CONTENTS

PREFACE

Preface xi
Changes to This Document xi
Obtaining Documentation and Submitting a Service Request xi

CHAPTER 1

New and Changed Feature Information 1
Interface and Hardware Component Features Added or Modified in IOS XR Release 6.3.x 1

CHAPTER 2

Preconfiguring Physical Interfaces 3
Physical Interface Preconfiguration Overview 4
Prerequisites for Preconfiguring Physical Interfaces 4
Benefits of Interface Preconfiguration 4
How to Preconfigure Physical Interfaces 5
Information About Preconfiguring Physical Interfaces 6
Use of the Interface Preconfigure Command 7

CHAPTER 3

Advanced Configuration and Modification of the Management Ethernet Interface 9
Prerequisites for Configuring Management Ethernet Interfaces 9
How to Perform Advanced Management Ethernet Interface Configuration 10
Configuring a Management Ethernet Interface 10
IPv6 Stateless Address Auto Configuration on Management Interface 13
Modifying the MAC Address for a Management Ethernet Interface 14
Verifying Management Ethernet Interface Configuration 16
Information About Configuring Management Ethernet Interfaces 16

CHAPTER 4

Configuring Ethernet Interfaces 17
Configuring Gigabit Ethernet Interfaces 17
CFM Protocol Messages  54
Continuity Check (IEEE 802.1ag and ITU-T Y.1731)  54
Loopback (IEEE 802.1ag and ITU-T Y.1731)  57
Linktrace (IEEE 802.1ag and ITU-T Y.1731)  58
Configurable Logging  60
Flexible VLAN Tagging for CFM  60
How to Configure Ethernet OAM  61
Configuring Ethernet Link OAM  61
  Configuring an Ethernet OAM Profile  62
  Attaching an Ethernet OAM Profile to an Interface  68
  Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration  69
Verifying the Ethernet OAM Configuration  70
Configuring Ethernet CFM  71
  Configuring a CFM Maintenance Domain  71
  Configuring Services for a CFM Maintenance Domain  72
  Enabling and Configuring Continuity Check for a CFM Service  74
  Configuring Automatic MIP Creation for a CFM Service  76
  Configuring Cross-Check on a MEP for a CFM Service  77
  Configuring Other Options for a CFM Service  79
Configuring CFM MEPs  81
Configuring Y.1731 AIS  83
Configuring AIS in a CFM Domain Service  83
Configuring AIS on a CFM Interface  85
Configuring Flexible VLAN Tagging for CFM  86
Verifying the CFM Configuration  87
Troubleshooting Tips  88
CFM Over Bundles  89
Y.1731 Performance Monitoring  89
  Two-Way Delay Measurement  90
    Configuring Two-Way Delay Measurement  90
  Synthetic Loss Measurement  96
    Configuring Synthetic Loss Measurement  97
Bit Error Rate  102
Configuration Examples for Ethernet OAM  103
Configuration Examples for EOAM Interfaces 103
   Configuring an Ethernet OAM Profile Globally: Example 103
   Configuring Ethernet OAM Features on an Individual Interface: Example 103
   Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example 104
   Clearing Ethernet OAM Statistics on an Interface: Example 104
   Enabling SNMP Server Traps on a Router: Example 104
Configuration Examples for Ethernet CFM 105
   Ethernet CFM Domain Configuration: Example 105
   Ethernet CFM Service Configuration: Example 105
   Flexible Tagging for an Ethernet CFM Service Configuration: Example 105
   Continuity Check for an Ethernet CFM Service Configuration: Example 105
   MIP Creation for an Ethernet CFM Service Configuration: Example 105
   Cross-check for an Ethernet CFM Service Configuration: Example 106
   Other Ethernet CFM Service Parameter Configuration: Example 106
   MEP Configuration: Example 106
   Ethernet CFM Show Command: Examples 106
   AIS for CFM Configuration: Examples 109
   AIS for CFM Show Commands: Examples 110
show ethernet cfm interfaces ais Command: Example 110
show ethernet cfm local meps Command: Examples 110
show ethernet cfm local meps detail Command: Example 112

CHAPTER 6
Configuring Integrated Routing and Bridging 113
   IRB Introduction 113
   Bridge-Group Virtual Interface 114
   Supported Features on a BVI 114
   BVI Interface and Line Protocol States 115
   Prerequisites for Configuring IRB 115
   Restrictions for Configuring IRB 116
   How to Configure IRB 117
      Configuring the Bridge Group Virtual Interface 117
      Configuration Guidelines 117
      Configuring the Layer 2 AC Interfaces 119
Contents

CHAPTER 7

Configuring Link Bundling 127
  Limitations and Compatible Characteristics of Ethernet Link Bundles 128
  Configuring Ethernet Link Bundles 129
  Configuring LACP Fallback 133
  Configuring EFP Load Balancing on an Ethernet Link Bundle 134
  VLANs on an Ethernet Link Bundle 137
  Configuring VLAN over Bundles 138
    LACP Short Period Time Intervals 142
    Configuring the Default LACP Short Period Time Interval 143
    Configuring Custom LACP Short Period Time Intervals 145
    Configuring VPWS Cross-Connects in MC-LAG 150
    Configuring VPLS in MC-LAG 153
  Information About Configuring Link Bundling 155
    IEEE 802.3ad Standard 155
    Link Bundle Configuration Overview 156
    Link Switchover 156
    LACP Fallback 157
    Failure Cases 157

CHAPTER 8

Configuring Traffic Mirroring 159
  Introduction to Traffic Mirroring 159

Configuring a Bridge Group and Assigning Interfaces to a Bridge Domain 120
Associating the BVI as the Routed Interface on a Bridge Domain 121
Displaying Information About a BVI 123
Additional Information on IRB 123
Packet Flows Using IRB 123
  Packet Flows When Host A Sends to Host B on the Bridge Domain 124
  Packet Flows When Host A Sends to Host C From the Bridge Domain to a Routed Interface 124
  Packet Flows When Host C Sends to Host B From a Routed Interface to the Bridge Domain 125
Configuration Examples for IRB 125
  Basic IRB Configuration: Example 125
  IRB With BVI and VRRP Configuration: Example 126

Configuring Link Bundling

Limitations and Compatible Characteristics of Ethernet Link Bundles
Configuring Ethernet Link Bundles
Configuring LACP Fallback
Configuring EFP Load Balancing on an Ethernet Link Bundle
VLANs on an Ethernet Link Bundle
Configuring VLAN over Bundles
  LACP Short Period Time Intervals
  Configuring the Default LACP Short Period Time Interval
  Configuring Custom LACP Short Period Time Intervals
  Configuring VPWS Cross-Connects in MC-LAG
  Configuring VPLS in MC-LAG
Information About Configuring Link Bundling
  IEEE 802.3ad Standard
  Link Bundle Configuration Overview
  Link Switchover
  LACP Fallback
  Failure Cases

Configuring Traffic Mirroring

Introduction to Traffic Mirroring
Traffic Mirroring Types 160
Additional Information on Traffic Mirroring 161
Traffic Mirroring Terminology 161
Characteristics of the Source Port 161
Characteristics of the Monitor Session 162
Characteristics of the Destination Port 162
Restrictions 163
How to Configure Traffic Mirroring 164
Configuring Remote Traffic Mirroring 164
Attaching the Configurable Source Interface 166
Traffic Mirroring Configuration Examples 168
Configuring ACLs for Traffic Mirroring 168
Configuring UDF-Based ACL for Traffic Mirroring 168
Verifying UDF-based ACL 170
Traffic Mirroring with Physical Interfaces (Local): Example 171
Viewing Monitor Session Status: Example 171
Monitoring Traffic Mirroring on a Layer 2 Interface 172
Troubleshooting Traffic Mirroring 172

CHAPTER 9
Configuring Virtual Loopback and Null Interfaces 177
Information About Configuring Virtual Interfaces 177
Virtual Loopback Interface Overview 177
Prerequisites for Configuring Virtual Interfaces 178
Configuring Virtual Loopback Interfaces 178
Null Interface Overview 180
Configuring Null Interfaces 180
Configuring Virtual IPv4 Interfaces 182

CHAPTER 10
Configuring 802.1Q VLAN Interfaces 185
How to Configure 802.1Q VLAN Interfaces 185
Configuring 802.1Q VLAN Subinterfaces 185
Verification 188
Configuring an Attachment Circuit on a VLAN 188
Removing an 802.1Q VLAN Subinterface 190
Information About Configuring 802.1Q VLAN Interfaces 191
  Subinterfaces 191
  Subinterface MTU 191
  EFPs 192
  Layer 2 VPN on VLANs 192
  Layer 3 QinQ 192

CHAPTER 11
Configuring GRE Tunnels 195
  Configuring GRE Tunnels 195
  IP-in-IP De-capsulation 196
  Single Pass GRE Encapsulation Allowing Line Rate Encapsulation 199
    Configuration 199
    Running Configuration 203
    Verification 206

CHAPTER 12
Configuring Controllers 209
  Optics Controllers 209
  Maintenance Mode 209
  Performance Monitoring 210
  How to Configure Controllers 210
    Configuring Optics Controller 210
    Configuring Port Mode Speed 212
    Configuring Wavelength 215
  Configuring Coherent DSP Controller 217
  Configuring Performance Monitoring 218
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 1: Changes to This Document

<table>
<thead>
<tr>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 2018</td>
<td>Republished for Release 6.3.2.</td>
</tr>
<tr>
<td>September 2017</td>
<td>Initial release of this document.</td>
</tr>
</tbody>
</table>

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly What’s New in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at: 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.
# 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 6.3.x, on page 1

## Interface and Hardware Component Features Added or Modified in IOS XR Release 6.3.x

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Interface Speed</td>
<td>Configuring 1Gig interface's speed</td>
<td>Release 6.3.2</td>
<td>Network Interface Speed, on page 21</td>
</tr>
<tr>
<td>Change in BVI number range</td>
<td>The number range that BVI interface supports is increased to 4294967295.</td>
<td>Release 6.3.1</td>
<td>Bridge-Group Virtual Interface, on page 114</td>
</tr>
<tr>
<td>CFM over bundle</td>
<td>Cisco supports CFM over bundle interfaces, bundle sub-interfaces, and bundle member interfaces.</td>
<td>Release 6.3.1</td>
<td>CFM Over Bundles, on page 89</td>
</tr>
<tr>
<td>LLDP Model Local System Information</td>
<td>The LLDP model local system information feature provides Open-Config (OC) Model support for LLDP.</td>
<td>Release 6.3.2</td>
<td>See the <em>Release Notes for Cisco NCS 5500 Series Routers, IOS XR Release 6.3.2.</em></td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Introduced/Changed in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LLDP Supported on Management Interface</td>
<td>This feature requires support from SPIO as LLDP uses SPIO for both its frame transmission and reception</td>
<td>Release 6.3.2</td>
<td>See the Configuring Ethernet Interfaces chapter of Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR.</td>
</tr>
<tr>
<td>DWDM Tunable Optics</td>
<td>The dense wavelength-division multiplexing (DWDM) wavelengths of the DWDM-SFP10G-C module on the Cisco NCS 5500 Series Aggregation Services Routers is tunable.</td>
<td>Release 6.3.2</td>
<td>Dense Wavelength Division Multiplexing Tunable Optics, on page 31</td>
</tr>
<tr>
<td>Bit Error Rate (BER)</td>
<td>Bit Error Rate (BER) is a number of bit errors per unit time that determines the reliability of a link.</td>
<td>Release 6.3.2</td>
<td>Bit Error Rate, on page 102</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 card is installed and the interfaces are up.

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

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 line card can be instantly configured and actively running during line card bootup.

Another advantage of performing a preconfiguration is that during a card replacement, when the line card 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
Information About Preconfiguring Physical Interfaces

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

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
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.

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 9
- How to Perform Advanced Management Ethernet Interface Configuration, on page 10
- Information About Configuring Management Ethernet Interfaces, on page 16

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.

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:**

```bash
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:**

```bash
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:**

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

or

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

Saves configuration changes.

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

  ```bash
  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, 1000Mb/s, 1000BASE-T, link type is autonegotiation
    loopback not set,
    Last link flapped 00:00:59
    ARP type ARPA, ARP timeout 04:00:00
Last input 00:00:00, output 00:00:02
Last clearing of "show interface" counters never
5 minute input rate 4000 bits/sec, 3 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
    21826 packets input, 4987886 bytes, 0 total input drops
    0 drops for unrecognized upper-level protocol
    Received 12450 broadcast packets, 8800 multicast packets
    0 runts, 0 giants, 0 throttles, 0 parity
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    1192 packets output, 217483 bytes, 0 total output drops
    Output 0 broadcast packets, 0 multicast packets
    0 output errors, 0 underruns, 0 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 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**

**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 100000000
ND advertised retransmit interval is 0 milliseconds
Hosts use stateless autoconfig for addresses.
Outgoing access list is not set
Inbound common access list is not set, access list is not set
Table Id is 0xe0800000
Complete protocol adjacency: 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
Advanced Configuration and Modification of the Management Ethernet Interface

Modifying the MAC Address for a Management Ethernet Interface

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.
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:

- 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.
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 POPs, 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 17
- Information About Configuring Ethernet, on page 21
- Link Layer Discovery Protocol (LLDP), on page 28
- Dense Wavelength Division Multiplexing Tunable Optics, on page 31
- How to Configure Interfaces in Breakout Mode, on page 43

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
```
Enter 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 1448
```

(Optional) Sets the MTU value for the interface.
- The default is 1514 bytes for normal frames and 1518 bytes for 802.1Q tagged frames.
- The range for 100-Gigabit Ethernet mtu values is 64 bytes to 65535 bytes.

**Step 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 1448
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end

Uncommitted changes found, commit them? [yes]: yes
```

```
RP/0/RP0/CPU0:router# show running-config interface HundredGigE 0/5/0/24
interface HundredGigE 0/5/0/24
  mtu 9216
  service-policy input linerate
  service-policy output elinerate
```

```
interface HundredGigE 0/5/0/24
  hardware is HundredGigE, address is 6219.8864.e330 (bia 6219.8864.e330)
  Internet address is 3.24.1.1/24
  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 apilcne, 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
```

```
Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.3.x
```
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.

Note

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

1 Gig 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 the 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 configuring network interface speed in the 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 the 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>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>(dBm)</td>
</tr>
<tr>
<td>0</td>
<td>n/a</td>
<td>0.0</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>Transceiver Temp (C):</td>
<td>0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>Transceiver Voltage (V):</td>
<td>0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>Laser Bias (mA):</td>
<td>0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>Transmit Power (mW):</td>
<td>1.000 1.000 1.000 1.000</td>
</tr>
<tr>
<td>Transmit Power (dBm):</td>
<td>0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>Receive Power (mW):</td>
<td>1.000 1.000 1.000 1.000</td>
</tr>
<tr>
<td>Receive Power (dBm):</td>
<td>0.000 0.000 0.000 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
Auto negotiation 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
0 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>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DOM alarms:
  No alarms

Alarm Thresholds:
  Transceiver Temp (C):
    (H) 0.000 0.000 0.000 0.000
  Transceiver Voltage (V):
    (H) 0.000 0.000 0.000 0.000
  Laser Bias (mA):
    (H) 0.000 0.000 0.000 0.000
  Transmit Power (mW):
    (H) 1.000 1.000 1.000 1.000
  Transmit Power (dBm):
    (H) 0.000 0.000 0.000 0.000
  Receive Power (mW):
    (H) 1.000 1.000 1.000 1.000
  Receive Power (dBm):
    (H) 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: 1000Mbps
  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 a Gig interface speed to 100Mbps and autonegotiate other parameters:

```
#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:

```
#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, 1 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.

#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 Lane</th>
<th>Tx Power (nm)</th>
<th>Rx Power (dBm)</th>
<th>Laser Bias (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</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   
Transceiver Voltage (V): 0.000   
Laser Bias (mA): 0.000   
Transmit Power (mW): 1.000   
Receive Power (mW): 1.000   
Receive Power (dBm): 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  

**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.

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

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

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/RSP0/CPU0:router # configure
2. RP/0/RSP0/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 Globally

• 1.0.8802.1.1.2.1.4.1.1.10
• 1.0.8802.1.1.2.1.4.1.1.11
• 1.0.8802.1.1.2.1.4.1.1.12
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
-----------------
<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></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
### Configuring the DWDM Tunable Optics

<table>
<thead>
<tr>
<th>Rx Power Threshold (dBm)</th>
<th>-2.9</th>
<th>-30.9</th>
<th>-7.0</th>
<th>-26.9</th>
</tr>
</thead>
<tbody>
<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

// 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**

<table>
<thead>
<tr>
<th>ITU Ch</th>
<th>G.694.1 Ch Num</th>
<th>Frequency (THz)</th>
<th>Wavelength (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-35</td>
<td>191.35</td>
<td>1566.723</td>
</tr>
<tr>
<td>2</td>
<td>-34</td>
<td>191.40</td>
<td>1566.314</td>
</tr>
<tr>
<td>3</td>
<td>-33</td>
<td>191.45</td>
<td>1565.905</td>
</tr>
<tr>
<td>4</td>
<td>-32</td>
<td>191.50</td>
<td>1565.496</td>
</tr>
<tr>
<td>5</td>
<td>-31</td>
<td>191.55</td>
<td>1565.087</td>
</tr>
<tr>
<td>6</td>
<td>-30</td>
<td>191.60</td>
<td>1564.679</td>
</tr>
<tr>
<td>7</td>
<td>-29</td>
<td>191.65</td>
<td>1564.271</td>
</tr>
<tr>
<td>8</td>
<td>-28</td>
<td>191.70</td>
<td>1563.863</td>
</tr>
<tr>
<td>9</td>
<td>-27</td>
<td>191.75</td>
<td>1563.455</td>
</tr>
<tr>
<td>10</td>
<td>-26</td>
<td>191.80</td>
<td>1563.047</td>
</tr>
<tr>
<td>11</td>
<td>-25</td>
<td>191.85</td>
<td>1562.640</td>
</tr>
<tr>
<td>12</td>
<td>-24</td>
<td>191.90</td>
<td>1562.233</td>
</tr>
<tr>
<td>13</td>
<td>-23</td>
<td>191.95</td>
<td>1561.826</td>
</tr>
<tr>
<td>14</td>
<td>-22</td>
<td>192.00</td>
<td>1561.419</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>15</td>
<td>-21</td>
<td>192.05</td>
<td>1561.013</td>
</tr>
<tr>
<td>16</td>
<td>-20</td>
<td>192.10</td>
<td>1560.606</td>
</tr>
<tr>
<td>17</td>
<td>-19</td>
<td>192.15</td>
<td>1560.200</td>
</tr>
<tr>
<td>18</td>
<td>-18</td>
<td>192.20</td>
<td>1559.794</td>
</tr>
<tr>
<td>19</td>
<td>-17</td>
<td>192.25</td>
<td>1559.389</td>
</tr>
<tr>
<td>20</td>
<td>-16</td>
<td>192.30</td>
<td>1558.983</td>
</tr>
<tr>
<td>21</td>
<td>-15</td>
<td>192.35</td>
<td>1558.578</td>
</tr>
<tr>
<td>22</td>
<td>-14</td>
<td>192.40</td>
<td>1558.173</td>
</tr>
<tr>
<td>23</td>
<td>-13</td>
<td>192.45</td>
<td>1557.768</td>
</tr>
<tr>
<td>24</td>
<td>-12</td>
<td>192.50</td>
<td>1557.363</td>
</tr>
<tr>
<td>25</td>
<td>-11</td>
<td>192.55</td>
<td>1556.959</td>
</tr>
<tr>
<td>26</td>
<td>-10</td>
<td>192.60</td>
<td>1556.555</td>
</tr>
<tr>
<td>27</td>
<td>-9</td>
<td>192.65</td>
<td>1556.151</td>
</tr>
<tr>
<td>28</td>
<td>-8</td>
<td>192.70</td>
<td>1555.747</td>
</tr>
<tr>
<td>29</td>
<td>-7</td>
<td>192.75</td>
<td>1555.343</td>
</tr>
<tr>
<td>30</td>
<td>-6</td>
<td>192.80</td>
<td>1554.940</td>
</tr>
<tr>
<td>31</td>
<td>-5</td>
<td>192.85</td>
<td>1554.537</td>
</tr>
<tr>
<td>32</td>
<td>-4</td>
<td>192.90</td>
<td>1554.134</td>
</tr>
<tr>
<td>33</td>
<td>-3</td>
<td>192.95</td>
<td>1553.731</td>
</tr>
<tr>
<td>34</td>
<td>-2</td>
<td>193.00</td>
<td>1553.329</td>
</tr>
<tr>
<td>35</td>
<td>-1</td>
<td>193.05</td>
<td>1552.926</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>193.10</td>
<td>1552.524</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>193.15</td>
<td>1552.122</td>
</tr>
<tr>
<td>38</td>
<td>2</td>
<td>193.20</td>
<td>1551.721</td>
</tr>
<tr>
<td>39</td>
<td>3</td>
<td>193.25</td>
<td>1551.319</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>193.30</td>
<td>1550.918</td>
</tr>
<tr>
<td>41</td>
<td>5</td>
<td>193.35</td>
<td>1550.517</td>
</tr>
<tr>
<td>42</td>
<td>6</td>
<td>193.40</td>
<td>1550.116</td>
</tr>
<tr>
<td>43</td>
<td>7</td>
<td>193.45</td>
<td>1549.715</td>
</tr>
<tr>
<td>44</td>
<td>8</td>
<td>193.50</td>
<td>1549.315</td>
</tr>
<tr>
<td>45</td>
<td>9</td>
<td>193.55</td>
<td>1548.915</td>
</tr>
<tr>
<td>46</td>
<td>10</td>
<td>193.60</td>
<td>1548.515</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<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 the DWDM Tunable Optics

// 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

Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.3.x
The document contains information about an optical transceiver, including alarm status, detected alarms, and threshold values. It also shows the configuration of the DWDM tunable optics, including wavelength changes and status information. The transceiver vendor details are also provided, along with form factor, optical type, and part number.
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

<table>
<thead>
<tr>
<th>Optics type</th>
<th>SFP+ 10G DWDM Tunable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CISCO-OCLARO</td>
</tr>
<tr>
<td>OUI Number</td>
<td>00.0b.40</td>
</tr>
<tr>
<td>Part Number</td>
<td>TRS7000FNCA033</td>
</tr>
<tr>
<td>Rev Number</td>
<td>0000</td>
</tr>
<tr>
<td>Serial Number</td>
<td>ONT2038009B</td>
</tr>
<tr>
<td>PID</td>
<td>DWDM-SFP10G-C</td>
</tr>
<tr>
<td>VID</td>
<td>V01</td>
</tr>
</tbody>
</table>

// Change Channel
RP/0/RP0/CP0:ios#conf t
Tue Sep 5 08:29:03.648 UTC
RP/0/RP0/CP0:ios#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
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
!
RP/0/RP0/CPU0:uut# show int br location 0/1/CPU0 | i Te0/1/0/28
Sat Oct 12 00:11:38.609 UTC
| up | up | ARPA 10000 10000000
Te0/1/0/27/0 | up | up | ARPA 10000 10000000
Te0/1/0/27/1 | up | up | ARPA 10000 10000000
Te0/1/0/27/2 | up | up | ARPA 10000 10000000
Te0/1/0/27/3 | up | up | ARPA 10000 10000000
Te0/1/0/28/0 | up | up | ARPA 10000 10000000
Te0/1/0/28/1 | up | up | ARPA 10000 10000000
Te0/1/0/28/2 | up | up | ARPA 10000 10000000
Te0/1/0/28/3 | up | up | ARPA 10000 10000000
```
Verify a Breakout Configuration
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>
</tbody>
</table>

- Configuring Ethernet OAM, on page 45
- Information About Configuring Ethernet OAM, on page 46
- How to Configure Ethernet OAM, on page 61
- CFM Over Bundles, on page 89
- Y.1731 Performance Monitoring, on page 89
- Bit Error Rate, on page 102
- Configuration Examples for Ethernet OAM, on page 103

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>
</tbody>
</table>
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.

**EFD**

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

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

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

Note

EFD can only be used for down MEPs. When EFD is used to shut down the interface, the 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.

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.

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.
Figure 2: CFM Maintenance Domain

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

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

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

Each organization uses a different CFM maintenance domain.

This figure shows an example of the different levels of maintenance domains in a network.
In CFM diagrams, the conventions are that triangles represent MEPs, pointing in the direction that the MEP sends CFM frames, and circles represent MIPs. For more information about MEPs and MIPs, see the “Maintenance Points” section on page 71.

**Figure 3: Different CFM Maintenance Domains Across a Network**

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

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

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

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

Note

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

Maintenance Points

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

A maintenance point is always associated with a particular CFM service, and therefore with a particular maintenance domain at a particular level. Maintenance points generally only process CFM frames at the same level as their associated maintenance domain. Frames at a higher maintenance level are always forwarded transparently, while frames at a lower maintenance level are normally dropped. This helps enforce the maintenance domain hierarchy described in the “Maintenance Domains” section on page 69, 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.

- NCS5500 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) or routed (Layer 3) interfaces.

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

Note

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

CFM Protocol Messages

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

This section describes the following CFM messages:

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

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

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

Figure 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: \texttt{ethernet cfm domain SMB level 3 id null}

An example for configuring SMAN is: \texttt{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>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>
<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 Up to 4K UMC values</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 1min 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 1min intervals.
- The status of the interface where the MEP is operating—for example, whether the interface is up, down, STP blocked, and so on.

\textbf{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.

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

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

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

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

Example:

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

#### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window milliseconds 60000</code></td>
<td>The range is from 100 to 60000. The default value is 100.</td>
</tr>
<tr>
<td><code>Step 9</code></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. The range is from 0 to 100000. The default low threshold is 1. To obtain the number of frames, the configured time interval is converted to a window size in frames using the interface speed. For example, for a 1Gbps interface, the IEEE defines minimum frame size as 512 bits. So, we get a maximum of approximately 1.5 million frames per second. If the window size is configured to be 8 seconds (8000ms) then this would give us a Window of 12 million frames in the specification's definition ofErrored Frame Window. The thresholds for frame-period are measured in errors per million frames. Hence, if you configure a window of 8000ms (that is a window of 12 million frames) and a high threshold of 100, then the threshold would be crossed if there are 1200 errored frames in that period (that is, 100 per million for 12 million).</td>
</tr>
<tr>
<td><code>Step 10</code></td>
<td>(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000. The default value is 6000.</td>
</tr>
<tr>
<td><code>Step 11</code></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. The range is 1 to 900</td>
</tr>
</tbody>
</table>

#### Example:

- **Step 9**
  ```
  RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period threshold low 100 high 1000000
  ```
- **Step 10**
  ```
  RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds window 900000
  ```
- **Step 11**
  ```
  RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 threshold high 900
  ```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 12</strong> exit</td>
<td>The default value is 1.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam-1m)# exit</td>
<td>Exits back to Ethernet OAM mode.</td>
</tr>
<tr>
<td><strong>Step 13</strong> mib-retrieval</td>
<td>Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# mib-retrieval</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> connection timeout &lt;timeout&gt;</td>
<td>Configures the connection timeout period for an Ethernet OAM session as a multiple of the hello interval.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# connection timeout 30</td>
<td>The range is 2 to 30. The default value is 5.</td>
</tr>
<tr>
<td><strong>Step 15</strong> hello-interval {100ms</td>
<td>1s}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# hello-interval 100ms</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> mode {active</td>
<td>passive}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# mode passive</td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong> require-remote mode {active</td>
<td>passive}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# require-remote mode active</td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong> require-remote mib-retrieval</td>
<td>Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval</td>
<td></td>
</tr>
<tr>
<td><strong>Step 19</strong> action capabilities-conflict {disable</td>
<td>efd</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# action capabilities-conflict efd</td>
<td></td>
</tr>
</tbody>
</table>

Note

- If you change the default, the log keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>`action critical-event {disable</td>
<td>error-disable-interface}`</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>`action discovery-timeout {disable</td>
<td>efd</td>
<td>error-disable-interface}`</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>`action dying-gasp {disable</td>
<td>error-disable-interface}`</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>`action high-threshold {error-disable-interface</td>
<td>log}`</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>`action session-down {disable</td>
<td>efd</td>
<td>error-disable-interface}`</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 25</strong> action session-up disable</td>
<td>Specifies that no action is taken on an interface when an Ethernet OAM session is established. The default action is to create a syslog entry.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# action session-up disable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 26</strong> action uni-directional link-fault {disable</td>
<td>efd</td>
<td>error-disable-interface}</td>
<td>Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# action uni-directional link-fault efd</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 27</strong> action wiring-conflict {disable</td>
<td>efd</td>
<td>log}</td>
<td>Specifies the action that is taken on an interface when a wiring-conflict event occurs. The default is to put the interface into error-disable state.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# action session-down efd</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 28</strong> uni-directional link-fault detection</td>
<td>Enables detection of a local, unidirectional link fault and sends notification of that fault to an Ethernet OAM peer.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-eoam)# uni-directional link-fault detection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 29</strong> commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# commit</td>
<td></td>
<td></td>
<td></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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | configure  
Example:  
RP/0/RP0/CPU0:router# configure terminal |
| | Enters global configuration mode. |
| **Step 2** | interface [FastEthernet | HundredGigE| TenGigE] interface-path-id  
Example:  
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0 |
| | Enters interface configuration mode and specifies the Ethernet interface name and notation `rack/slot/module/port`.  
Note  
*The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.* |
| **Step 3** | ethernet oam  
Example:  
RP/0/RP0/CPU0:router(config-if)# ethernet oam |
| | Enables Ethernet OAM and enters interface Ethernet OAM configuration mode. |
| **Step 4** | profile profile-name  
Example:  
RP/0/RP0/CPU0:router(config-if-eoam)# profile Profile_1 |
| | Attaches the specified Ethernet OAM profile (`profile-name`), and all of its configuration, to the interface. |
| **Step 5** | commit  
Example: |
| | Saves the configuration changes to the running configuration file and remains within the configuration session. |
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 RP/0/RP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface
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 |  Enters global configuration mode.  |
### Command or Action

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

### Purpose

- Enters interface configuration mode and specifies the Ethernet interface name and notation `rack/slot/module/port`.  
  - **Note**: The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
- Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.
- 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.
- Saves the configuration changes to the running configuration file and remains within the configuration session.
- Ends the configuration session and exits to the EXEC mode.

### Verifying the Ethernet OAM Configuration

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

```
RP/0/RP0/CPU0:router# show ethernet oam configuration  
Thu Aug 5 22:07:06.870 DST  
GigabitEthernet0/4/0/0:  
   Hello interval: 1s  
   Mib retrieval enabled: N  
   Uni-directional link-fault detection enabled: N  
   Configured mode: Active  
   Connection timeout: 5  
   Symbol period window: 0  
   Symbol period low threshold: 1  
   Symbol period high threshold: None  
   Frame window: 1000  
   Frame low threshold: 1  
   Frame high threshold: None  
   Frame period window: 1000  
   Frame period low threshold: 1  
   Frame period high threshold: None
```
Frame seconds window: 60000
Frame seconds low threshold: 1
Frame seconds high threshold: None
High threshold action: None
Link fault action: Log
Dying gasp action: Log
Critical event action: Log
Discovery timeout action: Log
Capabilities conflict action: Log
Wiring conflict action: Error-Disable
Session up action: Log
Session down action: Log
Require remote mode: Ignore
Require remote MIB retrieval: N

**Configuring Ethernet CFM**

To configure Ethernet CFM, perform the following tasks:

---

**Note**

CFM is not supported for the following:
- L3 Interfaces and Sub-Interfaces
- 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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router# configure</code></td>
<td></td>
</tr>
</tbody>
</table>
## 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ethernet cfm</td>
<td>Enters Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>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 CFM domain configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>The level must be specified.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td>The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.</td>
</tr>
<tr>
<td>4</td>
<td>traceroute cache hold-time minutes size entries</td>
<td>(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.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# commit</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Services for a CFM Maintenance Domain

You can configure up to 32000 CFM services for a maintenance domain. To configure services for a CFM maintenance domain, perform the following steps:
SUMMARY STEPS

1. configure
2. ethernet cfm
3. domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]
4. service service-name {down-meps | xconnect group xconnect-group-name p2p xconnect-name} [id [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>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ethernet cfm</td>
<td>Enters Ethernet CFM configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]</td>
<td>Creates and names a container for all domain configurations at a specified maintenance level, and enters CFM domain configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string string</td>
<td></td>
</tr>
<tr>
<td>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]</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group x1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<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.*
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
<td></td>
</tr>
<tr>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
<td></td>
</tr>
</tbody>
</table>

### 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 ice-string unc-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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RF0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>ethernet cfm</strong></td>
<td>Enters Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RF0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>domain</strong> domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]</td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The <strong>id</strong> is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RF0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Ethernet OAM

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> service service-name {down-meps</td>
<td>xconnect group xconnect-group-name p2p xconnect-name} [id [icc-based ice-string umc-string]] [ [number number]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> continuity-check interval time [loss-threshold threshold]</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>Example:</td>
<td></td>
</tr>
<tr>
<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> continuity-check archive hold-time minutes</td>
<td>(Optional) Configures how long information about peer MEPS is stored after they have timed out.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<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> continuity-check loss auto-traceroute</td>
<td>(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<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> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit</td>
<td>• When you use the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>• Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
</tbody>
</table>
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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>configure</strong></td>
<td>Enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RF0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td>2. <strong>ethernet cfm</strong></td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RF0/CPU0:router# ethernet cfm</code></td>
<td></td>
</tr>
<tr>
<td>3. <strong>domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]</strong></td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>The level must be specified. The only supported option is id [null] for less than 1 min interval MEPS.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.</td>
<td></td>
</tr>
</tbody>
</table>
| 4. **service service-name {down-meps | xconnect group xconnect-group-name p2p xconnect-name} [id [icc-basedicc-string umc-string] | [number number]}
<p>| <strong>Example:</strong> | 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 where MIPs and up MEPs will be created. | |</p>
<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 xconnect group X1</code></td>
<td>The <code>id</code> sets the short MA name.</td>
</tr>
</tbody>
</table>

**Step 5**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`mip auto-create {all</td>
<td>lower-mep-only} {ccm-learning}`</td>
</tr>
</tbody>
</table>

**Example:**

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

**Step 6**

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

**Example:**

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

---

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

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

**SUMMARY STEPS**

1. `configure`
2. `ethernet cfm`
3. `domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]`
4. `service service-name {bridge group bridge-domain-group bridge-domain bridge-domain-name | down-meps | xconnect group xconnect-group-name p2p xconnect-name}[id [icc-based ice-string umc-string]] [string text] [number number] [vlan-id id-number] [vpn-id oui-vpnid]`
5. `mep crosscheck`
6. `mep-id mep-id-number [mac-address mac-address]`
7. `end or commit`
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ethernet cfm</td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string]]</td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The 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></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>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. The id sets the short MA name.</td>
</tr>
<tr>
<td></td>
<td>down-meps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xconnect group xconnect-group-name p2p xconnect-name] [id [icc-based ice-string ume-string] [string text] [number number] [vlan-id id-number] [vpn-id oui-vpnid]]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>mep crosscheck</td>
<td>Enters CFM MEP crosscheck configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>mep-id mep-id-number [mac-address mac-address]</td>
<td>Enables cross-check on a MEP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Other Options for a CFM Service

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ethernet cfm</td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>3</td>
<td>domain &lt;domain-name&gt; level &lt;level-value&gt; [id &lt;id&gt;] [dns &lt;DNS-name&gt;] [mac &lt;mac&gt;] [string &lt;string&gt;]</td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>service &lt;service-name&gt; [bridge group &lt;bridge-domain-name&gt;] [down-meps] [connect group &lt;xconnect-name&gt;]</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. The id sets the short MA name.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge_group BD1 bridge-domain B1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>maximum-meps &lt;number&gt;</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></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>log {ais</td>
<td>continuity-check errors</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<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.
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 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> 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> interface {HundredGigE</td>
<td>TenGigE} interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface {HundredGigE</td>
<td>TenGigE</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1</td>
<td></td>
</tr>
</tbody>
</table>

Note: Use the show interfaces command to see a list of all interfaces currently configured on the router.

Naming convention is interface-path-id.subinterface. The period in front of the subinterface value is required as part of the notation.
### Configuring CFM MEPs

#### Command or Action

**Step 4**
- `vrf vrf-name`
  
  **Example:**
  
  ```
  RP/0/RP0/CPU0:router(config-if)# vrf vrf_A
  ```

  **Purpose:** Configures a VRF instance and enters VRF configuration mode.

**Step 5**
- `interface {HundredGigE | TenGigE} interface-path-id`
  
  **Example:**
  
  ```
  RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1
  ```

  **Purpose:** Type of Ethernet interface on which you want to create a MEP. Enter HundredGigE or TenGigE and the physical interface or virtual interface.

  **Note:**
  - Use the `show interfaces` command to see a list of all interfaces currently configured on the router.

**Step 6**
- `ethernet cfm`
  
  **Example:**
  
  ```
  RP/0/RP0/CPU0:router(config-if)# ethernet cfm
  ```

  **Purpose:** Enters interface Ethernet CFM configuration mode.

**Step 7**
- `mep domain domain-name service service-name mep-id id-number`
  
  **Example:**
  
  ```
  RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1
  ```

  **Purpose:** Creates a maintenance end point (MEP) on an interface and enters interface CFM MEP configuration mode.

**Step 8**
- `cos cos`
  
  **Example:**
  
  ```
  RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7
  ```

  **Purpose:** (Optional) Configures the class of service (CoS) (from 0 to 7) for all CFM packets generated by the MEP on an interface. If not configured, the CoS is inherited from the Ethernet interface.

  **Note:** For Ethernet interfaces, the CoS is carried as a field in the VLAN tag. Therefore, CoS only applies to interfaces where packets are sent with VLAN tags. If the `cos (CFM)` command is executed for a MEP on an interface that does not have a VLAN encapsulation configured, it will be ignored.

**Step 9**
- `end` or `commit`
  
  **Example:**
  
  ```
  RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit
  ```

  **Purpose:** 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.
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> 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 Ethernet CFM global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain name level level</td>
<td>Specifies the domain and domain level.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service <em>name</em> bridge group <em>name</em> bridge-domain <em>name</em></td>
<td>Specifies the service, bridge group, and bridge domain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> service <em>name</em> xconnect group <em>xconnect-group-name</em> p2p</td>
<td>Specifies the service and cross-connect group and name.</td>
</tr>
<tr>
<td>xconnect-name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> ais transmission [interval {1s</td>
<td>1m}] [cos cos]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> log ais</td>
<td>Configures AIS logging for a CFM domain service to indicate when AIS or LCK packets are received.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

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

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

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring 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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface gigabitethernet interface-path-id</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router# interface TenGigE 0/0/2</td>
</tr>
<tr>
<td>Step 3</td>
<td>ethernet cfm</td>
<td>Enters Ethernet CFM interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# ethernet cfm</td>
</tr>
<tr>
<td>Step 4</td>
<td>ais transmission up interval 1m cos cos</td>
<td>Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7</td>
</tr>
<tr>
<td>Step 5</td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When you issue the <strong>end</strong> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entering <code>no</code></td>
<td>exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td>Entering <code>cancel</code></td>
<td>leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td>Use the <code>commit</code> command</td>
<td>to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>

---

### 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> <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</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>ethernet cfm</code></td>
<td>Enters Ethernet CFM global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config)# ethernet cfm</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>domain name level level</code></td>
<td>Specifies the domain and domain level.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>service name bridge group name bridge-domain name</code></td>
<td>Specifies the service, bridge group, and bridge domain.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### 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>RP/0/RP0/CPU0:router(config-cfm-dmn)# service S2 bridge group BG1 bridge-domain BD2</td>
<td>Specifies the number of tags in CFM packets. Currently, the only valid value is 1.</td>
</tr>
</tbody>
</table>

**Step 5**

tag number

**Example:**

RP/0/RP0/CPU0:router(config-cfm-dmm-svc)# tags 1

**Step 6**

d or commit

**Example:**

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

Saves configuration changes.

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

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

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

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

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

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

---

### Verifying the 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>show ethernet cfm configuration-errors [domain domain-name] [interface interface-path-id ]</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>Displays a list of local maintenance points.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface type interface-path-id [mep</td>
<td>mip]</td>
</tr>
</tbody>
</table>

---

**Note**

After you configure CFM, the error message, `cfmd[317]: %L2-CFM-5-CCM_ERROR_CCMS_MISSED : Some received CCMs have not been counted by the CCM error counters`, may display. This error message does not have any functional impact and does not require any action from you.
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# 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# 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 0001.0203.0400 [Down] FDB
   0000-0001.0203.0400 TenGigE0/0/0/2
   2 abc 0001.0203.0401 [Ok] FDB
   TenGigE0/0/0/2
   Not present
   3 bcd 0001.0203.0402 [Ok] 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.

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.

NCS 5500 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
================================================================================
St ID     MAC Address Port Up/Downtime CcmRcvd SeqErr RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- ----- 
> 4001 70e4.227c.2865 Up 00:01:27 0 0 0 0

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

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 $1$QJT3$94M5/wK5J0v/lpAu/wz31/
!
line console
exec-timeout 0 0
!
eternet 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 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 TenGigE0/0/0/10.1
  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 TenGigE0/0/10.1
    interface FortyGigE0/0/1/2.1
  
!
end

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

Bucket started at 12:11:10 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)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
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: 0.001ms; 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 L2 transport interfaces, such as Physical, Bundle interfaces, L2 sub-interfaces, Pseudowire Head End interfaces or Attachment Circuits. Transport network can be EVPN or BGP-MPLS.
- Up and Down MEPs.
- Transparently pass SLM packets through MIP without punting to 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 TenGigE0/0/10.1 l2transport
    encapsulation dot1q 1
    ethernet cfm
    mep domain DOWNO 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:
Configuring Synthetic Loss Measurement

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 TenGigE0/0/0/10.1, 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
exe-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
!
    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 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
!  
! 12vpn
  xconnect group g1
  p2p p1
  interface TenGigE0/0/10.1
  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 low 10000000 high 60000000
    frame window 60
    frame-threshold low 10000000 high 60000000
    frame-period window 60000
    frame-period threshold low 100 high 12000000
    frame-seconds window 900000
    frame-seconds threshold 3 threshold 900
  exit
mib-retrieval
  connection timeout 30
  require-remote mode active
  require-remote mib-retrieval
  action dying-gasp error-disable-interface
  action critical-event error-disable-interface
  action discovery-timeout error-disable-interface
  action session-down error-disable-interface
  action capabilities-conflict error-disable-interface
  action wiring-conflict error-disable-interface
  action remote-loopback error-disable-interface
commit
```

Configuring Ethernet OAM Features on an Individual Interface: Example

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

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

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

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

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

Clearing Ethernet OAM Statistics on an Interface: Example

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

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

Enabling SNMP Server Traps on a Router: Example

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

```
configure terminal
```
ethernet oam profile Profile_1
snmp-server traps ethernet oam events

Configuration Examples for Ethernet CFM

This section includes the following examples:

Ethernet CFM Domain Configuration: Example

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

```plaintext
configure
eternet 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:

```plaintext
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:

```plaintext
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:

```plaintext
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:

```plaintext
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
```

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
```

```
Domain/Level  Service    Interface    Type  ID     MAC
--------------- ----------------------- ------- ------- ---------------
fig/5          bay         Gi0/10/0/12  Dn MEP  2  44:55:66
fig/5          bay         Gi0/0/1/0    MIP    55:66:77
fred/3         barney      Gi0/1/0/0    Dn MEP  5  66:77:88!
```

Example 2

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

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

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

* An Up MEP is configured for this domain on interface 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):

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

<table>
<thead>
<tr>
<th>Domain</th>
<th>Interface (State)</th>
<th>Dir</th>
<th>MEPs/Err</th>
<th>RD</th>
<th>Defects</th>
<th>AIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>foo (level 6), Service bar</td>
<td>100 Gi1/0/0/1 (Up)</td>
<td>Up</td>
<td>0/0</td>
<td>N</td>
<td>A</td>
<td>L7</td>
</tr>
</tbody>
</table>

Example 4

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

```sh
RP/0/RP0/CPU0# 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 (archived)
- C - Config (our ID received)
- M - Missing (cross-check)
- X - Cross-connect (wrong MAID)
- U - Unexpected (cross-check)
- P - Peer port down

Domain fred (level 7), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
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 TenGigE0/0/0/1 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: 0
    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 TenGigE0/0/0/2 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
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:

```
RP/0/RP0/CPU0:router# show ethernet cfm interfaces ais
```

<table>
<thead>
<tr>
<th>Interface (State)</th>
<th>AIS</th>
<th>Dir</th>
<th>L Defects</th>
<th>Levels</th>
<th>Trigger</th>
<th>Via</th>
<th>Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>TenGigE0/0/0/0 (Up)</td>
<td>Dn</td>
<td>5</td>
<td>RPC 6</td>
<td>7</td>
<td>1s 01:32:56 ago</td>
<td>5576</td>
<td></td>
</tr>
<tr>
<td>TenGigE0/0/0/0 (Up)</td>
<td>Up</td>
<td>0</td>
<td>M 2,3</td>
<td>5</td>
<td>1s 00:16:23 ago</td>
<td>983</td>
<td></td>
</tr>
<tr>
<td>TenGigE0/0/0/1 (Down)</td>
<td>Up</td>
<td>D</td>
<td></td>
<td>7</td>
<td>60s 01:02:44 ago</td>
<td>3764</td>
<td></td>
</tr>
</tbody>
</table>

**show ethernet cfm local meps Command: Examples**

**Example 1: Default**

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

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

<table>
<thead>
<tr>
<th>A - AIS received</th>
<th>I - Wrong interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>R - Remote Defect received</td>
<td>V - Wrong Level</td>
</tr>
<tr>
<td>L - Loop (our MAC received)</td>
<td>T - Timed out (archived)</td>
</tr>
<tr>
<td>C - Config (our ID received)</td>
<td>M - Missing (cross-check)</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

```
Interface state: Up MAC address: 1122.3344.5566
Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

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

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

```
Interface state: Up MAC address: 1122.3344.5566
Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

CCM generation enabled: No
AIS generation enabled: Yes (level: 7, interval: 1s)
Sending AIS: Yes (started 01:32:56 ago)
Receiving AIS: Yes (from lower MEP, started 01:32:56 ago)
```
Domain fred (level 5), Service barney
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: 0 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.*
CHAPTER 6

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 113
- Bridge-Group Virtual Interface, on page 114
- Supported Features on a BVI, on page 114
- BVI Interface and Line Protocol States, on page 115
- Prerequisites for Configuring IRB, on page 115
- Restrictions for Configuring IRB, on page 116
- How to Configure IRB, on page 117
- Additional Information on IRB, on page 123
- Packet Flows Using IRB, on page 123
- Configuration Examples for IRB, on page 125

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 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`
  - `arp timeout`
**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**

- BVI supports number range from 1 to 4294967295.

---

**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 bridgeport 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.

SUMMARY STEPS

1. configure
2. interface bvi identifier
3. arp purge-delay seconds
4. arp timeout seconds
5. bandwidth rate
6. mtu bytes
7. 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  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 4  
**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 214448000 seconds. The default is 14,400 seconds (4 hours).

Step 5  
**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 1000000 (10 Gbps).

Step 6  
**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 7  
**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)?
[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:
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.
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 into 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 as 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
!
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 TenGigE0/0/0/8
    routed interface BVI 100
  interface TenGigE0/0/0/8
    l2transport
  interface BVI 100
    ipv4 address 10.21.1.1 255.255.255.0
  router vrrp
    interface BVI 100
    vrrp 1 ipv4 10.21.1.100
    vrrp 1 priority 100
CHAPTER 7

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 128
- Configuring Ethernet Link Bundles, on page 129
- Configuring LACP Fallback, on page 133
- Configuring EFP Load Balancing on an Ethernet Link Bundle, on page 134
- VLANs on an Ethernet Link Bundle, on page 137
- Configuring VLAN over Bundles, on page 138
- LACP Short Period Time Intervals, on page 142
- Configuring the Default LACP Short Period Time Interval, on page 143
- Configuring Custom LACP Short Period Time Intervals, on page 145
- Configuring VPWS Cross-Conects in MC-LAG, on page 150
- Configuring VPLS in MC-LAG, on page 153
- Information About Configuring Link Bundling, on page 155
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 63 bundle interfaces. Link bundles of only physical interfaces are supported.

- A single router supports a maximum of 256 bundle interfaces and a default of 64 member links per bundle.

- If the HQueS 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 HQueS enabled on bundle interfaces, a maximum of 4 priority levels are supported.

- The following limitations apply to the number of supported bundle members with HQueS 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.

- 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.
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
Configuring Ethernet Link Bundles

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:
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
```

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
```

---

**Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.3.x**

131
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.
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

Example:

    RP/0/RP0/CPU0:router# configure

Enters global configuration mode.

Step 2 interface Bundle-Ether bundle-id

Example:

    RP/0/RP0/CPU0:router# interface Bundle-Ether bundle-id

Configuring LACP Fallback

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.
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)# bundle lacp-fallback timeout 4

Enables the LACP Fallback feature.

Step 4  `end` or `commit`

Example:

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:

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:

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.

---

### Configuring EFP Load Balancing on an Ethernet Link Bundle

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

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

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

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

### DETAILED STEPS

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

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

**Step 2**  
**interface Bundle-Ether bundle-id l2transport**  
*Example:*

```
RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3 l2transport
```
Creates a new Ethernet link bundle with the specified `bundle-id` and with Layer 2 transport enabled. The range is 1 to 65535.

**Step 3**  
**bundle load-balance hash hash-value [auto]**  
*Example:*

```
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1
or
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto
```
Configures all egressing traffic on the fixed members of a bundle to flow through the same physical member link.

- **hash-value**—Numeric value that specifies the physical member link through which all egressing traffic in this bundle will flow. The values are 1 through 8.
- **auto**—The physical member link through which all egressing traffic on this bundle will flow is automatically chosen.

**Step 4**  
**end or commit**  
*Example:*

```
RP/0/RP0/CPU0:router(config-if)# end
or
RP/0/RP0/CPU0:router(config-if)# commit
```
Saves configuration changes.
• When you issue the **end** command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)?

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

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

• 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.

```bash
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.

```bash
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:

```bash
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:
<table>
<thead>
<tr>
<th>Port:</th>
<th>LON</th>
<th>ULID</th>
<th>BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Te0/0/0/26</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Sub-interface Information:
<table>
<thead>
<tr>
<th>Sub-interface</th>
<th>Type</th>
<th>Load Balance</th>
<th>Locality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hash</td>
<td>Threshold</td>
<td></td>
</tr>
</tbody>
</table>
### 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.

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:

```config
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
```

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 dot1q vlan-id`
10. `ipv4 address ipv4-address mask`
11. `no shutdown`
12. `exit`
13. Repeat Step 9 through Step 12 to add more VLANs to the bundle you created in Step 2.
14. `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
```
```
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 HundredGigE interface-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 HundredGigE interface-path

Example:

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

Creates a Gigabit Ethernet interface and enters interface configuration mode.

**Step 3**

bundle id number mode active

Example:

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

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

**Step 4**

lacp period short

Example:

```
RP/0/RP0/CPU0:router(config-if)# lacp period short
```
Configures a short period time interval for the sending and receiving of LACP packets, using the default time period of 1000 milliseconds or 1 second.

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

**Example:**

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

or

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

Saves configuration changes.

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

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

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

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

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

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

**Example**

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

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

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

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

cfg
interface HundredGigE 0/1/0/1
lacp period short receive 100
commit

Configuring Custom LACP Short Period Time Intervals

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

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

**SUMMARY STEPS**

1. configure
2. interface Bundle-Ether bundle-id
3. ipv4 address ipv4-address mask
4. bundle minimum-active bandwidth kbps
5. bundle minimum-active links links
6. bundle maximum-active links links
7. exit
8. interface Bundle-Ether bundle-id.vlan-id
9. dot1q vlan vlan-id
10. ipv4 address ipv4-address mask
11. no shutdown
12. exit
13. Repeat Step 7 through Step 12 to add more VLANs to the bundle you created in Step 2.
14. end or commit
15. exit
16. exit
17. show ethernet trunk bundle-ether instance
18. configure
19. interface \{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`
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).
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:*

```
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 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
```
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.

Configuring VPWS Cross-Connects in MC-LAG

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

SUMMARY STEPS

1. configure
2. l2vpn

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

**DETAILED STEPS**

**Step 1**

configure  
Example:

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

Enters global configuration mode.

**Step 2**

`l2vpn`  
Example:

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

Enters L2VPN configuration mode.

**Step 3**

`pw-status`  
Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn)# pw-status
```

Enables pseudowire status.  

**Note**  
- When the attachment circuit changes redundancy state to Active, Active pw-status is sent over the primary and backup pseudowires.  
  When the attachment circuit changes redundancy state to Standby, Standby pw-status is sent over the primary and backup pseudowires.

**Step 4**

`xconnect group group-name`  
Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group grp_1
```

Enters the name of the cross-connect group.

**Step 5**

`p2p xconnect-name`  
Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p p1
```
Enters a name for the point-to-point cross-connect.

**Step 6**

```
interface type interface-path-id
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1
```

Specifies the interface type ID.

**Step 7**

```
neighbor A.B.C.D pw-id pseudowire-id
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 10.2.2.2 pw-id 2000
```

Configures the pseudowire segment for the cross-connect.

Optionally, you can disable the control word or set the transport-type to Ethernet or VLAN.

**Step 8**

```
pw-class {class-name}
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# pw-class c1
```

Configures the pseudowire class template name to use for the pseudowire.

**Step 9**

```
backup neighbor A.B.C.D pw-id pseudowire-id
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# backup neighbor 10.2.2.2 pw-id 2000
```

Adds a backup pseudowire.

**Step 10**

```
pw-class {class-name}
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# pw-class c2
```

Configures the pseudowire class template name to use for the backup pseudowire.

**Step 11**

```
end or commit
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# end
```

or

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# 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 VPLS in MC-LAG

Perform this task to configure VPLS in MC-LAG.

### SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **pw-status**
4. **bridge group** *bridge-group-name*
5. **bridge-domain** *bridge-domain-name*
6. **interface** type *interface-path-id*
7. **vfi** *{vfi-name}*
8. **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*
9. **pw-class** *{class-name}*
10. **end** or **commit**

### DETAILED STEPS

**Step 1**

**configure**

**Example:**

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

Enters global configuration mode.

**Step 2**

**l2vpn**

**Example:**

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

Enters L2VPN configuration mode.

**Step 3**

**pw-status**
Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn)# pw-status
```

(Optional) Enables pseudowire status.

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

**Step 4**

**bridge group** *bridge-group-name*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group csco
```

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg)#
```

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

**Step 5**

**bridge-domain** *bridge-domain-name*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain abc
```

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#
```

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

**Step 6**

**interface** *type* *interface-path-id*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface Bundle-Ether 1.1
```

Specifies the interface type ID.

**Step 7**

**vfi** *{vfi-name}*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# vfi vfi-east
```

Enters virtual forwarding instance (VFI) configuration mode.

**Step 8**

**neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi)# neighbor 10.2.2.2 pw-id 2000
```

Configures the pseudowire segment for the cross-connect.

Optionally, you can disable the control word or set the transport-type to Ethernet or VLAN.

**Step 9**

**pw-class** *{class-name}*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# pw-class canada
```

Configures the pseudowire class template name to use for the pseudowire.

**Step 10**

**end** or **commit**
Example:

RP/0/RSP0/CPU0:router(config-l2vpn-bq-bd-vfi-pw)# end

or

RP/0/RSP0/CPU0:router(config-l2vpn-bq-bd-vfi-pw)# 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.

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.

**Failure Cases**

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

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

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

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

MC-LAG by itself does not provide a means to avoid this situation; resiliency in the connection between the POAs is a requirement. The DHD is given the responsibility of mitigating the problem by setting a limit on the number of links, within the bundle, that can be active. As such only the links connected to one of the POAs can be active at any one point of time.
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 6.1.1</td>
<td>The local SPAN feature was introduced.</td>
</tr>
<tr>
<td>Release 6.1.31</td>
<td>Remote Traffic Monitoring feature was added.</td>
</tr>
</tbody>
</table>

- Introduction to Traffic Mirroring, on page 159
- How to Configure Traffic Mirroring, on page 164
- Configuring Remote Traffic Mirroring, on page 164
- Attaching the Configurable Source Interface, on page 166
- Traffic Mirroring Configuration Examples, on page 168
- Troubleshooting Traffic Mirroring, on page 172

Introduction to Traffic Mirroring

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

Traffic mirroring copies traffic from one or more source ports and sends the copied traffic to one or more destinations for analysis by a network analyzer or other monitoring device. Traffic mirroring does not affect the flow of traffic on the source interfaces or sub-interfaces, and allows the mirrored traffic to be sent to a destination interface or sub-interface.

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

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 directly attached to the destination interface. In other words, all monitored ports are all 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 un-routable, the end 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 `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 is only supported on permit type of access control entries (ACE)s.

  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.
Additional Information on Traffic Mirroring

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 the Source Port

A source port, also called a monitored port, is a routed port that you monitor for network traffic analysis. In a single traffic mirroring session, you can monitor source port traffic. The NCS 5500 Series Router can support 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: 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 cannot be used as a destination port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor for local traffic mirroring. Remote traffic mirroring is supported both in the ingress and egress directions. For bundles, the monitored direction applies to all physical ports in the group.

Figure 12: Network Analysis on a Cisco NCS 5500 Router With Traffic Mirroring

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

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

Monitor sessions have these characteristics:

• A single 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 the Destination Port

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

A destination port has these characteristics:

• A destination port must reside on the same router as the source port for local traffic mirroring. For remote mirroring the destination is always a GRE tunnel.
• A destination port for local mirroring can be any Ethernet physical port, EFP, and GRE tunnel interface, but not a bundle interface. It can be a Layer 2 or Layer 3 transport interface.
• A destination port 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.

Figure 13: Network Analysis on a Cisco NCS 5500 Series Router With Traffic Mirroring

In the above figure, the callouts indicate the following:
1. Source traffic mirroring ports (can be ingress or egress traffic ports).
2. Destination traffic mirroring 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 to avoid default mirroring of traffic. If a Bundle interface is a source interface, configure the ACL(s) on the bundle interface (not bundle members). Also ensure that the ACL(s) configured is a UDK (with capture field) and 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 UDK (with capture) or IPv6 UDK (with capture) 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 ERSPAN is not available.
  • ERSPAN next-hop must have ARP resolved.
    • Any other traffic or protocol will trigger ARP.
• ERSPAN cannot travel over MPLS.
  • Additional routers may encapsulate in MPLS.
• ERSPAN decapsulation is not supported.
• ERSPAN does not work if the GRE next hop is reachable over sub-interface. For ERSPAN to work, the next hop must be reachable over the main interface.
• SPAN-ACL is only supported in the Rx direction, that is, in the ingress direction v4 or v6 ACL.
• MPLS traffic cannot be captured with SPAN-ACL.
  • ACL for any MPLS traffic is not supported.
How to Configure Traffic Mirroring

These tasks describe how to configure traffic mirroring:

Configuring Remote Traffic Mirroring

SUMMARY STEPS

1. configure
2. monitor-session session-name
3. destination interface tunnel-ip
4. exit
5. interface type number
6. monitor-session session-name ethernet direction rx-only port-only
7. end or commit
8. show monitor-session [session-name] status [detail] [error]

DETAILED STEPS

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.

**Step 8**

*show monitor-session [session-name] status [detail] [error]*

**Example:**

```
RP/0/RP0/CPU0:router# show monitor-session
```
Attaching the Configurable Source Interface

SUMMARY STEPS

1. configure
2. interface type number
3. ipv4 access-group acl-name {ingress | egress}
4. monitor-session session-name ethernet direction rx-onlyport-level acl
5. acl
6. exit
7. end or commit
8. show monitor-session [session-name] status [detail] [error]

DETAILED STEPS

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-onlyport-level acl
Example:

RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only port-level acl
RP/0/RP0/CPU0:router(config-if-mon)#

Displays information about the traffic mirroring session.
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 overrides any ACL that may be configured on the interface.

Step 6  exit

Example:

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

Exits monitor session configuration mode and returns to interface configuration mode.

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
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.

```
Router(config)# hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port proto frag-bit enable-capture
Router(config)# commit
```

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

**SUMMARY STEPS**

1. configure
2. udf udf-name header {inner | outer} {12 | 13 | 14} offset offset-in-bytes length length-in-bytes
3. hw-module profile tcam format access-list {ipv4 | ipv6} [acl-qualifiers] [ udf1 udf-name1 ... udf8 udf-name8] enable-capture

4. ipv4 access-list acl-name

5. permit regular-ace-match-criteria udf udf-name1 value1 ... udf-name8 value8

6. exit

7. interface type number

8. ipv4 access-group acl-name ingress

9. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>udf udf-name header {inner</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 0 length 1 (config-mon)#</td>
</tr>
<tr>
<td></td>
<td>The inner or outer keywords indicate the start of 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.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# udf udf3 header inner l4 10 length 2 (config-mon)#</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 50 length 1 (config-mon)#</td>
</tr>
<tr>
<td></td>
<td>The length keyword specifies the length from the offset, in bytes. Range is from 1 to 4.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>hw-module profile tcam format access-list {ipv4</td>
</tr>
<tr>
<td>Example:</td>
<td>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>
</tr>
<tr>
<td></td>
<td>Adds the user-defined fields to the ACL key definition that is sent to the hardware.</td>
</tr>
<tr>
<td></td>
<td>A reload of the line card is required for the new TCAM profile to take effect.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>ipv4 access-list acl-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Creates the ACL and enters the IP ACL configuration mode.</td>
</tr>
<tr>
<td></td>
<td>The length of acl-name argument can be up to 64 characters.</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
```
Traffic Mirroring with Physical Interfaces (Local): Example

This example shows a 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/RP0/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/RP0/CPU0:router# show monitor-session status detail
Monitor-session sess1
Destination interface is not configured
Source Interfaces
-----------------
TenGigE0/1/0/0
Direction: Both
ACL match: Disabled
Portion: Full packet
Status: Not operational (destination interface not known).
TenGigE0/1/0/1
Direction: Both
ACL match: Disabled
Portion: First 100 bytes

RP/0/RP0/CPU0:router# show monitor-session status error
Monitor-session ms1
Destination interface TenGigE0/2/0/15 is not configured
Source Interface Dir Status
--------------------- ---- ----------------------------------------------------
Monitor-session ms2
Destination interface is not configured
Source Interface Dir Status
```

Interval: Mirror all packets
Status: Not operational (destination not active)
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:

```
RP/0/RP0/CPU0:router(config)# interface TenGigE0/0/0/42
RP/0/RP0/CPU0:router(config-if)# l2transport
RP/0/RP0/CPU0:router(config-if-l2)# monitor-session EASTON ethernet port-level
```

Verification

```
RP/0/RP0/CPU0:router# show monitor-session status
Thu Aug 29 21:42:22.829 UTC
Monitor-session EASTON
Destination interface TenGigE0/0/0/20
Source Interface Dir Status
--------------------- ---- -----------------
Te0/0/0/42 (port) Both Operational
```

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
```
### Configuring Traffic Mirroring

In the preceding example, the line marked as `<Session status>` can indicate one of these configuration errors:

<table>
<thead>
<tr>
<th>Session Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session is not configured globally</td>
<td>The session does not exist in global configuration. Check <code>show run</code> command output to ensure that a session with a correct name has been configured.</td>
</tr>
<tr>
<td>Destination interface <code>&lt;intf&gt;</code> (&lt;down-state&gt;)</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 see 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:

<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 <code>Session status</code> error messages for suggested resolution.</td>
</tr>
</tbody>
</table>
### Troubleshooting Traffic Mirroring

<table>
<thead>
<tr>
<th>Source Interface Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<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 of 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
CHAPTER 9

Configuring Virtual Loopback and Null Interfaces

This module describes the configuration of loopback and null interfaces. Loopback and null interfaces are considered virtual interfaces.

A virtual interface represents a logical packet switching entity within the router. Virtual interfaces have a global scope and do not have an associated location. Virtual interfaces have instead a globally unique numerical ID after their names. Examples are Loopback 0, 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 177

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 interface 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 unreachable 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 Virtual IPv4 Interfaces
CHAPTER 10

Configuring 802.1Q VLAN Interfaces

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

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

The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames. Cisco NCS 5000 Series Router supports VLAN subinterface configuration on 10-Gigabit Ethernet and 100-Gigabit Ethernet interfaces. The range for VLANs is 1-4094.

802.1Q Tagged Frames

The IEEE 802.1Q tag-based VLAN uses an extra tag in the MAC header to identify the VLAN membership of a frame across bridges. This tag is used for VLAN and quality of service (QoS) priority identification. The VLAN ID associates a frame with a specific VLAN and provides the information that switches must process the frame across the network. A tagged frame is four bytes longer than an untagged frame and contains two bytes of Tag Protocol Identifier (TPID) residing within the type and length field of the Ethernet frame and two bytes of Tag Control Information (TCI) which starts after the source address field of the Ethernet frame.

For detailed information on 802.1Q Tagged Frames, see the References for Carrier Ethernet Model section in L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers.

• How to Configure 802.1Q VLAN Interfaces, on page 185
• Information About Configuring 802.1Q VLAN Interfaces, on page 191

How to Configure 802.1Q VLAN Interfaces

This section contains the following procedures:

Configuring 802.1Q VLAN Subinterfaces

This task explains how to configure 802.1Q VLAN subinterfaces. To remove these subinterfaces, see the “Removing an 802.1Q VLAN Subinterface” section.
SUMMARY STEPS

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

DETAILED STEPS

Step 1 configure

Example:

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

Enters global configuration mode.

Step 2 interface {TenGigE | FortyGigE | HundredGigE | Bundle-Ether} interface-path-id.subinterface

Example:

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/4.10
```

Enters subinterface configuration mode and specifies the interface type, location, and subinterface number.

- Replace the `interface-path-id` argument with one of the following instances:
  - Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is `rack/slot/module/port`, and a slash between values is required as part of the notation.
  - Ethernet bundle instance. Range is from 1 through 65535.
  - Replace the `subinterface` argument with the subinterface value. Range is from 0 through 2147483647.
  - Naming notation is `interface-path-id.subinterface`, and a period between arguments is required as part of the notation.

Step 3 encapsulation dot1q

Example:

```
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
```

Sets the Layer 2 encapsulation of an interface.

Step 4 ipv4 address ip-address mask

Example:

```
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24
```

Assigns an IP address and subnet mask to the subinterface.
• Replace `ip-address` with the primary IPv4 address for an interface.

• Replace `mask` with the mask for the associated IP subnet. The network mask can be specified in either of two ways:

  • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.

  • The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.

**Step 5**  
**exit**

**Example:**

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

(Optional) Exits the subinterface configuration mode.

  • The `exit` command is not explicitly required.

**Step 6**  
Repeat Step 2 through Step 5 to define the rest of the VLAN subinterfaces.

**Step 7**  
**end** or **commit**

**Example:**

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

or

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

Saves configuration changes.

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

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

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

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

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

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

**Step 8**  
**show ethernet trunk bundle-ether instance**

**Example:**

```
RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5
```
Verification

This example shows how to verify the configuration of Ethernet interfaces:

```bash
# show ethernet trunk be 1020 Wed May 17 16:43:32.804 EDT
```

<table>
<thead>
<tr>
<th>Trunk</th>
<th>Interface</th>
<th>St.</th>
<th>Ly</th>
<th>MTU</th>
<th>Subs</th>
<th>L2</th>
<th>L3</th>
<th>Sub states</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE1020</td>
<td></td>
<td>Up L3</td>
<td>9100</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Summary: 3 3 0 3 0 0 0

Configuring an Attachment Circuit on a VLAN

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

SUMMARY STEPS

1. configure
2. interface [GigabitEthernet | TenGigE | Bundle-Ether | FortyGigE] interface-path id.subinterface l2transport
3. encapsulation dot1q 100
4. end or commit
5. show interfaces [GigabitEthernet | FortyGigE|Bundle-Ether | TenGigE] interface-path-id.subinterface

DETAILED STEPS

Step 1 configure

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

Enters global configuration mode.

Step 2 interface [GigabitEthernet | TenGigE | Bundle-Ether | FortyGigE] interface-path id.subinterface l2transport

```
Example:
RP/0//CPU0:router(config)# interface TenGigE 0/1/0/1 12transport
```

Enters subinterface configuration and specifies the interface type, location, and subinterface number.

- Replace the `interface-path-id` argument with one of the following instances:
- Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is `rack/slot/module/port`, and a slash between values is required as part of the notation.
- Ethernet bundle instance. Range is from 1 through 65535.
• Replace the subinterface argument with the subinterface value. Range is from 0 through 4095.

• Naming notation is instance.subinterface, and a period between arguments is required as part of the notation.

• You must include the l2transport keyword in the command string; otherwise, the configuration creates a Layer 3 subinterface rather than an AC.

**Step 3**

encapsulation dot1q 100

**Example:**

```
RP/0//CPU0:router (config-subif)# encapsulation dot1q 100
```

Sets the Layer 2 encapsulation of an interface.

**Note** The dot1q vlan command is replaced by the encapsulation dot1q command. It is still available for backward-compatibility, but only for Layer 3 interfaces.

**Step 4**

end or commit

**Example:**

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

or

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

Saves configuration changes.

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

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

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

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

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

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

**Step 5**

**Example:**

```
show interfaces [GigabitEthernet | FortyGigE|Bundle-Ether | TenGigE] interface-path-id.subinterface
```

```
RP/0//CPU0:router# show interfaces TenGigE 0/3/0/0.1
```

(Optional) Displays statistics for interfaces on the router.
Removing an **802.1Q VLAN Subinterface**

This task explains how to remove 802.1Q VLAN subinterfaces that have been previously configured using the Configuring 802.1Q VLAN subinterfaces section in this module.

**SUMMARY STEPS**

1. `configure`
2. `no interface {TenGigE | FortyGigE | HundredGigE | Bundle-Ether} interface-path-id.subinterface`
3. Repeat Step 2 to remove other VLAN subinterfaces.
4. `end` or `commit`

**DETAILED STEPS**

---

Step 1  
`configure`  
**Example:**

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

Enters global configuration mode.

Step 2  
`no interface {TenGigE | FortyGigE | HundredGigE | Bundle-Ether} interface-path-id.subinterface`

**Example:**

```
RP/0/RP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10
```

Removes the subinterface, which also automatically deletes all the configuration applied to the subinterface.

- Replace the `instance` argument with one of the following instances:
- Physical Ethernet interface instance, or with an Ethernet bundle instance. Naming notation is `rack/slot/module/port`, and a slash between values is required as part of the notation.
- Ethernet bundle instance. Range is from 1 through 65535.
- Replace the `subinterface` argument with the subinterface value. Range is from 0 through 2147483647.

Naming notation is `instance.subinterface`, and a period between arguments is required as part of the notation.

Step 3  
Repeat Step 2 to remove other VLAN subinterfaces.

---

Step 4  
`end` or `commit`  
**Example:**

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

or

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

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

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

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

Information About Configuring 802.1Q VLAN Interfaces

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

Subinterfaces

Subinterfaces are logical interfaces created on a hardware interface. These software-defined interfaces allow for segregation of traffic into separate logical channels on a single hardware interface as well as allowing for better utilization of the available bandwidth on the physical interface.

Subinterfaces are distinguished from one another by adding an extension on the end of the interface name and designation. For instance, the Ethernet subinterface 23 on the physical interface designated TenGigE 0/1/0/0 would be indicated by TenGigE 0/1/0/0.23.

Before a subinterface is allowed to pass traffic it must have a valid tagging protocol encapsulation and VLAN identifier assigned. All Ethernet subinterfaces always default to the 802.1Q VLAN encapsulation. However, the VLAN identifier must be explicitly defined.

These are the applicable scale values for sub-interfaces:

• Sub-interface per system = 1024
• Sub-interface per line card = 1024
• Sub-interface per NPU = 1024
• Sub-interface per interface = 512
• Sub-Interface per Core = 512

Subinterface MTU

The subinterface maximum transmission unit (MTU) is inherited from the physical interface with an additional four bytes allowed for the 802.1Q VLAN tag. By default subinterface inherits MTU of physical interface if
the MTU is not configured. We can have maximum 3 different MTU for a subinterface per NPU. For information about Ethernet MTU and Flow Control on Ethernet interfaces, see References for Carrier Ethernet Model section in L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers.

**EFPs**

An Ethernet Flow Point (EFP) is a Metro Ethernet Forum (MEF) term describing abstract router architecture. An EFP is implemented by an Layer 2 subinterface with a VLAN encapsulation. The term EFP is used synonymously with an VLAN tagged L2 subinterface. For more information on EFPs, see the Carrier Ethernet Model chapter in L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers.

**Layer 2 VPN on VLANs**

The Layer 2 Virtual Private Network (L2VPN) feature enables Service Providers (SPs) to provide Layer 2 services to geographically disparate customer sites.

The configuration model for configuring VLAN attachment circuits (ACs) is similar to the model used for configuring basic VLANs, where the user first creates a VLAN subinterface, and then configures that VLAN in subinterface configuration mode. To create an AC, you need to include the `l2transport` keyword in the `interface` command string to specify that the interface is a Layer 2 interface.

VLAN ACs support these modes of L2VPN operation:

- Basic Dot1Q AC—The AC covers all frames that are received and sent with a specific VLAN tag.
- QinQ AC—The AC covers all frames received and sent with a specific outer VLAN tag and a specific inner VLAN tag. QinQ is an extension to Dot1Q that uses a stack of two tags.

Each VLAN on a CE-to-PE link can be configured as a separate L2VPN connection (using either VC type 4 or VC type 5).

For more information about Layer 2 VPN on VLANs and their configuration, see the Implementing Point-to-Point Layer 2 Services chapter in L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers.

**Layer 3 QinQ**

Layer 3 QinQ is an extension of IEEE 802.1 QinQ VLAN tag stacking. This feature enables you to increase the number of VLAN tags in an interface and increments the number of sub-interfaces up to 4094. Hence, with dual tag, the number of VLANs can reach up to 4094*4094. With the L3 QinQ feature with dual tag, interfaces check for IP addresses along with MAC addresses.

This feature supports:

- 802.1Q standards like 0x8100, 0x9100, 0x9200 (used as outer tag ether-type) and 0x8100 (used as inner tag ether-type).
- L3 802.1ad VLAN sub-interfaces, with 0x88a8 as the outer S-tag ether-type.
- Co-existence of L2 and L3 single tagged and double tagged VLANs.
- QinQ and dot1ad over ethernet bundle sub-interfaces.
- Default VRF.
QinQ sub-interfaces support these IP features:

- QoS, with policy that matches outer VLAN (and COS) alone, and not both outer and inner VLANs together (2-level QoS/H-QoS support).
- ACL, Netflow, BFD, ARP.
- Routing protocols – static, BGP, OSFPv2.
- IPV4/IPV6 unicast/multicast.

**Prerequisites:**

1. Enable QinQ dual tag support on L3 sub-interfaces on the NSC 5500 and NCS 560 platforms.
2. Ensure the sub-interface scale is the same as what is supported per platform on single tag/802.1Q case. L3 QinQ feature is enabled on physical interfaces as well as on bundle interfaces.

**Types of sub-interfaces:**

<table>
<thead>
<tr>
<th>Interface type</th>
<th>Outer tag</th>
<th>Inner tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dot1q sub-interface</td>
<td>0x8100</td>
<td>None</td>
</tr>
<tr>
<td>QinQ sub-interface</td>
<td>0x8100</td>
<td>0x8100</td>
</tr>
<tr>
<td>QinQ sub-interface</td>
<td>0x88a8</td>
<td>0x8100</td>
</tr>
<tr>
<td>QinQ sub-interface</td>
<td>0x9100</td>
<td>0x8100</td>
</tr>
<tr>
<td>QinQ sub-interface</td>
<td>0x9200</td>
<td>0x8100</td>
</tr>
</tbody>
</table>

**Limitations:**

MPLS is not supported.

**Example:**

```
Example 1:
interface TenGigE0/0/0/6.111
  mtu 1400
  ipv4 address 10.1.1.1 255.255.255.0
  ipv6 address 10::1/64
  encapsulation dot1q 100 second-dot1q 200

interface Bundle-Ether10.1
  ipv4 address 10.1.2.1 255.255.255.0
  ipv6 address 1002::1/64
  encapsulation dot1ad 10 second-dot1q 20
```

Example 2:
Router(config)# interface gigabitethernet 1/0/0
Router(config-if)# dot1q tunneling ethertype 0x9100
Router(config-if)# interface gigabitethernet 1/0/0.1
Router(config-subif)# encapsulation dot1q 100 second-dot1q 200
Router(config-subif)# ipv4 address 172.16.1.2 255.255.255.0

Example 3:
interface GigabitEthernet0/7/0/2.100
description ** Business Services over DOCSIS **
encapsulation dot1q 100 second-dot1q 200-500
ipv4 address 192.168.212.6 255.255.255.252

Example 4:
interface Bundle-Ether1.2
encapsulation dot1q 3200 second-dot1q 2
ipv4 address 85.42.169.6 255.255.255.252
service-policy input BIZDSLIP_HSIHYP_NOBP_96KBMG
CHAPTER 11

Configuring GRE Tunnels

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

- Configuring GRE Tunnels, on page 195
- IP-in-IP De-capsulation, on page 196
- Single Pass GRE Encapsulation Allowing Line Rate Encapsulation, on page 199

Restrictions for Configuring GRE Tunnels

The following restrictions apply while configuring GRE tunnels:

- NCS5500 Series Routers support up to 500 GRE tunnels.
- Only up to 16 unique source IP addresses are supported for the tunnel source.

Configuration Example

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

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

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

The following illustration describes the basic IP-in-IP encapsulation and decapsulation model.
Use Case: Configure IP-in-IP de-capsulation

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

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

Stacked IP Header in an Encapsulated Packet

The encapsulated packet will have an outer IPv4 header stacked over the original IPv4 header, as shown in the following illustration.
You can use the following sample configuration on the routers to decapsulate the packet as it traverses the IP-in-IP tunnel:

```bash
RP/0/RP0/CPU0# interface tunnel-ip 10
RP/0/RP0/CPU0# tunnel mode ipv4 decap
RP/0/RP0/CPU0# tunnel source loopback 0
RP/0/RP0/CPU0# tunnel destination 10.10.1.2/32
```

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

**Running Configuration**

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

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

### 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 */

```
# 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 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 ucmp 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-if)# metric 20
Router(config-if)# commit
Router(config-if)# exit
!

Router# configure
Router(config)# interface Bundle-Ether111
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-if)# metric 15
Router(config-if)# commit
Router(config-if)# 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 (2906:125000)
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
!

Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.3.x
Router# configure
Router(config)# route-policy TRANSIT1_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:37500000
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy

Running Configuration

/* GRE Single-Pass configuration */

interface tunnel-ip30016
ipv4 address 216.1.1.1 255.255.255.0
ipv6 address 216:1:1::1/64
ipv6 enable
tunnel mode gre ipv4 encap
tunnel source Loopback22
tunnel destination 170.170.170.22
!
/* GRE Entropy(ECMP/UCMP) */

ECMP (ISIS)

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

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
ucmp
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
redistribute connected
!
interface Bundle-Ether3
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 20
!
interface Bundle-Ether111
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 15
!
/
/* ECMP(OSPF) */
/
router ospf 3
nsr
maximum paths 5
address-family ipv4 unicast
area 0
interface Bundle-Ether3
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
!
interface Loopback23
!
interface HundredGigE0/7/0/23
!
!
/* 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-Ether111
!
interface Bundle-Ether112
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 (2906:37500000)
else
pass
endif
end-policy
!

**Verification**

Verify if the tunnel mode GRE encapsulation is enabled.

Router# `show interfaces tunnel-ip 100`

Sun Jul 10 15:49:04.812 VN_TIME
tunnel-ip100 is up, line protocol is up
  Interface state transitions: 2
  Hardware is Tunnel
  Internet address is Unknown
  MTU 1500 bytes, BW 100 Kbit (Max: 100 Kbit)
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation TUNNEL_GRE, loopback not set,
  Tunnel TOS 0
  Tunnel mode GRE IPV4,
  Keepalive is enabled, interval 10 seconds, maximum retry 3
  Tunnel source 172.16.16.1 (GigabitEthernet0_0_0_0), destination 172.16.16.2
  Tunnel TTL 100
  Last input 2d03h, output 2d04h
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  689 packets input, 26212 bytes, 0 total input drops
  0 drops for unrecognized upper-level protocol
  Received 0 broadcast packets, 0 multicast packets
  3 packets output, 192 bytes, 0 total output drops
  Output 0 broadcast packets, 0 multicast packets

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

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

Number of tunnel updates to retry: 0
Number of tunnel updates retried: 0
Number of tunnel retries failed: 0
Platform:
Recycle VoQ: 48
  ReceivedBytes  ReceivedPackets  ReceivedKbps
  DroppedBytes  DroppedPackets  DroppedKbps
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
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>COS0</th>
<th>ReceivedPkts</th>
<th>ReceivedBytes</th>
<th>DroppedPkts</th>
<th>DroppedBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Asic Instance = 1
VOQ Base = 48

<table>
<thead>
<tr>
<th>COS0</th>
<th>ReceivedPkts</th>
<th>ReceivedBytes</th>
<th>DroppedPkts</th>
<th>DroppedBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Asic Instance = 2
VOQ Base = 48

<table>
<thead>
<tr>
<th>COS0</th>
<th>ReceivedPkts</th>
<th>ReceivedBytes</th>
<th>DroppedPkts</th>
<th>DroppedBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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).

Note

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.

Note

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)

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/s/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/s/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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>controller optics r/s/i/p</td>
<td>Enters optics controller configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# controller optics 0/3/0/1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>shutdown</td>
<td>Shuts down the optics controller.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-Optics)# shutdown</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-Optics)# commit</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>rx-high-threshold <em>rx-high</em></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></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-Optics)# rx-high-threshold 200</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>tx-high-threshold <em>tx-high</em></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></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-Optics)# tx-high-threshold 300</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>no shutdown</td>
<td>Removes the shutdown configuration on the optics controller.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-Optics)# no shutdown</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

- `commit`

**Example:**

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

**Purpose**

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

### Note

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.

### Note

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`
### Configuring Port Mode Speed

3. shutdown
4. commit
5. port-mode speed \{ 100G | 150G | 200G \} fec \{ 15percent | 25percent \} diff \{ enable | disable \}
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>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: [configure] (\textbf{RP/0/RP0/CPU0:router# configure})</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>controller optics (r/s/i/p)</td>
<td>Enters optics controller configuration mode</td>
</tr>
<tr>
<td></td>
<td>Example: [controller optics (0/3/0/1)]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>shutdown</td>
<td>Shuts down the optics controller.</td>
</tr>
<tr>
<td></td>
<td>Example: [shutdown] (\textbf{RP/0/RP0/CPU0:router(config-Optics)# shutdown})</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td></td>
<td>Example: [commit] (\textbf{RP/0/RP0/CPU0:router(config-Optics)# commit})</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>port-mode speed { 100G</td>
<td>150G</td>
</tr>
<tr>
<td></td>
<td>Example: [port-mode speed 200G fec 15percent diff enable] (\textbf{RP/0/RP0/CPU0:router(config-Optics)# port-mode speed 200G fec 15percent diff enable})</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>no shutdown</td>
<td>Removes the shutdown configuration on the optics controller.</td>
</tr>
<tr>
<td></td>
<td>Example: [no shutdown] (\textbf{RP/0/RP0/CPU0:router(config-Optics)# no shutdown})</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>commit</td>
<td>Saves the configuration changes to the running configuration file.</td>
</tr>
<tr>
<td></td>
<td>Example: [commit] (\textbf{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 ::

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_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR : HI-RXPOWER : DECLARE : Optics0/0/2/0:
PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR : LO-RXPOWER : CLEAR : Optics0/0/2/0:
PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR : HI-RXPOWER : CLEAR : Optics0/0/2/0:
PKT_INFRA-FM-4-FAULT_MINOR : ALARM_MINOR : LO-TXPOWER : CLEAR : 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)# shutdown
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# no port-mode
RP/0/RP0/CPU0:router(config)# port-mode speed 100G fec 15percent diff enable
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# no shutdown
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# 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 port:
  - `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.

**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>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 0/3/0/1 | 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** dwdm-carrier \{100MHz-grid frequency frequency\} \| \{50GHz-grid [ frequency frequency | channel-number ]\}  
   Example:  
   RP/0/RP0/CPU0:router(config-Optics)# dwdm-carrier 100MHz-grid frequency 1960875 | Configures the frequency on the trunk port. |
| **Step 6** no shutdown  
   Example:  
   RP/0/RP0/CPU0:router(config-Optics)# no shutdown | Removes the shutdown configuration on the optics controller. |
### Command or Action

<table>
<thead>
<tr>
<th>Step 7</th>
<th>commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-Optics)# commit</td>
</tr>
</tbody>
</table>

**Purpose**

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/0/RP0/CPU0:router#config
RP/0/RP0/CPU0:router(config)#controller Optics0/3/0/0
RP/0/RP0/CPU0:router(config-Optics)#dwdm-carrier
RP/0/RP0/CPU0:router(config-Optics)#dwdm-carrier 100MHz-grid
RP/0/RP0/CPU0:router(config-Optics)#dwdm-carrier 100MHz-grid frequency
RP/0/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/0/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 gbpsk fec 25sdfec diff disable
dwdm-carrier 100MHz-grid frequency 1960625
```

---

**Note**

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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>controller coherentDSP r/s/i/p</td>
<td>Enters Coherent DSP optics controller configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# controller coherentDSP 0/3/0/1</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>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>RP/0/RP0/CPU0:router(config-CoDSP)# secondary-admin-state maintenance</td>
<td></td>
</tr>
<tr>
<td>Step 4</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-CoDSP)# commit</td>
<td></td>
</tr>
</tbody>
</table>

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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# configure</td>
<td>Enters optics or Coherent DSP controller configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller { optics</td>
<td>coherentDSP } r/s/i/p</td>
<td>Example: RP/0/RP0/CPU0:router(config)# controller coherentDSP 0/3/0/1</td>
</tr>
<tr>
<td><strong>Step 3</strong> pm { 30-sec</td>
<td>15-min</td>
<td>24-hour } { optics</td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session. Example: RP/0/RP0/CPU0:router(config-CoDSP)# commit</td>
<td></td>
</tr>
</tbody>
</table>