Only port-level source interfaces are supported.

Limitations

The following functionalities are not supported for SPAN over PW:

- Monitor session statistics
- RSPAN
- Partial packet SPAN
- Sampled SPAN
- ERSPAN Tunnel statistics
- A destination port cannot be a source port.

Configuring SPAN over Pseudo-Wire

Use the following steps to configure SPAN over Pseudo-Wire:
Configure SPAN monitor session

RP/0/RP0/CPU0:router#config
RP/0/RP0/CPU0:router(config)#monitor-session M1
RP/0/RP0/CPU0:router(config-mon)#destination pseudowire
RP/0/RP0/CPU0:router(config-mon)#commit

Configure SPAN source

RP/0/RP0/CPU0:router#config
Fri Sep 6 03:49:59.312 UTC
RP/0/RP0/CPU0:router(config)#interface Bundle-Ether100
RP/0/RP0/CPU0:router(config-if)#monitor-session M1 ethernet port-level
RP/0/RP0/CPU0:router(config-if-mon)#commit

Configure l2vpn xconnect

RP/0/RP0/CPU0:router(config)#l2vpn
RP/0/RP0/CPU0:router(config-l2vpn)#pw-class span
RP/0/RP0/CPU0:router(config-l2vpn-pwc)#encapsulation mpls
RP/0/RP0/CPU0:router(config-l2vpn-pwc-mpls)#transport-mode ethernet
RP/0/RP0/CPU0:router(config-l2vpn)#xconnect group 1
RP/0/RP0/CPU0:router(config-l2vpn-xc)#p2p 2
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)#monitor-session M1
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)#neighbor ipv4 1.1.1.1 pw-id 2
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)#commit

Verifying SPAN over Pseudo-Wire

The following examples show how to verify SPAN over Pseudo-Wire.

To check monitor session status:

RP/0/RP0/CPU0:router#show run monitor-session M1
monitor-session M1 ethernet destination pseudowire

RP/0/RP0/CPU0:router#show monitor-session M1 status
Monitor-session M1
Destination pseudowire
Source Interface    Dir    Status
BE100 (port)        Both   Operational
BE400 (port)        Both   Operational

RP/0/RP0/CPU0:router#show monitor-session M1 status detail
Monitor-session M1
Destination pseudowire
Source Interfaces
---------------------
Bundle-Ether100
Direction: Both
Port level: True
ACL match: Disabled
Portion: Full packet
Interval: Mirror all packets
Status: Operational

Bundle-Ether400
Direction: Both
Port level: True
ACL match: Disabled
Portion: Full packet
Interval: Mirror all packets
Status: Operational
To check underlying l2vpn xconnect:

```bash
RP/0/RP0/CPU0:router# show run l2vpn
l2vpn
  pw-class span
  encapsulation mpls
  transport-mode ethernet

! 

p2p 2
  monitor-session M1
  neighbor ipv4 1.1.1.1 pw-id 2

!

p2p 10
  monitor-session M2
  neighbor ipv4 1.1.1.1 pw-id 10
  pw-class span

!

RP/0/RP0/CPU0:router# show l2vpn xconnect
Fri Sep 6 03:41:15.691 UTC
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

<table>
<thead>
<tr>
<th>Group</th>
<th>Name</th>
<th>ST</th>
<th>Segment 1 Description</th>
<th>ST</th>
<th>Segment 2 Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>UP</td>
<td>M1</td>
<td>UP</td>
<td>1.1.1.1</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>UP</td>
<td>M2</td>
<td>UP</td>
<td>1.1.1.1</td>
</tr>
</tbody>
</table>

To verify traffic:

```bash
RP/0/RP0/CPU0:router# show controllers npu voq-usage interface tenGigE 0/0/0/12 instance all location 0/0/CPU0
Tue Jul 16 14:06:45.040 UTC

<table>
<thead>
<tr>
<th>Intf</th>
<th>Intf handle</th>
<th>NPU #</th>
<th>NPU Port</th>
<th>Sys Port</th>
<th>VOQ Flow</th>
<th>VOQ Port</th>
<th>Port speed</th>
<th>Port type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Te0/0/0/12</td>
<td>1d0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>1200</td>
<td>11448</td>
<td>local</td>
<td>10G</td>
</tr>
</tbody>
</table>

RP/0/RP0/CPU0:router# show controllers fia diagshell 0 "diag last core=0" location 0/0/CPU0
Tue Jul 16 14:11:19.124 UTC

Node ID: 0/0/CPU0
Core 0:

Last packet information: is_valid=1  tm_port=30
pp_port=30  src_syst_port=49153  port_header_type=eth  packet_size=0
Packet start, offset in bytes:
00bc6016 64db7ae7 4f59b241 884705dc 91ff7ae7 4f59b048 001a9400 0002a6dd
60000000 00463bff 10000000 00000000 00000000 00000002 20000000 00000000
00000000 00000002 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 000008a47 c35c7674
```
SPAN to File

Starting with Cisco IOS XR Software Release 7.1.2, the SPAN to File is an extension of the pre-existing SPAN feature to allow network packets to be mirrored to a file instead of an interface, so that they can be analyzed later. The file format is PCAP, so that it can be easily used with tools such as tcpdump or wireshark.

Note

SPAN to File supports 100 local SPAN sessions, each supporting 1 destination file and 100 source interfaces. A maximum of 1000 source ports are supported across the system. Individual platforms may support lower numbers. The SPAN session may be any of these currently supported classes: Ethernet, IPv4, IPv6, MPLS-IPv4, and MPLS-IPv6.

When a file is configured as a destination for a SPAN session, a buffer is created 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 may be providing packets for the same buffer. The buffers are deleted when the session configuration is removed. The file is written by each node to a location on the active RP which contains the node ID of the node on which the buffer was located.

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. For example, this would result in a file on each node on which the bundle has members which contains packets from all members on that node.

Note

SPAN to File may not capture all the frames when SPAN is applied on high traffic rate.

Action Commands for SPAN to File

Action commands are added to start and stop network packet collection. The commands may only be run on sessions where the destination is a file. The action command auto-completes names of globally configured SPAN to File sessions. See the table below for more information on action commands.

Table 4: Action Commands for SPAN to File

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>monitor-session &lt;name&gt; packet-collection start</td>
<td>Issue this command to start writing packets for the specified session to the configured buffer.</td>
</tr>
</tbody>
</table>
Stop

**Action** | **Command** | **Description**
--- | --- | ---
Stop | `monitor-session <name> packet-collection stop [ discard-data | write directory <dir> filename <filename> ]` | Issue this command to stop writing packets to the configured buffer. If the discard-data option is specified, the buffer is simply cleared, whereas if the write option is specified, the buffer is written to disk before clearing. If the buffer is to be written, it is done so in .pcap format to this location: `/<directory>/<node_id>/<filename>.pcap`. If the user adds a .pcap extension when specifying the filename, this is removed so that the extension is not 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”.

The `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.
- Mirror interval from 512 to 16k.
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/RP0/CPU0:router#show run interface Bundle-Ether 1
Fri Apr 24 12:12:59.348 EDT
interface Bundle-Ether1
monitor-session ms7 ethernet
```

**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
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

Introduction to File Mirroring

Prior to Cisco IOS XR Software Release 7.1.2, 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 7.1.2, 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       |76fc1b906be4fe08ed06e0c93667815 |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/RSP0/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
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 (RSP). The configuration and control plane are mirrored onto the standby RSP and, in the event of a failover, the virtual interfaces move to the ex-standby, which then becomes the newly active RSP.

• Information About Configuring Virtual Interfaces, on page 203

Information About Configuring Virtual Interfaces

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

Virtual Loopback Interface Overview

A virtual loopback interface is a virtual interface with a single endpoint that is always up. Any packet transmitted over a virtual loopback interface is immediately received by the 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.
Prerequisites for Configuring Virtual Interfaces

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

Configuring Virtual Loopback Interfaces

This task explains how to configure a basic loopback interface.

Restrictions

The IP address of a loopback interface must be unique across all routers on the network. It must not be used by another interface on the router, and it must not be used by an interface on any other router on the network.

SUMMARY STEPS

1. configure
2. interface loopback instance
3. ipv4 address ip-address
4. end or commit
5. show interfaces type instance

DETAILED STEPS

Step 1 configure
Example:

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

Step 2 interface loopback instance
Example:

```
RP/0/RP0/CPU0:router#(config)# interface Loopback 3
```
Enters interface configuration mode and names the new loopback interface.

Step 3 ipv4 address ip-address
Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 100.100.100.69 255.255.255.255
```
Assigns an IP address and subnet mask to the virtual loopback interface using the `ipv4 address` configuration command.

Step 4 end or commit
Example:

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

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

Saves configuration changes.

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

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

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

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

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

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

Step 5  show interfaces type instance

Example:

RP/0/RP0/CPU0:router# show interfaces Loopback0

(Optional) Displays the configuration of the loopback interface.

---

Example

This example shows how to configure a loopback interface:

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Loopback0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 100.100.100.69 255.255.255.255
RP/0/RP0/CPU0:router(config-if)# ipv6 address 100::69/128
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Loopback0

Loopback0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Loopback interface(s)
  Internet address is 100.100.100.69/32
  MTU 1500 bytes, BW 0 Kbit
  reliability Unknown, txload Unknown, rxload Unknown
  Encapsulation Loopback, loopback not set,
  Last link flapped 01:57:47
  Last input Unknown, output Unknown
  Last clearing of "show interface" counters Unknown
  Input/output data rate is disabled.
Null Interface Overview

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

The only interface configuration command that you can specify for the null interface is the `ipv4 unreachables` command. With the `ipv4 unreachables` command, if the software receives a non-broadcast 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 `ipv4 unreachables` command is enabled. If we do not want ICMP to send protocol unreachable, then we need to configure using the `ipv4 icmp unreachable disable` command.

The Null 0 interface is created by default during boot process and cannot be removed. The `ipv4 unreachables` command can be configured for this interface, but most configuration is unnecessary because this interface just discards all the packets sent to it.

The Null 0 interface can be displayed with the `show interfaces null0` command.

Configuring Null Interfaces

This task explains how to configure a basic null interface.

**SUMMARY STEPS**

1. `configure`
2. `interface null 0`
3. `end` or `commit`
4. `show interfaces null 0`

**DETAILED STEPS**

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

**Step 2**  
*interface null 0*  
**Example:**  
```
RP/0/RP0/CPU0:router(config)# interface null 0
```
Enters the null 0 interface configuration mode.

**Step 3**  
*end* or *commit*  
**Example:**


RP/0/RP0/CPU0:router(config-null0)# end

or

RP/0/RP0/CPU0:router(config-null0)# commit

Saves configuration changes.

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

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

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

Step 4  show interfaces null 0

Example:

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

Verifies the configuration of the null interface.

Example

This example shows 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 icmp unreachables disable
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
Interface state transitions: 1
Hardware is Null interface
Internet address is Unknown
MTU 1500 bytes, BW 0 Kbit
reliability 255/255, txload Unknown, rxload Unknown
Encapsulation Null, loopback not set,
Last link flapped 4d20h
Last input never, output never
Last clearing of "show interface" counters 05:42:04
5 minute input rate 0 bits/sec, 0 packets/sec
Configuring Virtual IPv4 Interfaces

This task explains how to configure an IPv4 virtual interface.

**SUMMARY STEPS**

1. configure
2. ipv4 virtual address ipv4-
3. end or commit

**DETAILED STEPS**

**Step 1** configure

Example:

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

Enters global configuration mode.

**Step 2** ipv4 virtual address ipv4-

Example:

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

Defines an IPv4 virtual address for the management Ethernet interface.

**Step 3** end or commit

Example:

```
RP/0/RP0/CPU0:router(config-null0)# end
```

or

```
RP/0/RP0/CPU0:router(config-null0)# commit
```

Saves configuration changes.

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

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

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

---

**Example**

This is an example for configuring a virtual IPv4 interface:

```
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
```
Configuring GRE Tunnels

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

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

Restrictions for Configuring GRE Tunnels

The following restrictions apply while configuring GRE tunnels:

- The router supports up to 500 GRE tunnels.
- Only up to 16 unique source IP addresses are supported for the tunnel source.
- 2-pass to Single-pass migration, which means converting the same GRE tunnel, is not possible in a single configuration step. You must first delete the 2-pass tunnel and then add the Single-pass tunnel.
• Configurable MTU is not supported on Single-pass GRE interface, but supported on 2-pass GRE interface.

Configuration Example

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

```
RP/0/RP0/CPU0:Router1# configure
RP/0/RP0/CPU0:Router1(config)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-if)# tunnel mode gre ipv4
RP/0/RP0/CPU0:Router1(config-if)# ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/CPU0:Router1(config-if)# tunnel source 192.168.1.1
RP/0/RP0/CPU0:Router1(config-if)# tunnel destination 192.168.2.1
RP/0/RP0/CPU0:Router1(config-if)# exit
RP/0/RP0/CPU0:Router1(config)# interface Loopback 0
RP/0/RP0/CPU0:Router1(config-if)# ipv4 address 1.1.1.1
RP/0/RP0/CPU0:Router1(config-if)# exit
```

```
RP/0/RP0/CPU0:Router1(config)# router ospf 1
RP/0/RP0/CPU0:Router1(config-ospf)# router-id 192.168.4.1
RP/0/RP0/CPU0:Router1(config-ospf)# area 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-ospf-ar)# interface Loopback 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# commit
```

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

Single Pass GRE Encapsulation Allowing Line Rate Encapsulation

Single Pass GRE Encapsulation Allowing Line Rate Encapsulation feature, also known as Prefix-based GRE Tunnel Destination for Load Balancing feature, enables line rate GRE encapsulation traffic and enables flow entropy. Data-plane forwarding performance supports full line rate, which is adjusted to consider added encapsulation. GRE tunnel goes down if the destination is not available in RIB. Routing over GRE Single-pass tunnel is not supported in Release 6.3.2, so the traffic that is eligible for GRE encapsulation is identified using an ACL filter that is based on GRE encapsulation. GRE tunnel destination address is an anycast address. All of the GRE encapsulation must be assigned based upon either an ACL or a policy-map, or both. Destinations may be individual addresses or /28 prefixes.
Configuration

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

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

/* GRE Single-Pass */

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

/* GRE Entropy(ECMP/UCMP) */

ECMP (ISIS)

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

/* UCMP(ISIS) */

Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# ucmp
Router(config-isis-af)# metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# redistribute connected
Router(config-isis-af)# commit
Router(config-isis-af)# exit
!
Router# configure
Router(config)# interface Bundle-Ether3
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 20
Router(config-af)# commit
Router(config-af)# exit
!

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

/* ECMP (OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface HundredGigE0/7/0/23
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit
!

/* UCMP (OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# ucmp
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface HundredGigE0/7/0/23
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit

/* ECMP (BGP) */
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# network 170.170.170.3/32
Router(config-bgp-af)# network 170.170.170.10/32
Router(config-bgp-af)# network 170.170.170.11/32
Router(config-bgp-af)# network 170.170.172.3/32
Router(config-bgp-af)# network 180.180.180.9/32
Router(config-bgp-af)# network 180.180.180.20/32
Router(config-bgp-af)# network 180.180.180.21/32
Router(config-bgp-af)# network 180.180.180.24/32
Router(config-bgp-af)# network 180.180.180.25/32
Router(config-bgp-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 4.1.1.2
Router(config-bgp-nbr)# remote-as 300
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# commit
!
/* UCMP (BGP) */

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

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

Router# configure
Router(config)# route-policy TRANSIT1_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:3750000)
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
!

Running Configuration

/* GRE Single-Pass configuration */

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

/* GRE Entropy(ECMP/UCMP) */

ECMP (ISIS)

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

/* UCMP(ISIS) */

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

! interface Bundle-Ether3
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 20

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

! /* ECMP(OSPF) */

router ospf 3
nsr
maximum paths 5
address-family ipv4 unicast
area 0
interface Bundle-Ether3

! interface Bundle-Ether4
! interface Bundle-Ether111
! interface Bundle-Ether112
! interface Loopback23
! interface HundredGigE0/7/0/23
!
!
/* UCMP (OSPF) */

router ospf 3
nsr
maximum paths 5
ucmp
address-family ipv4 unicast
area 0
interface Bundle-Ether3
cost 2
! interface Bundle-Ether4
! interface Bundle-Ether11
! interface Bundle-Ether12
cost 2
! interface Loopback23
! interface HundredGigE0/7/0/23
!

/* ECMP(BGP)*/

router bgp 800
bgp bestpath as-path multipath-relax
address-family ipv4 unicast
maximum-paths ebgp 5
network 170.170.170.3/32
network 170.170.170.10/32
network 170.170.170.11/32
network 170.170.172.3/32
network 180.180.180.9/32
network 180.180.180.20/32
network 180.180.180.21/32
network 180.180.180.24/32
network 180.180.180.25/32
!
neighbor 4.1.1.2
remote-as 300
address-family ipv4 unicast
route-policy PASS-ALL in
route-policy PASS-ALL out
next-hop-self
!

/* UCMP(BGP) */

router bgp 800
bgp bestpath as-path multipath-relax
address-family ipv4 unicast
maximum-paths ebgp 5
network 180.180.180.9/32
network 180.180.180.20/32
network 180.180.180.21/32
network 180.180.180.24/32
network 180.180.180.25/32
!
neighbor 7.1.5.2
remote-as 4000
address-family ipv4 unicast
route-policy TRANSIT0_IN in
route-policy PASS-ALL out
next-hop-self
!
neighbor 4.1.111.2
remote-as 4000
address-family ipv4 unicast
route-policy TRANSIT1_IN in
route-policy PASS-ALL out
next-hop-self
!

/* Configure route policy */

route-policy TRANSIT0_IN
if destination in (170.170.170.24/32) then
  set extcommunity bandwidth (2906:1250000)
else
  pass
endif
end-policy
!
route-policy TRANSIT1_IN
if destination in (170.170.170.24/32) then
  set extcommunity bandwidth (2906:3750000)
else
  pass
endif
end-policy
!

Verification

Verify if the tunnel mode GRE encapsulation is enabled.

Router# show interfaces tunnel-ip 100

Sun Jul 10 15:49:04.812 VN_TIME
tunnel-ip100 is up, line protocol is up
  Interface state transitions: 2
  Hardware is Tunnel
  Internet address is Unknown
  MTU 1500 bytes, BW 100 Kbit (Max: 100 Kbit)
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation TUNNEL_GRE, loopback not set,
  Tunnel TOS 0
  Tunnel mode GRE IPV4,
  Keepalive is enabled, interval 10 seconds, maximum retry 3
  Tunnel source 172.16.16.1 (GigabitEthernet0_0_0_0), destination 172.16.16.2
  Tunnel TTL 100
  Last input 2d03h, output 2d04h
  Last clearing of "show interface" counters never
5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec
689 packets input, 26212 bytes, 0 total input drops
0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 0 multicast packets
3 packets output, 192 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets

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

Router# sh interfaces tunnel-ip 5 accounting
Wed May 16 01:50:57.258 UTC
tunnel-ip5

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Pkts In</th>
<th>Chars In</th>
<th>Pkts Out</th>
<th>Chars Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPV4_UNICAST</td>
<td>489</td>
<td>55746</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IPV6_UNICAST</td>
<td>489</td>
<td>55746</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MPLS</td>
<td>587</td>
<td>69266</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

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

Number of tunnel updates to retry: 0
Number of tunnel updates retried: 0
Number of tunnel retries failed: 0
Platform:
Recycle VoQ: 48

<table>
<thead>
<tr>
<th>ReceivedBytes</th>
<th>DroppedBytes</th>
<th>ReceivedPackets</th>
<th>DroppedPackets</th>
<th>ReceivedKbps</th>
<th>DroppedKbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPU 0:0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPU 1:0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPU 2:0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Verify if the tunnel mode GRE encapsulation is enabled.
Router# `show interfaces tunnel-ip * brief`

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

Verify the tunnel endpoint route in RIB.

Router# `show route 10.1.1.1`

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

Verify if the tunnel mode GRE encapsulation is enabled.

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

----- node0_0_CPU0 -----
tunnel ifhandle 0x80022cc
tunnel source 161.115.1.2
tunnel destination 162.1.1.1/32
tunnel transport vrf table id 0xe0000000
tunnel mode gre ipv4, encap
tunnel bandwidth 100 kbps
tunnel platform id 0x0

tunnel flags 0x40003400
IntfStateUp
BcStateUp
Ipv4Caps
Encap
tunnel mtu 1500
tunnel tos 0
tunnel ttl 255
tunnel adjacency flags 0x1
tunnel o/p interface handle 0x0
tunnel key 0x0, entropy length 0 (mask 0xffffffff)
tunnel QT next 0x0
tunnel platform data {nil}

Platform:
Handle: {nil}
Decap ID: 0
Decap RIF: 0
Decap Recycle Encap ID: 0x00000000
Encap RIF: 0
Encap Recycle Encap ID: 0x00000000
Encap IPv4 Encap ID: 0x4001381b
Encap IPv6 Encap ID: 0x00000000
Encap MPLS Encap ID: 0x00000000
DecFEC DecRcyLIF DecStatsId EncRcyLIF

Verify if the QoS table is updated properly.

Router# `show controllers npu stats voq base 48 instance all location 0/0/CPU0`

Asic Instance = 0
VOQ Base = 48

<table>
<thead>
<tr>
<th>ReceivedPkt</th>
<th>ReceivedBytes</th>
<th>DroppedPkt</th>
<th>DroppedBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ReceivedPkts</td>
<td>ReceivedBytes</td>
<td>DroppedPkts</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Asic Instance - 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>VOQ Base - 48</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COS0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COS1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COS2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COS3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ReceivedPkts</th>
<th>ReceivedBytes</th>
<th>DroppedPkts</th>
<th>DroppedBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Asic Instance - 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>VOQ Base - 48</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COS0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COS1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COS2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COS3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Configuring Generic UDP Encapsulation

Read this section to get an overview of Generic UDP Encapsulation technique, and know how to configure Generic UDP Encapsulation.

- Understand Generic UDP Encapsulation, on page 223
- Restrictions, on page 225
- Configure GUE, on page 225

Understand Generic UDP Encapsulation

UDP encapsulation is a technique of adding network headers to the 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 only handle the outer Source IP address and parts of the destination address and may not provide sufficient load balance 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, that 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:
For example, if the data stream is sent from Host 1 to Host 2. The server acts as a GUE encapsulator that is sending 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 2 bits of the version number in the IP header.

Benefits of using GUE

- Allows direct encapsulation of payloads like 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. Destination also could be terminated based on a subnet providing additional bits for entropy.

Avoids special handling for transport 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 variant 1 only.
- Receives IPv4 packets with the defined GUE port of 6080.
- Decapsulates IPv6 packets with the defined GUE port of 6615.
- Receives MPLS packets with the UDPMPLS port of 6635
- Range of source or destination ports is not supported.
- Range, Source, or Destination addresses are not supported, but subnet mask entries are allowed.
- Destination Port is mandatory to perform decapsulation.
- 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.

**Configure GUE**

Use the following configuration workflow to configure GUE:

1. 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.

2. Configure a policy map: Define a policy map and associate the traffic class with the traffic policy.

3. Apply the policy per VRF basis, and apply this policy on all the interfaces that are part of the VRF.

**Configuration Example**

1. Configure a traffic class:

   ```
   Router#configure
   Router(config)#class-map type traffic match-all gre-1
   Router(config-cmap)#match destination-address ipv4 225.100.20.0 255.255.255.0
   Router(config-cmap)#match protocol gre
   Router(config-cmap)#end-class-map
   Router(config-cmap)#commit
   ```
Router(config)#class-map type traffic match-all udp-v4
Router(config-cmap)#match destination-address ipv4 220.100.20.0 255.255.255.0
Router(config-cmap)#match source-address ipv4 210.100.20.0 255.255.255.0
Router(config-cmap)#match protocol udp
Router(config-cmap)#match destination-port 6080
Router(config-cmap)#end-class-map
Router(config-cmap)#commit

Router(config)#class-map type traffic match-all udp-mpls1
Router(config-cmap)#match destination-address ipv4 220.100.20.0 255.255.255.0
Router(config-cmap)#match destination-port 6635
Router(config-cmap)#end-class-map
Router(config-cmap)#commit

Router(config)#class-map type traffic match-all udp-v6
Router(config-cmap)#match destination-address ipv4 220.100.20.0 255.255.255.0
Router(config-cmap)#match source-address ipv4 210.100.20.0 255.255.255.0
Router(config-cmap)#match protocol udp
Router(config-cmap)#match destination-port 6615
Router(config-cmap)#end-class-map
Router(config-cmap)#commit

2. 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 gre-1
Router(config-pmap-c)#decapsulate gre
Router(config-pmap-c)#exit

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

3. Apply the policy per VRF basis:

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

Configure Generic UDP Decapsulation for Load Balancing

On transit routers, the outer IP for hashing is used to encode the entropy parameters. But at the terminating or decapsulating router, the payload is used for hashing. However, you can use the outer IP at the decapsulating
router as well, as payloads may have limited entropy. To enable the outer IP based hashing on the decapsulation router, use this command:

```
Router(config)# hw-module profile load-balance algorithm ip-tunnel
Router(config)# commit
```

Note

The `hw-module profile load-balance algorithm ip-tunnel` command does not recover a reload of the system, unlike other `hw-module` commands.
Configuring Generic UDP Encapsulation

Configure GUE
Configuring Controllers

This chapter describes the Optics Controller and Coherent DSP Controller for the 6-port Coherent Line Card (NC55-6X200-DWDM-S). This chapter also describes the procedures used to configure the controllers.

When two MACsec enabled Cisco NCS 5500 routers with Coherent Line Cards are connected, there is no compatibility between Coherent Line Cards of IOS XR Release version 6.5.x (or lower) and 6.6.1 (or higher).

- Optics Controllers, on page 229
- Maintenance Mode, on page 230
- Performance Monitoring, on page 230
- How to Configure Controllers, on page 231

Optics Controllers

Controllers are represented in the rack/slot/instance/port format (r/s/i/p); for example, 0/3/0/1. Each port has an optics controller that is created on startup.

You must shut down the optics controller before you perform any of the following tasks:
- Configure the controller
- Restore a saved configuration
- Upgrade the DSP processor or CFP2 optics module Field Programmable Device (FPD)

CFP2 DCO Optics Support

There are two hardware versions of the CFP DCO optics (A0 and B0). You can identify the version A0 and B0 using show coherent driver internal location 0/0/CPU0 command and looking at “VID”.

A0 = V01
B0 = V02

The CFP2 DCO version A0 optics support the following traffic types:
### Maintenance Mode

Coherent DSP controllers can be placed in maintenance mode. Use the `controller coherentDSP secondary-admin-state maintenance` command to place controllers in maintenance mode.

Use the `show controllers optics r/i/p` command to view optics parameter values, laser state, controller state, admin state, and trunk alarms on the card, and threshold values for the different optics parameters.

Use the `show controllers coherentDSP r/i/p` command to view the DSP controller state and alarm status and statistics.

---

**Note**

In maintenance mode, all alarms are suppressed and the `show alarms` command does not display alarm details. However, traffic is not affected in maintenance mode.

### 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.

PM for optical parameters include input signal power and transmit power, optical signal-to-noise ratio, chromatic dispersion, polarization dependent loss, second order polarization mode dispersion, differential group delay, and transmitter laser bias current.
PM for DSP parameters include:

- FEC: error corrected bits, uncorrectable blocks, pre-FEC BER (block errors ratio)
- OTN: errored seconds, severely effected seconds, unavailable seconds, failed counts

These parameters simplify troubleshooting operations and enhance data that can be collected directly from the equipment.

**How to Configure Controllers**

This section contains the following procedures:

**Configuring Optics Controller**

You can configure parameters such as performance monitoring, high power threshold, and wavelength for Optics controller.

To configure the Optics controller, use the following commands:

**Before you begin**

You must shut down the optics controller before you perform any of the following tasks:

- Configure the controller
- Restore a saved configuration
- Upgrade the DSP processor or CFP2 optics module Field Programmable Device (FPD)

**SUMMARY STEPS**

1. `configure`
2. `controller optics r/s/i/p`
3. `shutdown`
4. `commit`
5. `rx-high-threshold rx-high`
6. `tx-high-threshold tx-high`
7. `no shutdown`
8. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>controller optics (r/s/i/p)</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# controller optics 0/3/0/1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>shutdown</td>
<td>Shuts down the optics controller.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-Optics)# shutdown</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-Optics)# commit</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rx-high-threshold (rx-high)</td>
<td>Configures the high receive power threshold. The range is -400 to 300 (in the units of 0.1 dBm).</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-Optics)# rx-high-threshold 200</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tx-high-threshold (tx-high)</td>
<td>Configures the high transmit power threshold. The range is -400 to 300 dBm (in the units of 0.1 dBm).</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-Optics)# tx-high-threshold 300</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no shutdown</td>
<td>Removes the shutdown configuration on the optics controller.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-Optics)# no shutdown</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-Optics)# commit</td>
<td></td>
</tr>
</tbody>
</table>
When you bring up the local optics controller, you might briefly see transient loss of signal (LOS) alarms on the console. This behavior might be observed during the initial tuning of the channel.

PKT_INFRA-FM-2-FAULT_CRITICAL : ALARM_CRITICAL :LOS-P :DECLARE :CoherentDSP0/3/0/1:  
PKT_INFRA-FM-2-FAULT_CRITICAL : ALARM_CRITICAL :LOS-P :CLEAR :CoherentDSP0/3/0/1:  

During the laser-on process, you might briefly see transient loss of line (LOL) alarms on the console. This alarm is cleared when the laser-on process is complete.

PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR :CTP2 RX LOL :DECLARE ::  
PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR :CTP2 RX LOL :CLEAR ::  

The laser-on process can take up to 120 seconds to complete.

Configuring Port Mode Speed

Each port on the 6-port Coherent Line Card can support 100 Gbps (DWDM QPSK), 150Gbps (DWDM 8 QAM), or 200Gbps (DWDM 16 QAM) WDM signals.

The line card has three Digital Signal Processors (DSPs), one for each pair of ports:
- Ports 0 and 1 – DSP0
- Ports 2 and 3 – DSP1
- Ports 4 and 5 – DSP2

When you configure the port-mode speed for 150Gbps (8 QAM), the port pairs belonging to a DSP are coupled. You need to configure the port-mode speed on each port of the port pair that belongs to the same DSP.

To configure the port mode speed, use the following commands:

**Before you begin**

You must shut down the controller before you configure the controller or restore a saved configuration.

**SUMMARY STEPS**

1. configure
2. controller optics r/s/i/p
3. shutdown
4. commit
5. port-mode speed { 100G | 150G | 200G } mod { 16qam | 8qam | qpsk } fec { 15sdfec | 15sdfecde | 25sdfec | otu7staircase } diff { enable | disable }  
6. no shutdown
7. commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** configure  
Example:  
RP/0/RP0/CPU0:router# configure | Enters global configuration mode. |
| **Step 2** controller optics r/s/i/p  
Example:  
RP/0/RP0/CPU0:router(config)# controller optics 0/3/0/0 | Enters optics controller configuration mode |
| **Step 3** shutdown  
Example:  
RP/0/RP0/CPU0:router(config-Optics)# shutdown | Shuts down the optics controller. |
| **Step 4** commit  
Example:  
RP/0/RP0/CPU0:router(config-Optics)# commit | Saves the configuration changes to the running configuration file and remains within the configuration session. |
| **Step 5** port-mode speed { 100G | 150G | 200G } mod { 16qam | 8qam | qpsk } fec { 15sdfec | 15sdfecde | 25sdfec | otu7staircase } diff { enable | disable }  
Example:  
RP/0/RP0/CPU0:router(config-Optics)# port-mode speed 100G mod qpsk fec 15sdfec diff | Configures the port mode speed. |
| **Step 6** no shutdown  
Example:  
RP/0/RP0/CPU0:router(config-Optics)# no shutdown | Removes the shutdown configuration on the optics controller. |
| **Step 7** commit  
Example:  
RP/0/RP0/CPU0:router(config-Optics)# commit | Saves the configuration changes to the running configuration file. |
When you bring up the local optics controller, you might briefly see transient loss of signal (LOS) alarms on the console. This behavior might be observed during the initial tuning of the channel.

PKT_INFRA-FM-2-FAULT_CRITICAL : ALARM_CRITICAL : LOS-P : DECLARE : CoherentDSP0/3/0/1:
PKT_INFRA-FM-2-FAULT_CRITICAL : ALARM_CRITICAL : LOS-P : CLEAR : CoherentDSP0/3/0/1:

During the laser-on process, you might briefly see transient loss of line (LOL) alarms on the console. This alarm is cleared when the laser-on process is complete.

PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR : CTP2 RX LOL : DECLARE :
PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR : CTP2 RX LOL : CLEAR :

Note

On NCS-55A2-MOD-S and NC55-MOD-A-S with CFP2-DCO optics:

- During the laser-on process, you might briefly see Optical Transport Network (OTN) alarms on the console. This alarm is cleared when the laser-on process is complete.

PKT_INFRA-FM-6-FAULT_INFO : OTUK-BDI : DECLARE : CoherentDSP0/0/2/2:
PKT_INFRA-FM-6-FAULT_INFO : OTUK-BDI : CLEAR : CoherentDSP0/0/2/2:

- During the laser-on process, you might briefly see transient transmit power and receive power alarms on the console. These alarms are cleared when the laser-on process is complete.

PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR : LO-RXPOWER : DECLARE : Optics0/0/2/0:
PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR : LO-TXPOWER : DECLARE : Optics0/0/2/0:
PKT_INFIRA-FM-4-FAULT_MINOR : ALARM_MINOR : HI-RXPOWER : DECLARE : Optics0/0/2/0:
PKT_INFIRA-FM-4-FAULT_MINOR : ALARM_MINOR : HI-TXPOWER : DECLARE : Optics0/0/2/0:

- When you bring up the local optics controller, you might see repeated remote faults on the console.

PLATFORM-DPA-2-RX_FAULT : Interface HundredGigE0/0/2/2/0, Detected Remote Fault
PLATFORM-DPA-2-RX_FAULT : Interface HundredGigE0/0/2/2/1, Detected Remote Fault
PLATFORM-DPA-2-RX_FAULT : Interface HundredGigE0/0/2/2/0, Detected Local Fault
PLATFORM-DPA-2-RX_FAULT : Interface HundredGigE0/0/2/2/1, Detected Local Fault
PLATFORM-DPA-2-RX_FAULT : Interface HundredGigE0/0/2/2/0, Detected Remote Fault
PLATFORM-DPA-2-RX_FAULT : Interface HundredGigE0/0/2/2/1, Detected Remote Fault

If you need to change the port-mode speed, you must remove the existing port mode speed configuration by entering the no port-mode command. You can then change the port mode speed.

The following example shows how to change the port mode speed to 100Gbps.

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# controller optics 0/3/0/0
RP/0/RP0/CPU0:router(config-Optics)# shutdown
RP/0/RP0/CPU0:router(config-Optics)# commit
RP/0/RP0/CPU0:router(config-Optics)# no port-mode
RP/0/RP0/CPU0:router(config-Optics)# commit
RP/0/RP0/CPU0:router(config-Optics)# port-mode speed 100G mod qpsk fec 15sdfec diff enable
RP/0/RP0/CPU0:router(config-Optics)# commit
RP/0/RP0/CPU0:router(config-Optics)# no shutdown
RP/0/RP0/CPU0:router(config-Optics)# commit
RP/0/RP0/CPU0:router(config-Optics)# exit
What to do next

After you configure the port-mode speed, you can configure the following interfaces:

- **100G** – Each optics controller configuration creates a single 100GE port:
  - `interface HundredGigE r/s/i/p/0` (where `p` = CTP2 port 0-5)
    - 0/3/0/0/0
    - 0/3/0/1/0
    - 0/3/0/2/0
    - 0/3/0/3/0
    - 0/3/0/4/0
    - 0/3/0/5/0

- **200G** – Each optics controller configuration creates two 100GE ports:
  - `interface HundredGigE r/s/i/p/0, r/s/i/p/1` (where `p` = CTP2 port 0-5)
    - 0/3/0/0/0, 0/3/0/0/1
    - 0/3/0/1/0, 0/3/0/1/1
    - 0/3/0/2/0, 0/3/0/2/1
    - 0/3/0/3/0, 0/3/0/3/1
    - 0/3/0/4/0, 0/3/0/4/1
    - 0/3/0/5/0, 0/3/0/5/1

- **150G (coupled)** – Coupled optics controller configuration creates three 100GE ports:
  - `interface HundredGigE r/s/i/p/0, r/s/i/p/1, r/s/i/p+1/0` (where `p` = CTP2 port: 0, 2, 4 [port `p` and `p` +1 are coupled])
    - 0/3/0/0/0, 0/3/0/0/1, 0/3/0/1/0
    - 0/3/0/2/0, 0/3/0/2/1, 0/3/0/3/0
    - 0/3/0/4/0, 0/3/0/4/1, 0/3/0/5/0

For more information, see the Configuring Ethernet Interfaces chapter.

To configure speed on 25G quad ports, use the following command:

```
RP/0/RP0/CPU0:router(config)# hw-module quad X location 0/0/CPU0 mode 10g
```

`X` is the number of quads (0,1,2,3...n) supported. Each quad has a default speed of 25G. This can be configured in 10G or reverted to 25G using `no` form of the command for a 10G port:

```
RP/0/RP0/CPU0:router(config)# no hw-module quad X location 0/0/CPU0 mode 10g
```
A quad number always starts from 0 to the maximum supported number. The number of quads supported varies from platform to platform and the CLI validates it. For example, the NCS 540 Series Router supports two quads (0 and 1). If the user enters \( X = 3 \), the CLI returns an error.

### Configuring Wavelength

To configure wavelength, use the following commands:

**Before you begin**

- Before configuring the wavelength, use the `show controllers optics r/s/i/p dwdm-carrier-map` command to display the wavelength and channel mapping for optics controllers.
- You must shut down the controller before you configure the controller or restore a saved configuration.

#### SUMMARY STEPS

1. configure
2. controller optics \( r/s/i/p \)
3. shutdown
4. commit
5. `dwdm-carrier \{100MHz-grid frequency frequency \} \{50GHz-grid [ frequency frequency | channel-number ] \}`
6. no shutdown
7. commit

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure&lt;br&gt;Example: RP/0/RP0/CPU0:router# configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>controller optics ( r/s/i/p )&lt;br&gt;Example: RP/0/RP0/CPU0:router(config)# controller optics 0/3/0/1</td>
<td>Enters optics controller configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>shutdown&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-Optics)# shutdown</td>
<td>Shuts down the optics controller.</td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action** | **Purpose**
--- | ---
**Step 4** | **commit**
*Example:* RP/O/RP0/CPU0:router(config-Optics) # commit
Saves the configuration changes to the running configuration file and remains within the configuration session.

**Step 5** | **dwdm-carrier \{100MHz-grid frequency frequency \} \| \{50GHz-grid [ frequency frequency \| channel-number ] \}**
*Example:* RP/O/RP0/CPU0:router(config-Optics) # dwdm-carrier 100MHz-grid frequency 1960875
Configures the frequency on the trunk port.

**Step 6** | **no shutdown**
*Example:* RP/O/RP0/CPU0:router(config-Optics) # no shutdown
Removes the shutdown configuration on the optics controller.

**Step 7** | **commit**
*Example:* RP/O/RP0/CPU0:router(config-Optics) # commit
Saves the configuration changes to the running configuration file and remains within the configuration session.

---

To configure a DWDM carrier with the required frequency:

```
RP/O/RP0/CPU0:router#config
RP/O/RP0/CPU0:router(config)#controller Optics0/3/0/0
RP/O/RP0/CPU0:router(config-Optics)#dwdm-carrier
RP/O/RP0/CPU0:router(config-Optics)#dwdm-carrier 100MHz-grid
RP/O/RP0/CPU0:router(config-Optics)#dwdm-carrier 100MHz-grid frequency
RP/O/RP0/CPU0:router(config-Optics)#dwdm-carrier 100MHz-grid frequency 1960625
```

The output of `show run controller optics 0/3/0/0` command is:

```
RP/O/RP0/CPU0:router#show run controller optics 0/3/0/0
Wed Nov 6 13:47:33.178 UTC
controller Optics0/3/0/0
transmit-power -7
port-mode speed 100G mod qpsk fec 25sdfec diff disable
dwdm-carrier 100MHz-grid frequency 1960625
```
When you bring up the local optics controller, you might briefly see transient loss of signal (LOS) alarms on the console. This behavior might be observed during the initial tuning of the channel.

```
PKT_INFRA-FM-2-FAULT_CRITICAL : ALARM_CRITICAL :LOS-P :DECLARE :CoherentDSP0/3/0/1:
PKT_INFRA-FM-2-FAULT_CRITICAL : ALARM_CRITICAL :LOS-P :CLEAR :CoherentDSP0/3/0/1:
```

During the laser-on process, you might briefly see transient loss of line (LOL) alarms on the console. This alarm is cleared when the laser-on process is complete.

```
PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR :CTP2 RX LOL :DECLARE ::
PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR :CTP2 RX LOL :CLEAR ::
```

## Configuring Coherent DSP Controller

You can configure the administrative state for the Coherent DSP controller. To configure the Coherent DSP controller, use the following commands.

### Note

The coherent DSP controller doesn’t support Q factor, Q margin, and post FEC BER reporting. Therefore, no threshold crossing alert (TCA) is raised for these parameters.

## SUMMARY STEPS

1. `configure`
2. `controller coherentDSP r/s/i/p`
3. `secondary-admin-state admin-state`
4. `commit`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller coherentDSP r/s/i/p</td>
<td>Enters Coherent DSP optics controller configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# controller coherentDSP 0/3/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> secondary-admin-state admin-state</td>
<td>Configures the administrative state of the controller indicating that the controller is under maintenance.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

```
RP/0/RP0/CPU0:router(config-CoDSP)#
secondary-admin-state maintenance
```

**Step 4**

**commit**

**Example:**

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

Saves the configuration changes to the running configuration file and remains within the configuration session.

### Configuring Performance Monitoring

You can configure the performance monitoring parameters for the optics and Coherent DSP controllers. To configure PM parameters, use the following commands.

#### SUMMARY STEPS

1. `configure`
2. `controller  { optics | coherentDSP } r/s/i/p`
3. `pm  { 30-sec | 15-min | 24-hour } { optics | fec | otn } [ report | threshold value]`
4. `commit`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | `configure`  
**Example:**
```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode. |
| **Step 2** | `controller  { optics | coherentDSP } r/s/i/p`  
**Example:**
```
RP/0/RP0/CPU0:router(config)# controller coherentDSP 0/3/0/1
```

Enters optics or Coherent DSP controller configuration mode. |
| **Step 3** | `pm  { 30-sec | 15-min | 24-hour } { optics | fec | otn } [ report | threshold value]`  
**Example:**
```
RP/0/RP0/CPU0:router(config-CoDSP)# pm 15-min otn
threshold es-ne
```

Configures the performance monitoring parameters. |
| **Step 4** | `commit`  
**Example:**
```
RP/0/RP0/CPU0:router(config-CoDSP)# commit
```

Saves the configuration changes to the running configuration file and remains within the configuration session. |
Global Navigation Satellite System

This chapter describes the Global Navigation Satellite System (GNSS) NCS-55A2-MOD-SE-S Line Card. This chapter also describes the procedures used to configure the GNSS port.

- Configuring the Global Navigation Satellite System, on page 241
- Information About GNSS, on page 241
- Configure GNSS, on page 243

Configuring the Global Navigation Satellite System

In typical telecom networks, synchronization works in a hierarchical manner where the core network is connected to a stratum-1 clock. The stratum-1 clock is then distributed along the network in a tree-like structure. However, with a GNSS receiver, clocking is changed to a flat architecture, where access networks can directly take clock from satellites in sky by using an on-board GPS chip.

IOS XR NCS-55A2-MOD-SE-S Router now uses a satellite receiver, also called the Global Navigation Satellite System (GNSS), as the new timing interface.

This capability simplifies network synchronization planning, provides flexibility and resilience in resolving network synchronization issues in the hierarchical network.

Information About GNSS

Overview of GNSS

The following routers support the GNSS receiver:

- NCS-55A2-MOD-SL
- NCS-55A2-MOD-HD-S
- NCS-55A2-MOD-HX-S
- NCS-55A2-MOD-SE-S

No license is required to enable the GNSS module. The GNSS LED on the front panel indicates the status of the module. The following table describes the different LED status:
When connected to an external antenna, the module can acquire satellite signals and track up to 32 GNSS satellites, and compute location, speed, heading, and time. GNSS provides an accurate one pulse-per-second (PPS), a stable 10 MHz frequency output to synchronize broadband wireless, aggregation and pre-aggregation routers, and an accurate time-of-day (ToD).

Operation of GNSS Module

The GNSS module has the following stages of acquiring and providing timing signals to the Cisco router:

- **Self-surveymode** - When the router is reset, the GNSS module comes up in self-surveymode. It tries to lock on to a minimum of four different satellites and computes approximately 2000 different positions of the satellites to obtain a 3-D location (Latitude, Longitude, and Height) of its current position. This operation takes about 35 to 40 minutes. During this stage also, the module is able to generate accurate timing signals and achieve a Normal or Phase-locked state.

When GNSS moves into Normal state, you can start using the 1PPS, 10 MHz, and ToD inputs from GNSS. The quality of the signal in self-surveymode with Normal state is considered good enough to lock to GNSS.

- **Over determined clock mode** - The router switches to over determined (OD) mode when the self-surveymode is complete and the position information is stored in non-volatile memory on the router. In this mode, the module only processes the timing information based on satellite positions captured in self-surveymode.

The router saves the tracking data, which is retained even when the router is reloaded.

The GNSS module stays in the OD mode unless one of the following conditions occur:

- A position relocation of the antenna of more than 100 meters is detected. This detection causes an automatic restart of the self-surveymode.

- A manual restart of the self-surveymode or when the stored reference position is deleted.

- A worst-case recovery option after a jamming-detection condition that cannot be resolved with other methods.

<table>
<thead>
<tr>
<th>LED Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>GNSS NormalState. Selfsurvey is complete.</td>
</tr>
<tr>
<td>Amber</td>
<td>All other states</td>
</tr>
</tbody>
</table>
You can configure the GNSS module to automatically track any satellite or configure it to explicitly use a specific constellation. However, the module uses configured satellites only in the OD mode.

**Note**

GLONASS and BeiDou satellites cannot be enabled simultaneously.

When the router is reloaded, it always comes up in the OD mode unless:

- the router is reloaded when the self-survey mode is in progress.
- the physical location of the router is changed to more than 100 m from its pre-reloaded condition.

When the system restarts GNSS self-survey by using the default gnss slot R0/R1 command in config mode, the 10MHz, 1PPS, and ToD signals are not changed and remain up.

**Prerequisites for GNSS**

To use GNSS, the antenna must see as much as possible from the sky. For proper timing, a minimum of four satellites must be locked. For more information, see the *Cisco NCS 5500 Series Router Hardware Installation Guide*.

**Restrictions for GNSS**

- The GNSS module is not supported through SNMP; all configurations are performed through commands.
- The GNSS holdover performance is one microsecond in two hours of holdover after twelve hours of GNSS lock time.
- TDEV fails marginally on NCS-55A2-MOD-SE-S with GNSS input.

**Configure GNSS**

**Configuration Example**

This section describes how you can configure GNSS for a router.

```plaintext
/* Enable the GNSS receiver and enter the gnss-receiver submode */

RP/0/RP0/CPU0(config)# gnss-receiver 0 location 0/0/CPU00/RP0/CPU0
RP/0/RP0/CPU0(config-gnss)# frequency synchronization
RP/0/RP0/CPU0(config-gnss-freqsync)# selection input
```

**Optional Configuration Example**

```plaintext
RP/0/RP0/CPU0(config)# gnss-receiver 0 location 0/0/CPU00/RP0/CPU0
RP/0/RP0/CPU0(config-gnss)# anti-jam disable
RP/0/RP0/CPU0(config-gnss)# constellation GPS
RP/0/RP0/CPU0(config-gnss)# snr threshold 10
RP/0/RP0/CPU0(config-gnss)# frequency synchronization
RP/0/RP0/CPU0(config-gnss-freqsync)# selection input
```
Configure GNSS

RP/0/RP0/CPU0(config-gnss-freqsync)# priority 5
RP/0/RP0/CPU0(config-gnss-freqsync)# wait-to-restore 180

Running Configuration

gnss-receiver 0 location 0/RP0/CPU0
frequency synchronization
selection input
wait-to-restore 0
!

Verification

# show gnss-receiver
GNSS-receiver 0 location 0/RP0/CPU0
 Status: Available, Up
 Position: 741:12.12 N 4451:39.60 E 0.827km
 Time: 2019:01:17 14:43:08 (UTC offset: 18s)
 Firmware Version: 1.4
 Lock Status: Phase Locked, Receiver Mode: 3D-fix
 Survey Progress: 100, Holdover Duration: 0
 Major Alarm: Not used
 Minor Alarm: Not used
 Anti-jam: Enabled, Cable-delay compensation: 0
 1PPS polarity: Positive
 PDOP: 6.000, HDOP: 0.000, VDOP: 0.000, TDOP: 1.000
 Constellation: GPS, Satellite Count: 10