

Configuration Guide for Cisco NCS 1014, IOS XR Release 24.4.x

First Published: 2024-12-16

Last Modified: 2026-06-18

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

© 2025 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1 **Cisco NCS 1014 Advanced Multihaul Optical Platform - An Overview** 1

Document objective 1

Cisco NCS 1014 chassis and line cards 1

1.2T line cards 2

2.4T line cards 2

CCMD-16 line cards 2

2.4TX line cards 3

QXP-K9 line cards 5

CHAPTER 2 **Configuring the Card Mode** 7

1.2T line cards 7

Supported data rates on the 1.2T line card 8

Configure split client port mapping 11

Configure the card mode 13

Verify the 1.2T line card configuration 14

Clear alarm statistics 16

Regeneration mode 18

Configure the card in Regen mode 18

Verify the Regen mode 19

Configure the BPS 19

Trunk rates and BPSK modulation 20

Configure trunk rate for BPSK modulation 21

View BPSK trunk rate ranges 21

QXP cards 23

Configure 400G transponder mode 25

Configure 400G muxponder mode 26

Configure 2x100G muxponder mode 27
Cisco 400G QSFP-DD high-power (Bright ZR+) optical module support on QXP card 27
Configure OTN and Ethernet data path on the Bright ZR plus pluggable optical modules 28
2.4T and 2.4TX card modes 30
2.4T and 2.4TX card trunk pluggables and datarates 30
Muxponder slice mode for 2.4T and 2.4TX cards 31
Data rate capabilities for 2.4T and 2.4TX cards in muxponder slice mode 32
Recommended trunk parameters in the 2.4T and 2.4TX cards 33
Customize baud rates 34
Client pluggables for configuring muxponder slice modes 35
Set up the client and trunk rate in the muxponder slice mode for 2.4T and 2.4TX cards 37
Set up 2x100G clients in 800G to 1200G trunk rates in the muxponder slice mode for 2.4TX card 40
Muxponder mode for 2.4TX card 43
Recommended connections for point-to-point topology in muxponder mode 43
Data rate capabilities for the 2.4TX card card 44
Client pluggables for configuring 2.4TX muxponder mode 45
Set up the client and trunk rate in the muxponder mode for the 2.4TX card 48
800GE support in bundle configuration mode on 2.4TA cards 50
Configure 800GE in bundle mode on 2.4TA cards 51

CHAPTER 3 **Configuring Controllers** 55
Automatic-In-Service 56
AINS states 56
Soak time periods 56
Configure Automatic In-Service (AINS) 57
Disable AINS 57
Display AINS configuration 58
FEC 62
Configure FEC on the Ethernet controller 63
Configure FEC on CoherentDSP controllers 65
Verify FEC on CoherentDSP controllers 65
Laser squelching 66
Configure laser squelching 66

Protection switching use cases for Fast-Squelching 68
Hold-off timer 70
Configure holdoff-time on QXP Card 70
Idle Insertion for Ethernet controllers 72
Recommended topology for link verification 72
Configure idle insertion on Ethernet controllers 73
Verify idle insertion configuration for Ethernet controllers 73
FlexO GID and IID 75
Configure FlexO GID and IID 75
Link Layer Discovery Protocol (LLDP) support on management interface 76
Configure LLDP 79
Verify LLDP configuration 80
MAC address snooping on client ports 81
Configure MAC address snooping on client ports 81
View neighbor MAC address 82
Transmit shutdown 83
Configure transmit shutdown on trunk optics controller 83
Verify transmit shutdown on trunk optics controller 83
Loopback 85
Configure loopback 85
Loop-and-drop mechanism on Ethernet controller 92
Restore factory settings 95
Headless mode 97
Trail Trace Identifier 97
Configure Trail Trace Identifier 97
Configure Trail Trace Identifier on QXP card 99
Configure chromatic dispersion 100
Configure chromatic dispersion threshold 102
Configure Transmit Power 104
Configure laser bias current high threshold 105
Configure differential group delay threshold 107
Configure optical signal to noise ratio 109
Configure receive power threshold 111
Frequency 113

Change the trunk port frequency on a transponder card 113
Baud rate 114
CCMD-16 controllers 115
View OCH controller and parameters 115
View OMS controller and parameters 117
Configure Controller Parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L Cards 119
Verify Rx VOA fixed ratio to zero 124
Verify Rx VOA fixed ratio to zero 125
Last link flap detection 127

CHAPTER 4 **Performance Monitoring** 131
Performance monitoring 131
Configure the PM parameters 132
View the PM parameters 133
Clear the PM parameters 148
PM history persistence 149
Instantaneous Q-Margin 150
View Ethernet statistics 151
Performance monitoring for NCS1K14-2.4T-X-K9 card 153
Performance monitoring for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards 153
Configure PM parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards 154
View PM parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards 155

CHAPTER 5 **Pseudo Random Binary Sequence** 157
Pseudo Random Binary Sequence 157
Configure PRBS on CoherentDSP controller 157
Configure PRBS on ODU controller 158
Configure PRBS on Ethernet controller 159
Verify PRBS on controllers 159
Verify the PRBS performance monitor parameters 161
Clear PRBS bit errors and lock time 162
Trunk PRBS 164
Configure the trunk PRBS 164
Client PRBS 166

Configure the client PRBS 166

CHAPTER 6 **Implementing Host Services and Applications 169**

HTTP Client application 169

Configure HTTP Client application 170

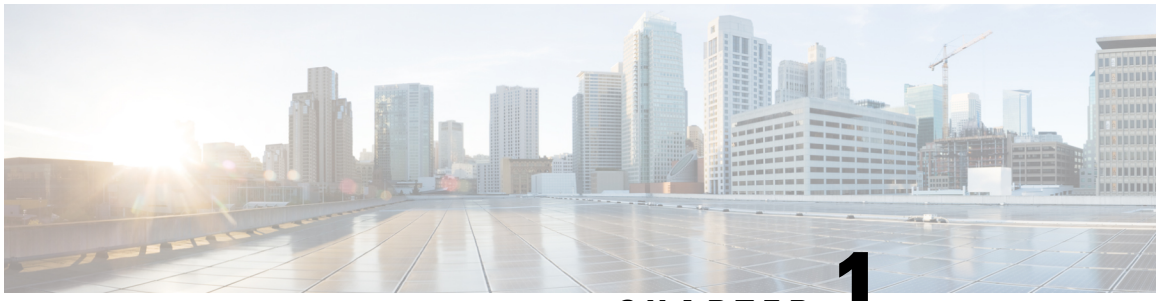
Transmission Control Protocol 171

TCP dump file converter 171

Limitations and restrictions for TCP dump file converter 172

View binary files in text format manually 172

Convert binary files to readable format using TCP dump file converter 173



CHAPTER 1

Cisco NCS 1014 Advanced Multihaul Optical Platform - An Overview

This chapter provides an overview for Cisco NCS 1014 Advanced Multihaul Optical Platform.

- Document objective, on page 1
- Cisco NCS 1014 chassis and line cards, on page 1

Document objective

The Cisco NCS 1014 Configuration Guide describes how to configure various card modes for the line cards that are supported in the Cisco NCS 1014 chassis.

Cisco NCS 1014 chassis and line cards

The Cisco NCS 1014 chassis is an advanced multihaul optical platform that supports transponders and line system cards.

- It is a 2RU chassis,
- delivers a universal transponder solution, and
- provides excellent performance for metro, long-haul, and submarine applications.

Cisco NCS 1014 modules and supported line cards

The NCS 1014 chassis includes these modules:

- Removable controller
- Removable backup solid state drive (SSD)
- Two replaceable power supply units (PSU)
- Three replaceable fan modules
- Four line cards

The NCS 1014 chassis supports these line cards and modules:

Table 1: NCS 1014 supported line cards and modules

From release	PID	Description
7.11.1	NCS1K14-2.4T-K9	2.4T DWDM transponder card
	NCS1K14-CCMD-16-C/L	16-port colorless multiplexer and demultiplexer optical line card.
	NCS1K4-1.2T-K9	1.2T DWDM transponder card
24.1.1	NCS1K14-2.4T-X-K9	2.4T-X DWDM transponder and muxponder card
	NCS1K4-QXP-K9	3.2T QXP transponder card

1.2T line cards

A 1.2T DWDM line card is a transponder that

- enables high-capacity optical transport for Cisco NCS 1014 chassis,
- secures client-side data with AES256-based Layer-1 encryption, and
- delivers flexible client and trunk configurations for 100GE and OTU4 traffic.

Key features

The NCS1K4-1.2T-K9 line card is a single-slot unit.

- It provides 12 client ports for 100GE and OTU4 traffic and two trunks, which can be configured anywhere from 100G to 600G in 50G steps.
- It supports C-band traffic.

2.4T line cards

A 2.4T line card is a coherent optics transponder and muxponder that

- fits a single slot on the Cisco NCS 1014 chassis,
- supports C-band traffic at trunk ports, and
- delivers 400GE, 100GE, and OTU4 client traffic over two trunk ports operating at speeds from 400G to 1.2T each.

CCMD-16 line cards

The CCMD-16 optical line card is a type of optical line card designed for multiplexing and demultiplexing functions in an optical network.

- It features two line ports to transmit and receive using the same LC connectors,

- includes 16 ports for add/drop functionality with LC connector-based interfaces, and
- is available in two variants: NCS1K14-CCMD-16-C (C-band) and NCS1K14-CCMD-16-L (L-band).

CCMD-16 optical line card variants and specifications

The CCMD-16 optical line card is available in two main variants, each supporting different wavelength bands:

- **NCS1K14-CCMD-16-C:** This is a C-band, 16-port Colorless Direct attach optical line card with EDFA. It can host up to 16 channels and supports any signal distribution between 191250 and 196200 GHz, for example, a 64-channel grid with 75-GHz spacing.
- **NCS1K14-CCMD-16-L:** This is an L-band, 16-port Colorless Direct attach optical line card with EDFA. It can host up to 16 channels and supports any signal distribution between 186025 and 191000 GHz, for example, a 64-channel grid with 75-GHz spacing.

2.4TX line cards

The 2.4TX line card (NCS1K14-2.4T-X-K9) is a coherent optics Transponder and Muxponder for the Cisco NCS 1014 chassis.

- It is a single-slot card that supports C-band traffic at trunk ports,
- delivers 400GE and 100GE client traffic over two trunk ports operating at speeds ranging from 400G to 1.2T each, and
- operates in two configurable card modes: Muxponder mode and Muxponder Slice mode.

Table 2: Feature History

Feature Name	Release Information	Feature Description
NCS1K14-2.4T-X-K9 Line Card	Cisco IOS XR Release 24.1.1	The new NCS1K14-2.4T-X-K9 line card is a single-slot Transponder and Muxponder card that delivers up to 1.2T C-band traffic at each trunk port. It has six QDD client ports that support 400GE and 4x100GE traffic on each port. This card provides two trunk ports that support 1.2T traffic on each port. The mxponder-slice and muxponder keywords in the **hw-module** command enable this card to operate in the following modes. • **Muxponder slice mode:** The card virtually splits into two slices in this mode. It is possible to configure both trunks to carry different data rates. The supported trunk rates are 400G, 500G, 600G, 800G, 1000G, and 1200G. • **Muxponder mode:** The card acts as one unit in this mode, configuring both trunks to the same data rate. The supported trunk rates are 600G and 1000G. Additionally, the capacity of a specific shared client port is consumed by two trunk ports.

2.4TX line card features and specifications

The card's operation modes are enabled using the `mxponder-slice` and `muxponder` keywords in the `hw-module` command:

- **Muxponder slice mode:** The card virtually splits into two slices, allowing both trunks to carry different data rates. Supported trunk rates are 400G, 500G, 600G, 800G, 1000G, and 1200G.
- **Muxponder mode:**The card acts as one unit, configuring both trunks to the same data rate. Supported trunk rates are 600G and 1000G. Additionally, the capacity of a specific shared client port is consumed by two trunk ports.

Note In Release 24.1.1, the 2.4TX line card supports only 400G, 500G, 600G, 800G, 1000G, and 1200G trunk payloads.

The 2.4TX card supports various pluggable form factors and data rates for its client and trunk ports, as detailed in this table:

Table 3: Interfaces and data rates

Interfaces	Form factor	Ports	Data rates
Client	QSFP-DD56	1, 2, 3, 4, 5, 6	400G
	QSFP-DD112	2, 5	400, 800G
Trunk	Coherent Interface Module 8 (CIM8)	0, 7	up to 1.2T in each port

Note In Release 24.1.1, ports 2 and 5 support only up to 400G data rates.

The 2.4TX card has two trunk ports and six client ports, as shown in the this figure:

Figure 1: 2.4TX Line card front view

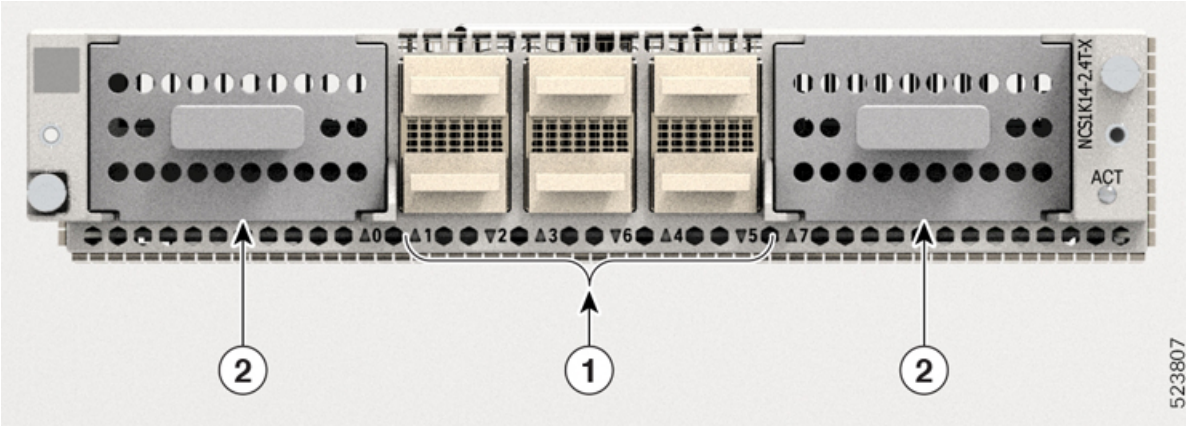


Table 4: 2.4TX line card interfaces

Callout	Interface
1	Client port
2	Trunk port

This figure illustrates the mapping between the client and trunk ports:

Figure 2: 2.4TX card client and trunk mapping

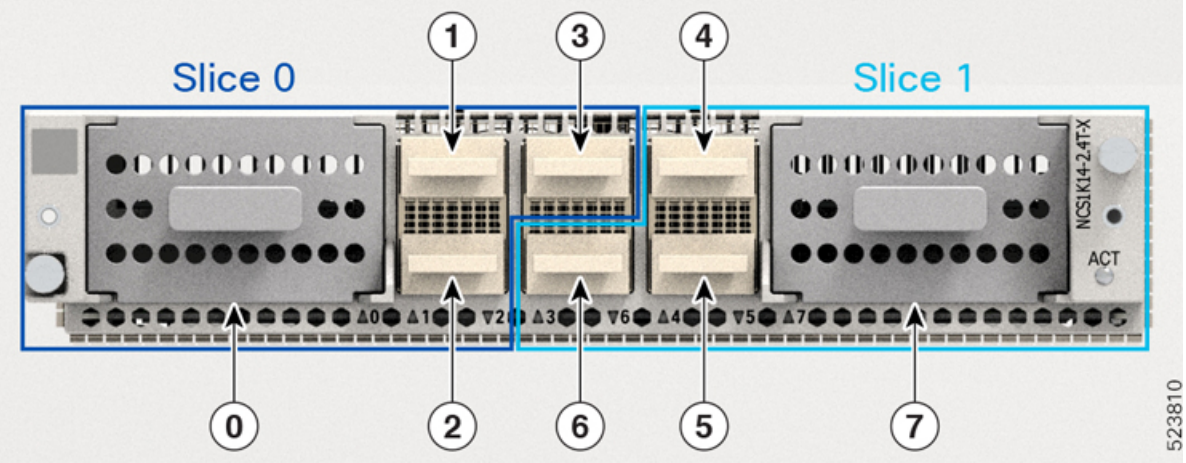


Table 5: Client-to-trunk port mapping

Trunk port	Client ports
0	1, 2, 3
7	4, 5, 6

Note This client-to-trunk port mapping is only applicable for the mxponder-slice mode.

For more information about the 2.4TX card, refer to the datasheet.

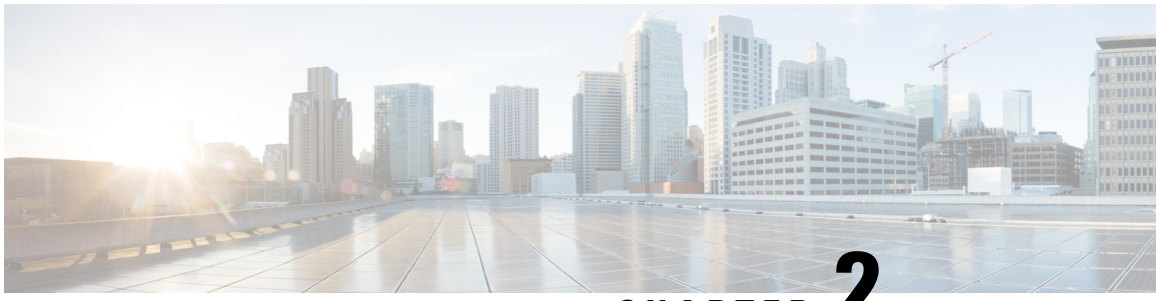
QXP-K9 line cards

A QXP-K9 line card is a high-density QSFP-DD transponder card that

- provides eight client ports (QSFP-DD) and eight trunk ports (QSFP-DD ZR+),
- supports up to 3.2T traffic through its eight QSFP-DD trunk ports, and
- operates each trunk port at speeds up to 400G in 50G increments.

QXP-K9 line card port capabilities

- **Client ports:** Eight QSFP-DD client ports. Each client port supports 400GE, 4x100GE, and 100GE without FEC client rates.
- **Trunk ports:** Eight QSFP-DD ZR+ trunk ports. Each trunk port operates at speeds up to 400G, configurable in 50G increments, contributing to the card's 3.2T total traffic capacity.



CHAPTER 2

Configuring the Card Mode

This chapter lists the supported configurations and the procedures to configure the card mode on the line cards.

Note Unless otherwise specified, “line cards” refers to 1.2T and 1.2TL line cards.

- 1.2T line cards, on page 7
- QXP cards, on page 23
- 2.4T and 2.4TX card modes , on page 30

1.2T line cards

This section describes the supported card modes, card configurations, and procedures to configure the card modes on the 1.2T line card.

A 1.2T line card

- supports both module and slice configurations,
- provides flexible trunk and client port assignment.

Port details:

The line cards are equipped with trunk and client ports:

- Two trunk ports (0 and 1), and
- 12 client ports (2 through 13).

Configuration modes

You can configure the line cards in these two modes:

- **Muxponder mode:** both trunk ports are configured with the same trunk rate. The client-to-trunk mapping is sequential.
- **Muxponder slice mode**: the client-to-trunk mapping is fixed.

This table lists the client ports assigned to each trunk for the 1.2T card in muxponder slice mode.

Table 6: Client-to-trunk mapping for muxponder slice mode

Card	Trunk 0 client ports	Trunk 1 client ports
1.2T	2 to 7	8 to 13

Supported data rates on the 1.2T line card

The 1.2T line card supports these data rates.

This table shows the client and trunk ports that are enabled on these muxponders for the 100GE and OTU4 data rates.

- Muxponder
- Muxponder slice 0
- Muxponder slice 1

Table 7: Data rates for muxponder and muxponder slice 0 and slice 1 mode configuration

Trunk data rate	Client data rate (100GE, OTU4)	Muxponder mode		Muxponder slice mode	
		Trunk ports	Client ports	Client ports for trunk 1	Client ports for trunk 0
100	100GE, OTU4	0	2	8	2
200	100GE, OTU4	0, 1	2, 3, 4, 5	8, 9	2, 3
300	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7	8, 9, 10	2, 3, 4
400	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7, 8, 9	8, 9, 10, 11	2, 3, 4, 5
500	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11	8, 9, 10, 11, 12	2, 3, 4, 5, 6
600	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	8, 9, 10, 11, 12, 13	2, 3, 4, 5, 6, 7

All configurations can be accomplished by using appropriate values for client bitrate and trunk bitrate parameters of the **hw-module** command.

This table shows the trunk parameter ranges for the 1.2T line card.

Table 8: Trunk parameter range for the 1.2T line card

Trunk payload	FEC	Min BPS	Max BPS	Min GBd	Max GBd
50G	15%	1	1.3125	24.0207911	31.5272884
50G	27%	1	1.4453125	24.0207911	34.7175497

Trunk payload	FEC	Min BPS	Max BPS	Min GBd	Max GBd
100G	15%	1	2.625	24.0207911	63.0545768
100G	27%	1	2.890625	24.0207911	69.4350994
150G	15%	1.3203125	3.9375	24.0207911	71.6359689
150G	27%	1.453125	4.3359375	24.0207911	71.6749413
200G	15%	1.7578125	5.25	24.0207911	71.7420962
200G	27%	2	4.40625	31.51	69.43
250G	15%	2.1953125	6	26.2727403	71.8059237
250G	27%	2.4140625	6	28.9312914	71.9068991
300G	15%	2.6328125	6	31.5272884	71.8485385
300G	27%	2.8984375	6	34.7175497	71.8681352
350G	15%	3.0703125	6	36.7818364	71.8790086
350G	27%	3.3828125	6	40.503808	71.8404724
400G	15%	3.5078125	6	42.0363845	71.9018782
400G	27%	3.8671875	6	46.2900663	71.8197392
450G	15%	3.9453125	6	47.2909326	71.9196757
450G	27%	4.34375	6	52.0763245	71.9327648
500G	15%	4.3828125	6	52.5454806	71.93392
500G	27%	4.8281250	6	57.8625828	71.9068991
550G	15%	4.8203125	6	57.8000287	71.9455787
550G	27%	5.3125	6	63.6488411	71.88575
600G	15%	5.2578125	-	-	71.9552971

Trunk Payload	FEC	Min BPS	Max BPS	Min GBd	Max GBd
100G	15%	1	2.625	24.0207911	63.0545768
100G	27%	1	2.890625	24.0207911	69.4350994
150G	15%	1.3203125	3.9375	24.0207911	71.6359689
150G	27%	1.453125	4.3359375	24.0207911	71.6749413
200G	15%	2	4	31.5272884	63.0545768

Trunk Payload	FEC	Min BPS	Max BPS	Min GBd	Max GBd
200G	27%	2	4.40625	31.51664088	69.43509943
250G	15%	2.1953125	4.5	35.0303204	71.8059237
250G	27%	2.4140625	4.5	38.5750552	71.9068991
300G	15%	2.6328125	4.5	42.0363845	71.8485385
300G	27%	2.8984375	4.5	46.2900662857142	71.86813526
350G	15%	3.0703125	4.5	49.0424486	71.8790086
350G	27%	3.3828125	4.5	54.0050773	71.8404724
400G	15%	3.5078125	4.5	56.0485127	71.9018782
400G	27%	3.8671875	4.5	61.72008838	71.81973921

You can configure the sub 50G or coupled mode on the 1.2T line card only in the muxponder mode.

This table shows the port configuration for the supported data rates in the muxponder mode.

Table 9: Supported data rates for muxponder mode

Trunk data rate (per trunk)	Total configured data rate	Trunk ports	Client ports for trunk 0 (100G)	Shared client port (50G per trunk)	Client ports for trunk 1 (100G)
50G	100G	0, 1	-	2	-
150G	300G	0, 1	2	3	4
350G	700G	0, 1	2, 3, 4	5	6, 7, 8
450G	900G	0, 1	2, 3, 4, 5	6	7, 8, 9, 10
550G	1.1T	0, 1	2, 3, 4, 5, 6	7	8, 9, 10, 11, 12

The 1.2T line card supports an alternate port configuration for Sub 50G (split client port mapping) that you can configure using CLI.

This table shows the port configuration for the supported data rates in the split client port mapping mode.

Table 10: Supported data rates for split client port mapping mode

Trunk data rate (per trunk)	Total configured data rate	Trunk ports	Client ports for trunk 0 (100G)	Shared client port (50G per trunk)	Client ports for trunk 1 (100G)
50G	100G	0, 1	-	7	-
150G	300G	0, 1	2	7	8
250G	500G	0, 1	2, 3	7	8, 9

Trunk data rate (per trunk)	Total configured data rate	Trunk ports	Client ports for trunk 0 (100G)	Shared client port (50G per trunk)	Client ports for trunk 1 (100G)
350G	700G	0, 1	2, 3, 4	7	8, 9, 10
450G	900G	0, 1	2, 3, 4, 5	7	8, 9, 10, 11
550G	1.1T	0, 1	2, 3, 4, 5, 6	7	8, 9, 10, 11, 12

Note In all x50G configurations, client traffic on the middle port is affected with ODUK-BDI and LF alarms after the **power cycle or link flap** on the trunk side. This issue is raised when the two network lanes work in coupled mode and move from low to high power. To solve this issue, create a new frame either at the near-end or far-end by performing **shut** or **no shut** of the trunk ports.

Coupled mode restrictions

These restrictions apply to the coupled mode configuration:

- Both trunk ports must be configured with the same bits-per-symbol or baud rate. Both must be sent over the same fiber and direction.
- The chromatic dispersion must be configured to the same value for both trunk ports.
- When trunk internal loopback is configured, you must configure it on both trunk ports. If you configure internal loopback only on one trunk, traffic loss occurs.
- Fault on a trunk port of a coupled pair may cause errors on all clients including those running only on the unaffected trunk port.

Configure split client port mapping

Use this task to configure the trunk port to client port mapping for sub 50G data ratesin the default mode or in the split client port mapping mode.

Procedure

Step 1 Perform any of these steps to configure or remove the split client port mapping mode:

- To configure the trunk port to client port mapping for sub 50G configuration in the split client port mapping mode, run the **split-client-port-mapping** command.

  Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#split-client-port-mapping
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

- To remove the split client port-mapping configuration, run the **no split-client-port-mapping** command.

  Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#no split-client-port-mapping
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

Step 2 Run the **commit** command to apply the changes.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

Step 3 Verify the port mapping using the **show hw-module location** *location* **mxponder** command.

Example:

This example shows how to verify the split client port-mapping configuration.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder
Location:             0/1/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       450G
Status:               Provisioning In Progress
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0   CoherentDSP0/1/0/1
Traffic Split Percentage
HundredGigECtrlr0/1/0/2         ODU40/1/0/0/1                         100                    0
HundredGigECtrlr0/1/0/3         ODU40/1/0/0/2                         100                    0
HundredGigECtrlr0/1/0/4         ODU40/1/0/0/3                         100                    0
HundredGigECtrlr0/1/0/5         ODU40/1/0/0/4                         100                    0
HundredGigECtrlr0/1/0/7         ODU40/1/0/0/5                          50                   50
HundredGigECtrlr0/1/0/8         ODU40/1/0/1/1                           0                  100
HundredGigECtrlr0/1/0/9         ODU40/1/0/1/2                           0                  100
HundredGigECtrlr0/1/0/10        ODU40/1/0/1/3                           0                  100
HundredGigECtrlr0/1/0/11        ODU40/1/0/1/4                           0                  100
```

The split client port mapping is configured.

This is a sample in which split-client-port-mapping is configured with a 450G trunk payload.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#split-client-port-mapping
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

This is a sample in which split-client-port-mapping is removed.

```
RP/0/RP0/CPU0:ios#configur
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#no split-client-port-mapping
```

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

Configure the card mode

Use this task to configure the 1.2T line card in the module (muxponder) or slice configuration (muxponder slice), specifying its client and trunk data rates.

Procedure

Step 1 Configure the client and trunk rates for the muxponder at the specified location.

- **hw-module location** *location* **mxponder client-rate {100GE** | **OTU4}**
- **hw-module location** *location* **mxponder trunk-rate** {**50G** | **100G150G** | **200G** | **250G** | **300G** | **350G** | **400G** | **450G** | **500G** | **550G** | **600G** }

Example:

This is a sample in which the card is configured in the muxponder mode with a 550G trunk payload.

```
RP/0/RP0/CPU0:ios#config
Tue Oct 15 01:24:56.355 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder trunk-rate 550G
RP/0/RP0/CPU0:ios(config)#commit
```

This is a sample in which the card is configured in the muxponder mode with a 500G trunk payload.

```
RP/0/RP0/CPU0:ios#config
Sun Feb 24 14:09:33.989 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder client-rate OTU4
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder trunk-rate 500G
RP/0/RP0/CPU0:ios(config)#commit
```

Step 2 Use these commad to configure the client data rates and trunk data rates of the card in the muxponder slice mode.

- **configure hw-module location** *location* **mxponder-slice** *mxponder-slice-number* **client-rate { 100GE|OTU4}**
- **hw-module location** *location* **mxponder-slice trunk-rate** { **100G** | **200G** | **300G** | **400G** | **500G** | **600G** }

Example:

This is a sample in which the card is configured in the muxponder mode with a 550G trunk payload.

```
RP/0/RP0/CPU0:ios#config
Tue Oct 15 01:24:56.355 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder trunk-rate 550G
RP/0/RP0/CPU0:ios(config)#commit
```

This is a sample in which the card is configured in the muxponder mode with a 500G trunk payload.

```
RP/0/RP0/CPU0:ios#config
Sun Feb 24 14:09:33.989 UTC
```

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder client-rate OTU4
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder trunk-rate 500G
RP/0/RP0/CPU0:ios(config)#commit
```

This is a sample in which the card is configured in the muxponder slice 0 mode with a 500G trunk payload.

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0 client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0 trunk-rate 500G
RP/0/RP0/CPU0:ios(config)#commit
```

This is a sample in which the card is configured in the muxponder slice 1 mode with a 400G trunk payload.

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1 client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1 trunk-rate 400G
RP/0/RP0/CPU0:ios(config)#commit
```

This is a sample in which the card is configured with mixed client rates in the muxponder slice mode.

```
RP/0/RP0/CPU0:ios#configure
Mon Mar 23 06:10:22.227 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0 client-rate OTU4 trunk-rate
500G
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1 client-rate 100GE trunk-rate
 500G
RP/0/RP0/CPU0:ios(config)#commit
```

Verify the 1.2T line card configuration

Use this task to verify the configured settings of the 1.2T line card in either muxponder or muxponder slice mode.

Procedure

Use the **show hw-module location** <*location*> **mxponder** command to verify the card configuration.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0 mxponder
Fri Mar 15 11:48:48.344 IST

Location:             0/2/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
Client Port            Mapper/Trunk Port   CoherentDSP0/2/0/0  CoherentDSP0/2/0/1
                     Traffic Split Percentage

HundredGigECtrlr0/2/0/2  ODU40/2/0/0/1               100                  0
HundredGigECtrlr0/2/0/3  ODU40/2/0/0/2               100                  0
HundredGigECtrlr0/2/0/4  ODU40/2/0/0/3               100                  0
HundredGigECtrlr0/2/0/5  ODU40/2/0/0/4               100                  0
HundredGigECtrlr0/2/0/6  ODU40/2/0/0/5               100                  0
HundredGigECtrlr0/2/0/7  ODU40/2/0/1/1                 0                100
```

```
HundredGigECtrlr0/2/0/8  ODU40/2/0/1/2                  0                  100
HundredGigECtrlr0/2/0/9  ODU40/2/0/1/3                  0                  100
HundredGigECtrlr0/2/0/10 ODU40/2/0/1/4                  0                  100
HundredGigECtrlr0/2/0/11 ODU40/2/0/1/5                  0                  100
```

This is a sample output of the coupled mode configuration where the shared client port is highlighted.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder
Tue Oct 15 01:25:57.358 UTC

Location:             0/1/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       550G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
Client Port           Mapper/Trunk Port    CoherentDSP0/1/0/0 CoherentDSP0/1/0/1
                  Traffic Split Percentage

HundredGigECtrlr0/1/0/2    ODU40/1/0/0/1             100                  0
HundredGigECtrlr0/1/0/3    ODU40/1/0/0/2             100                  0
HundredGigECtrlr0/1/0/4    ODU40/1/0/0/3             100                  0
HundredGigECtrlr0/1/0/5    ODU40/1/0/0/4             100                  0
HundredGigECtrlr0/1/0/6    ODU40/1/0/0/5             100                  0
HundredGigECtrlr0/1/0/7    ODU40/1/0/0/6              50                 50
HundredGigECtrlr0/1/0/8    ODU40/1/0/1/1               0                100
HundredGigECtrlr0/1/0/9    ODU40/1/0/1/2               0                100
HundredGigECtrlr0/1/0/10   ODU40/1/0/1/3               0                100
HundredGigECtrlr0/1/0/11   ODU40/1/0/1/4               0                100
HundredGigECtrlr0/1/0/12   ODU40/1/0/1/5               0                100
```

This is a sample output of all the muxponder slice 0 configurations.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice  0
Fri Mar 15 06:04:18.348 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
Client Port                    Mapper/Trunk Port          CoherentDSP0/1/0/0
                               Traffic Split Percentage

HundredGigECtrlr0/1/0/2        ODU40/1/0/0/1                      100
HundredGigECtrlr0/1/0/3        ODU40/1/0/0/2                      100
HundredGigECtrlr0/1/0/4        ODU40/1/0/0/3                      100
HundredGigECtrlr0/1/0/5        ODU40/1/0/0/4                      100
HundredGigECtrlr0/1/0/6        ODU40/1/0/0/5                      100
```

This is a sample output of all the muxponder slice 1 configurations.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Fri Mar 15 06:11:50.020 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    TRUE
Client Port                    Mapper/Trunk Port          CoherentDSP0/1/0/1
                               Traffic Split Percentage
```

```
HundredGigECtrlr0/1/0/8         ODU40/1/0/1/1                       100
HundredGigECtrlr0/1/0/9         ODU40/1/0/1/2                       100
HundredGigECtrlr0/1/0/10        ODU40/1/0/1/3                       100
HundredGigECtrlr0/1/0/11        ODU40/1/0/1/4                       100
```

This is a sample output of the muxponder slice 1 configuration with client configured as OTU4.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/0/NXR0 mxponder-slice 1

Wed Mar 11 13:59:11.073 UTC

Location:             0/0/NXR0
Slice ID:             1
Client Bitrate:       OTU4
Trunk  Bitrate:       200G
Status:               Provisioned
Client Port                     Peer/Trunk Port            CoherentDSP0/0/0/1
                              Traffic Split Percentage
OTU40/0/0/8                     ODU40/0/0/1/1                       100
OTU40/0/0/9                     ODU40/0/0/1/2                       100
```

This is a sample to verify the mixed client rate configuration in the muxponder slice mode.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder
Mon Mar 23 06:20:22.227 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       OTU4
Trunk  Bitrate:       500G
Status:               Provisioned
Client Port                     Peer/Trunk Port            CoherentDSP0/1/0/0
                                Traffic Split Percentage

OTU40/1/0/2                     ODU40/1/0/0/1                       100
OTU40/1/0/3                     ODU40/1/0/0/2                       100
OTU40/1/0/4                     ODU40/1/0/0/3                       100
OTU40/1/0/5                     ODU40/1/0/0/4                       100
OTU40/1/0/6                     ODU40/1/0/0/5                       100


Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/1
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/8         ODU40/1/0/1/1                         100
HundredGigECtrlr0/1/0/9         ODU40/1/0/1/2                         100
HundredGigECtrlr0/1/0/10        ODU40/1/0/1/3                         100
HundredGigECtrlr0/1/0/11        ODU40/1/0/1/4                         100
HundredGigECtrlr0/1/0/12        ODU40/1/0/1/5                         100
```

Clear alarm statistics

Use this task to clear alarm statistics on the optics or coherent DSP controller.

Procedure

Run the **clear counters controller** *controllertype R/S/I/P* command to clear alarm statistics on the optics or coherent DSP controller.

Example:

This is a sample in which the alarm statistics are cleared on the coherent DSP controller.

```
RP/0/RP0/CPU0:ios#show controller coherentDSP 0/1/0/0
Tue Jun 11 05:15:12.540 UTC

Port                                        : CoherentDSP 0/1/0/0
Controller State                            : Up
Inherited Secondary State                   : Normal
Configured Secondary State                  : Normal
Derived State                               : In Service
Loopback mode                               : None
BER Thresholds                              : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                      : Enable

Alarm Information:
LOS = 1 LOF = 1 LOM = 0
OOF = 1 OOM = 1 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 2      BDI = 2 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0
Detected Alarms                             : None

Bit Error Rate Information
PREFEC  BER                                 : 8.8E-03
POSTFEC BER                                 : 0.0E+00

TTI :
        Remote hostname                     : P2B8
        Remote interface                    : CoherentDSP 0/1/0/0
        Remote IP addr                      : 0.0.0.0

FEC mode                                    : Soft-Decision 15

AINS Soak                                   : None
AINS Timer                                  : 0h, 0m
AINS remaining time                         : 0 seconds
RP/0/RP0/CPU0:ios#clear counters controller coherentDSP 0/1/0/0
Tue Jun 11 05:17:07.271 UTC
All counters are cleared
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/1/0/1
Tue Jun 11 05:20:55.199 UTC

Port                                        : CoherentDSP 0/1/0/1
Controller State                            : Up
Inherited Secondary State                   : Normal
Configured Secondary State                  : Normal
Derived State                               : In Service
Loopback mode                               : None
BER Thresholds                              : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                      : Enable

Alarm Information:
LOS = 0 LOF = 0 LOM = 0
```

```
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0
Detected Alarms                                 : None

Bit Error Rate Information
PREFEC  BER                                     : 1.2E-02
POSTFEC BER                                     : 0.0E+00

TTI :
        Remote hostname                         : P2B8
        Remote interface                        : CoherentDSP 0/1/0/1
        Remote IP addr                          : 0.0.0.0

FEC mode                                        : Soft-Decision 15

AINS Soak                                       : None
AINS Timer                                      : 0h, 0m
AINS remaining time                             : 0 seconds
```

Regeneration mode

In an optical transmission system, 3R regeneration helps extend the reach of the optical communication links by reamplifying, reshaping, and retiming the data pulses.

Regeneration helps to correct any distortion of optical signals by converting it to an electrical signal, processing that electrical signal, and then retransmitting it again as an optical signal.

In Regeneration (Regen) mode, the OTN signal is received on a trunk port and the regenerated OTN signal is sent on the other trunk port of the 1.2T line card and the other way round. In this mode, only the trunk optics controller and coherentDSP controllers are created.

Regeneration can be configured only on the 1.2T line card.

Configure the card in Regen mode

Use this task to configure the regeneration mode on the 1.2T line card. The supported trunk rates are 100G to 600G in multiples of 100G.

Procedure

Step 1 Run the **hw-module location** *location* **regen trunk-rate** command to configure the regeneration mode on the 1.2T card.

Example:

The is a sample to configure the regeneration mode on the 1.2T line card with the trunk-rate 300.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#regen
RP/0/RP0/CPU0:ios(config-regen)#trunk-rate 300
```

Step 2 Commit and exit the configuration.

Example:

```
RP/0/RP0/CPU0:ios(config-regen)#commit
RP/0/RP0/CPU0:ios(config-regen)#exit
```

Verify the Regen mode

Use this task to verify the regeneration mode configuration.

Procedure

Run the **show hw-module location** *location* **regen** command to verify the regen mode.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/0 regen
Mon Mar 25 09:50:42.936 UTC
Location:             0/0/NXR0
Trunk  Bitrate:       400G
Status:               Provisioned
East Port              West Port
CoherentDSP0/0/0/0      CoherentDSP0/0/0/1
```

The terms, **East Port** and **West Port** are used to represent OTN signal regeneration at the same layer.

Configure the BPS

This section provides instructions for configuring the Bits-Per-Symbol (BPS) parameter and viewing BPS and baud rate ranges on supported optical interfaces.

The `bits-per-symbol` parameter allows you to configure the modulation format on optical interfaces. This setting directly affects the spectral efficiency and data rate on a per-wavelength basis.

Supported baud rates are shown in the table.

Table 11: Supported baud rates

Traffic rate	Minimum baud rate	Maximum baud rate
400	43.34518	130.4647
600	59.53435	148.0555
800	79.37913	148.0555
1000	99.22392	148.0555

Use this task to configure BPS value to 3.4375 on the 1.2T line cards to support 300G trunk configurations on 75 GHz networks.

Procedure

Step 1 Run the **controller optics** *R/S/I/P* **bits-per-symbol value** command to configure the BPS to 3.4375 on the 1.2T line cards. This configuration supports 300G trunk configurations on 75 GHz networks.

Example:

This is a sample in which the BPS is configured to 3.4375.

```
RP/0/RP0/CPU0:ios#configure
Wed Mar 27 14:12:49.932 UTC
RP/0/RP0/CPU0:ios(config)#controller optics 0/3/0/0 bits-per-symbol 3.4375
RP/0/RP0/CPU0:ios(config)#commit
```

Step 2 Run these commands to view the BPS and baud for a specific range.

- **show controller optics** *R/S/I/P* **bps-range** *bps-range* **| include** *data-rate* **| include** *fec-type*
- **show controller optics** *R/S/I/P* **baud-rate-range** *baud-range* **| include** *data-rate* **| include** *fec-type*

Example:

```
RP/0/RP0/CPU0:ios#show controllers optics 0/3/0/0 bps-range 3 3.05 | include 300G | include SD27
Thu Mar 28 03:01:39.751 UTC
300G            SD27            3.0000000       69.4350994
300G            SD27            3.0078125       69.2547485
300G            SD27            3.0156250       69.0753320
300G            SD27            3.0234375       68.8968428
300G            SD27            3.0312500       68.7192736
300G            SD27            3.0390625       68.5426174
300G            SD27            3.0468750       68.3668671
```

Example:

```
RP/0/RP0/CPU0:ios#show controllers optics 0/3/0/0 baud-rate-range 43 43.4 | include 300G | include
SD27
Thu Mar 28 03:12:36.521 UTC
300G            SD27            4.8046875       43.3545986
300G            SD27            4.8125000       43.2842178
300G            SD27            4.8203125       43.2140651
300G            SD27            4.8281250       43.1441394
300G            SD27            4.8359375       43.0744397
300G            SD27            4.8437500       43.0049648
```

Trunk rates and BPSK modulation

Trunk rates on the 1.2T line cards can be configured to 50G, 100G, and 150G to support Binary Phase-Shift Keying (BPSK) modulation.

This configuration optimizes the efficiency of carrying information over radio signals.

Supported trunk rates and BPSK modulation

This table lists the trunk rates with the supported BPSK modulation:

Table 12: Trunk rates with the supported BPSK modulation

Trunk rate	BPSK modulation
50G	1 to 1.4453125
100G	1 to 2.890625
150G	1.453125 to 4.3359375

Configure trunk rate for BPSK modulation

Use this task to configure the trunk rate for BPSK modulation.

Procedure

Run the **configurehw-module location** *location* **mxponder trunk-rate {50G | 100G | 150G}**command to configure the trunk rate for BPSK modulation.

Example:

This example shows how to configure trunk rate to 50G:

```
RP/0/RP0/CPU0:(config)#hw-module location 0/0/NXR0 mxponder
RP/0/RP0/CPU0:(config-hwmod-mxp)#trunk-rate 50G
RP/0/RP0/CPU0:(config-hwmod-mxp)#commit
```

View BPSK trunk rate ranges

Determine the trunk rate configured for BPSK modulation on network hardware.

Use this task to verify modulation settings or to troubleshoot device performance. It displays the current trunk rate for BPSK modulation.

Procedure

Run the **show** command to view the trunk rate configured for BPSK modulation.

Example:

```
RP/0/RP0/CPU0:ios(hwmod-mxp)#show hw-module location 0/0/NXR0 mxponder

Tue Feb 25 11:13:41.934 UTC

Location:             0/0/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       50G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                    Mapper/Trunk Port          CoherentDSP0/0/0/0   CoherentDSP0/0/0/1
```

```
                                Traffic Split Percentage

HundredGigECtrlr0/0/0/2         ODU40/0/0/0                          50                      50


RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/0
Thu Mar  5 07:12:55.681 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Green
 Optics Status
         Optics Type:  DWDM optics
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm

         Alarm Status:
         -------------
         Detected Alarms: None


         LOS/LOL/Fault Status:

         Alarm Statistics:

         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 2
         HIGH-TX-PWR = 0            LOW-TX-PWR = 0
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = 1.97 dBm
         RX Power = 1.58 dBm
         RX Signal Power = 0.60 dBm
         Frequency Offset = 386 MHz

         Performance Monitoring: Enable

         THRESHOLD VALUES
         ----------------

         Parameter                High Alarm  Low Alarm  High Warning  Low Warning
         -----------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)          4.9      -12.0           0.0          0.0
         Tx Power Threshold(dBm)          3.5      -10.1           0.0          0.0
         LBC Threshold(mA)                N/A        N/A          0.00         0.00

         Configured Tx Power = 2.00 dBm
         Configured CD High Threshold = 180000 ps/nm
         Configured CD lower Threshold = -180000 ps/nm
         Configured OSNR lower Threshold = 0.00 dB
         Configured DGD Higher Threshold = 180.00 ps
         Baud Rate =  34.7175521851 GBd
         Bits per Symbol = 1.0000000000  bits/symbol
         Modulation Type: BPSK
         Chromatic Dispersion -9 ps/nm
         Configured CD-MIN -180000 ps/nm  CD-MAX 180000 ps/nm
         Polarization Mode Dispersion = 0.0 ps
         Second Order Polarization Mode Dispersion = 125.00 ps^2
```

```
          Optical Signal to Noise Ratio = 34.60 dB
          SNR = 20.30 dB
          Polarization Dependent Loss = 0.20 dB
          Polarization Change Rate = 0.00 rad/s
          Differential Group Delay = 2.00 ps
          Filter Roll Off Factor : 0.100
          Rx VOA Fixed Ratio : 15.00 dB
          Enhanced Colorless Mode : 0
          Enhanced SOP Tolerance Mode : 0
          NLEQ Compensation Mode : 0
          Cross Polarization Gain Mode : 0
          Cross Polarization Weight Mode : 0
          Carrier Phase Recovery Window : 0
          Carrier Phase Recovery Extended Window : 0
AINS Soak                : None
AINS Timer               : 0h, 0m
AINS remaining time      : 0 seconds
```

QXP cards

Table 13: Feature History

Feature Name	Release Information	Description
NCS1K4-QXP-K9 Line Card Support on NCS 1014	Cisco IOS XR Release 24.1.1	NCS1K4-QXP-K9 line card delivers low cost 100G and 400G DWDM transmission with ZR+ optics on a router. This line card can be used in both traditional Optical Networking solution and in Routed Optical Networking solution. This line card has 16 pluggable ports with eight QSFP-DD client ports and eight QSFP-DD trunk ports. For more information about the NCS1K4-QXP-K9 card, see the datasheet.

The NCS1K4-QXP-K9 3.2T QSFP-DD DCO Transponder Line Card has eight client ports (QSFP-DD) and eight trunk ports (QSFP-DD ZR+). Each line card supports up to 3.2 Tbps traffic. The client rates that are supported are 400GE, 4x100GE, and 100GE Ethernet only. The modulation formats supported are 16 QAM for 400GE Txp/4x100GE Mxp.

The QXP line card provides up to 16 QSFP-DD ports (eight QSFP-DD client ports and eight QSFP-DD trunk ports). The supported operating modes are:

- 400GE-TXP
- 4X100GE MXP
- 2x100GE MXP

The QXP card has 8 slices. Each slice consists of one client and one trunk port with a slice capacity of 400G. The total capacity is 3.2T.

Table 14: Slice and port mapping on the QXP card

Slice	Trunk port	Client port
0	0	1
1	2	3
2	4	5
3	6	7
4	8	9
5	10	11
6	12	13
7	14	15

Note

- When you use OPENROADM trunk mode by configuring the **trunk-mode OR** command, use only alternate slices on the QXP card. Either use slices 0, 2, 4, 6 or 1, 3, 5, 7.
- QDD-400G-ZR-S pluggable module supports FEC mode CFEC only.
- QDD-400G-ZR-S pluggable module operates only as an Ethernet transponder.

Supported data rates for QXP card

This table shows the client and trunk ports that are enabled for transponder and muxponder modes.

Operating mode	Card support	Client data rate	Client optics	Trunk ports	Client ports
400GE-TXP	QXP Card	400G	• QDD-400G-DR4-S • QDD-400G-FR4-S • QDD-400-AOCxM	0,2,4,6,8,10,12,14	1,3,5,7,9,11,13,15
4X100GE MXP	QXP Card	4X100G Break out	• QDD-400G-DR4-S • QDD-4X100G-LR-S	0,2,4,6,8,10,12,14	1,3,5,7,9,11,13,15
2X100GE MXP	QXP Card	2X100G Break out	• QDD-400G-DR4-S • QDD-4X100G-LR-S	0,2,4,6,8,10,12,14	1,3,5,7,9,11,13,15

DAC supported modes for NCS1K4-QXP-K9 card

DAC support is enabled on the NCS1K4-QXP-K9 card for 2x100G, 4x100G, and 400G operating modes.

This table provides the details of the respective DAC rates for the different trunk rates for NCS1K4-QXP-K9 card.

Table 15: DAC supported data rates for NCS1K4-QXP-K9 card

Trunk rate	Modulation format	Default value	Modified DAC supported
200G	QPSK	1x1	1x1.50
200G	8QAM	1x1.25	N/A
200G	16-QAM	1x1.25	N/A
400G	16-QAM	1x1	1x1.50

This example changes the DAC rate to 1x1.5 on an optics controller.

```
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/0
RP/0/RP0/CPU0:ios(config-Optics)#dac-Rate 1x1.50
RP/0/RP0/CPU0:ios(config-Optics)#commit
```

Note

- Changing the DAC turns the laser Off and then back on for the optics. This is a traffic impacting operation.
- The DAC rate configuration must match on both ends of a connection.

Configure 400G transponder mode

Use this task to configure and provision the 400G transponder mode.

Procedure

Run the **hw-module location** *location* **mxponder-slice** *slice-number* **trunk-rate 400G trunk-mode [ZR | OR] client-port-rate** *port-number* **client-type 400 GE** to configure and provision 400G TXP.

Example:

This is a sample configuration for setting up a 400G TXP.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 100G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 client-type 100GE
```

This is a sample output of **show hw-module location** *location* **mxponder-slice** *slice-number* when configured in 100G Transponder Mode.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/0 mxponder-slice 0
Sat Jun 25 21:58:15.417 UTC
Location:             0/0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       100G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/0/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/0/0/1                 -                           100
```

Note
The **trunk-mode** command allows you to choose between OTN and ethernet traffic on the trunk port.

Configure 400G muxponder mode

Use this task to configure and provision a 400G MXP.

Procedure

Run the **hw-module location** *location* **mxponder-slice** *slice-number* **trunk-rate 400G client-port-rate***port-number* **lane** *lane-number* **client-type 100GE** to configure and provision 400G MXP.

Example:

The is a sample to configure a 400G MXP.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 4 client-type 100GE
```

This is a sample output of **show hw-module location** *location* **mxponder-slice** *slice-number* when configured in 400G MXP Mode.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/3 mxponder-slice 1
Sat Jun 25 23:03:20.823 UTC
Location: 0/3
Slice ID: 1
Client Bitrate: 100GE
Trunk Bitrate: 400G
Status: Provisioned
LLDP Drop Enabled: FALSE
ARP Snoop Enabled: FALSE
Client Port Mapper/Trunk Port CoherentDSP0/3/0/2
Traffic Split Percentage
HundredGigECtrlr0/3/0/3/1 - 100
```

```
HundredGigECtrlr0/3/0/3/2 - 100
HundredGigECtrlr0/3/0/3/3 - 100
HundredGigECtrlr0/3/0/3/4 - 100
```

Configure 2x100G muxponder mode

Use this task to configure and provision 2x100G MXP.

Procedure

Run the **hw-module location** *location* **mxponder-slice** *slice-number* **trunk-rate 200G client-port-rate***port-number* **lane** *lane-number* **client-type 100GE** to configure 2x100G MXP.

Example:

This is a sample to configure 2x100G MXP.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 200G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 2 client-type 100GE
```

This is a sample output of **show hw-module location** *location* **mxponder-slice** *slice-number* when configured in 2x100G MXP mode.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/3 mxponder-slice 1
Sat Jun 25 23:03:20.823 UTC

Location:             0/3
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       200G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                    Mapper/Trunk Port          CoherentDSP0/3/0/2
                               Traffic Split Percentage

HundredGigECtrlr0/3/0/3/1              -                          100
HundredGigECtrlr0/3/0/3/2              -                          100
```

Cisco 400G QSFP-DD high-power (Bright ZR+) optical module support on QXP card

QXP card supports Cisco 400G QSFP-DD High-Power (Bright) optical modules.

- DP04QSDD-HK9 operates as Ethernet or OTN transponder.

- DP04QSDD-HE0 operate only as an Ethernet transponder.

Table 16: Operating modes supported for Bright ZR+ pluggable modules on QXP card

Operating mode	Modulation	FEC
4x100GE MXP	16-QAM	CFEC
4x100GE MXP	16-QAM	OFEC
2x100GE MXP	QPSK	OFEC
400GE TXP	16-QAM	CFEC
400GE TXP	16-QAM	OFEC

Configure OTN and Ethernet data path on the Bright ZR plus pluggable optical modules

Use this task to configure the OTN and Ethernet data path on Bright ZR+ pluggable optical modules.

Procedure

Step 1 Run the **hw-module location** *location* **mxponder-slice** *slice-number* **trunk-mode OR client-rate***rate* to configure OTN data path on the Bright ZR+ pluggable optical modules. The **trunk-mode OR** refers to OpenROADM.

Example:

This is a sample to configure a 4x100G OTN trunk on a Bright ZR+ pluggable.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 4
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-mode OR
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#  client-port-rate 9 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#  client-port-rate 9 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#  client-port-rate 9 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#  client-port-rate 9 lane 4 client-type 100GE
```

Step 2 Run the **hw-module location** *location* **mxponder-slice** *slice-number* **trunk-mode ZR client-rate** *rate* to configure Ethernet data path on the Bright ZR+ pluggable optical modules.

Example:

This is a sample to configure an Ethernet trunk on a Bright ZR+ pluggable.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 4
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-mode ZR
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
```

This is a sample configuration to set 0dBm transmit power on a Bright ZR+ pluggable.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/2
RP/0/RP0/CPU0:ios(config-Optics)#transmit-power 0
Thu Mar  9 13:02:30.662 UTC
WARNING! Changing TX power can impact traffic
RP/0/RP0/CPU0:ios(config-Optics)#commit
Thu Mar  9 13:02:31.566 UTC
```

The is a sample output of the **show controllers optics** command, with the transmit power set to 0 dBm.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/8
Thu Apr 13 13:54:33.163 UTC
 Controller State: Up
 Transport Admin State: In Service
 Laser State: On
 LED State: Green
 Optics Status
         Optics Type:  QSFP-DD DWDM
         DWDM carrier Info: C BAND, MSA ITU Channel=49, Frequency=193.70THz,
         Wavelength=1547.715nm
         Alarm Status:
         -------------
         Detected Alarms: None
         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 4
         HIGH-TX-PWR = 0            LOW-TX-PWR = 1
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 4
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = 0.00 dBm
         RX Power = -10.50 dBm
         RX Signal Power = -10.35 dBm
         Frequency Offset = 199 MHz

         Performance Monitoring: Enable

         THRESHOLD VALUES
         ----------------

         Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
         ------------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)          3.0      -24.5           0.0          0.0
         Tx Power Threshold(dBm)          0.0      -16.0           0.0          0.0
         LBC Threshold(mA)                N/A        N/A          0.00         0.00

         LBC High Threshold = 90 %
         Configured Tx Power = 0.00 dBm
         Configured CD High Threshold = 52000 ps/nm
         Configured CD lower Threshold = -52000 ps/nm
         Configured OSNR lower Threshold = 21.10 dB
         Configured DGD Higher Threshold = 67.00 ps
```

Note
DP04QSDD-HK9 operates as Ethernet or OTN transponder. DP04QSDD-HE0 operate only as an Ethernet transponder. DP04QSDD-HE0 support only trunk-mode ZR. Configuring trunk-mode OR on the DP04QSDD-HE0 pluggable raises the MEA alarm.

2.4T and 2.4TX card modes

This section helps you familiarize with the different card modes available in the 2.4T and 2.4TX cards , their corresponding data rates, baud rate of each data rate, and the step-by-step procedure to configure line card in muxponder modes with the QDD-4x100GE and QDD-400GE pluggables.

Available card modes

The 2.4T and 2.4TX line cards have two trunk ports (0 and 7) and six client ports (from 1 to 6) each. You can configure the line card in:

- Muxponder slice: You can configure each trunk port independent of the other with different trunk rates. The client-to-trunk mapping is fixed. For Trunk 0, the client ports are 1 to 3. For Trunk 7, the client ports are 4 to 6.
- Muxponder: You can configure both trunk ports with the same trunk rate. The client-to-trunk mapping is fixed.



Note The 2.4T card does not support muxponder mode.

2.4T and 2.4TX card trunk pluggables and datarates

Coherent Interconnect Module 8

The 2.4T and 2.4TX cards support Coherent Interconnect Module 8 (CIM8) pluggables as trunk pluggables.

The Coherent Interconnect Module 8 (CIM8) is a pluggable, high-capacity multi-haul transceiver. The module can operate at line rates between 400G and 1200G in 100G increments. It utilizes a single optical carrier for both C-band and L-band operations.

CIM8-C-K9

CIM8-C-K9 is the C-band Coherent Interconnect module 8.

The frequency range supported on a 50 GHz or 100 MHz flex grid is from 1912500 to 1961000. Any frequency outside this range will trigger a "Port Pluggable Module Mismatched With Pre-Provisioned PPM" alarm, causing the link to go down.

The default frequency is 193.10 THz.

CIM8-CE-K9

CIM8-CE-K9 includes a pre-amplifier (EDFA).

The frequency range supported on a 50 GHz or 100 MHz flex grid is from 1912500 to 1961000. Any frequency outside this range will trigger a "Port Pluggable Module Mismatched With Pre-Provisioned PPM" alarm, causing the link to go down.

Due to the inclusion of the pre-amplifier, the optical performance is enhanced compared to the CIM8-C-K9, enabling longer reach.

CIM8-LE-K9

This variant of the CIM8 supports the L-band spectrum and includes a pre-amplifier (EDFA).

The frequency range supported on a 100 MHz flex grid is from 1861500 to 1909250.Any frequency outside this range triggers a "Port Pluggable Module Mismatched With Pre-Provisioned PPM" alarm, causing the link to go down.

There is no default frequency for the CIM8-LE-K9. You must configure the frequency for the laser to be activated.

In Release 24.3.1 and later releases, if a C-band CIM8 is replaced with an LE CIM8 and the frequency is configured within the specified range, the traffic should resume seamlessly.

From Release 24.4.1, there is no default frequency for any CIM8 pluggables. You must configure the frequency for the laser to be activated.

If data path is configured and default frequency is not configured on a port, then `Mandatory Configuration Missing` alarm is raised on that optics port.

PID	Frequency range supported
CIM8-C-K9	1912500 to 1961000
CIM8-CE-K9	1912500 to 1961000
CIM8-LE-K9	1861500 to 1909250

This table shows the different pluggables and datarates that each pluggable supports.

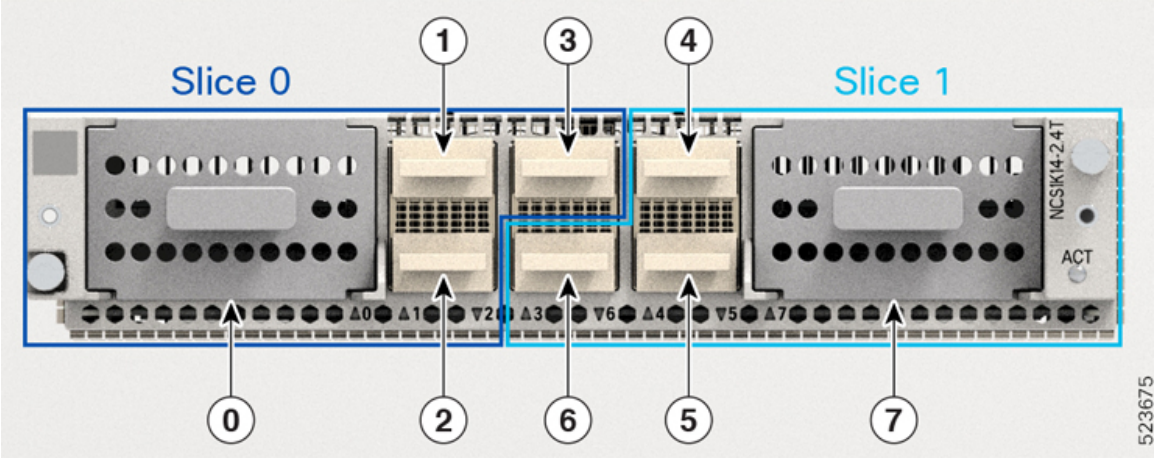
PID	Cards supported	Supported rates
CIM8-C-K9	2.4T and 2.4TX	400G, 500G, 600G, 700G, 800G, 900G, 1000G, 1100G, 1200G
CIM8-CE-K9	2.4TX	400G, 500G, 600G, 700G, 800G, 900G, 1000G, 1100G, 1200G
CIM8-LE-K9	2.4TX	400G, 500G, 600G, 700G, 800G, 900G, 1000G

Muxponder slice mode for 2.4T and 2.4TX cards

This section details the slice and ports supported on the 2.4T and 2.4TX cards.

The line card is divided into two slices, namely, Slice 0 and Slice 1. Each slice contains a trunk port and three client ports. In this mode, the trunk ports operate independently, carrying different data rates. The slices enable the card to function as two different modules. For example, if you set the trunk as 400 G for Slice 0 and 600 G for Slice 1, then Trunk 0 delivers 400 G and Trunk 7 delivers 600 G.

Figure 3: 2.4T line card slices and ports



2.4TX

Figure 4: line card slices and ports

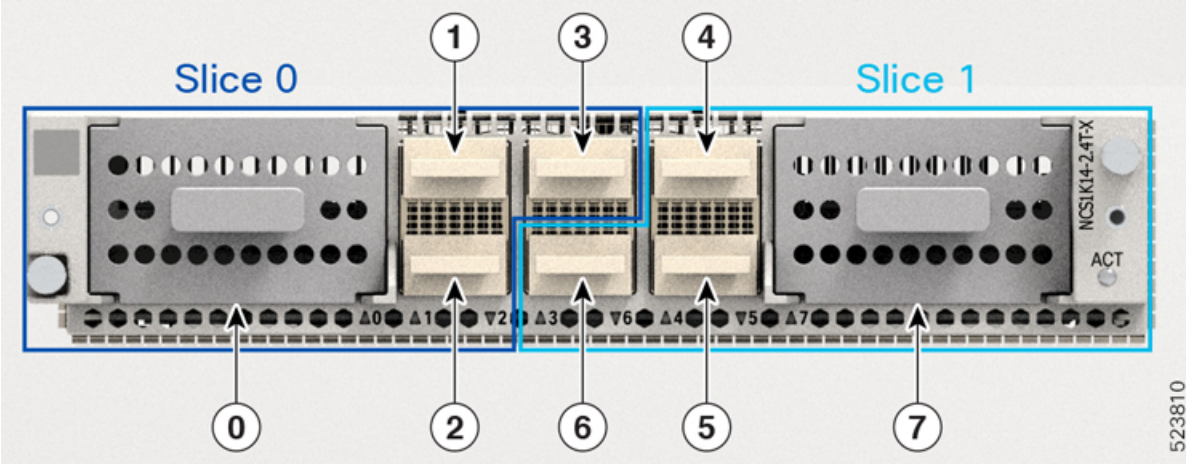


Table 17: Client-to-trunk mapping in slice 0 and slice 1 modes

Slice 0		Slice 1	
Trunk port	**Client ports**	**Trunk port**	**Client ports**
0	1, 2, 3	7	4, 5, 6

Data rate capabilities for 2.4T and 2.4TX cards in muxponder slice mode

The 2.4T and 2.4TX line cards support various trunk rates.

This table shows the releases from which the 2.4T and 2.4TX cards started supporting each trunk rate.

Table 18: Release-wise trunk rates supported by the 2.4T and 2.4TX cards

Trunk rate (G)	2.4T	2.4TX
400	7.11.1	24.1.1
500	—	24.1.1
600	7.11.1	24.1.1
700	-	24.2.1
800	7.11.1	24.1.1
900	-	24.2.1
1000	7.11.1	24.1.1
1100	-	24.2.1
1200	-	24.1.1

Recommended trunk parameters in the 2.4T and 2.4TX cards

This section details the baud rate range for each trunk rate in the 2.4T and 2.4TX cards.

Baud rate ranges for each trunk rate in the 2.4T card

The 2.4T card carries signals at different trunk rates, with each trunk rate operating within a baud rate range.

In the *Baud Rate Ranges for Each Trunk Rate in the 2.4T card* table, you can find the recommended baud rate ranges to maintain the signal health for each trunk rate in the network.

Data rate per trunk (G)	Minimum baud rate (GBd)	Maximum baud rate (GBd)
400	43.34518	130.4647
500	49.61196	147.7235
600	59.53435	148.0555
700	69.45674	147.8182
800	79.37913	148.0555
900	89.30152	147.8709
1000	99.22392	148.0555
1100	109.1463	148.2068
1200	119.0687	148.0555

Data rate per trunk (G)	Minimum baud rate (GBd)	Maximum baud rate (GBd)
400	43.34518	130.4647

Data rate per trunk (G)	Minimum baud rate (GBd)	Maximum baud rate (GBd)
500	49.61196	147.7235
600	59.53435	148.0555
700	69.45674	147.8182
800	79.37913	148.0555
900	89.30152	147.8709
1000	99.22392	148.0555
1100	109.1463	148.2068
1200	119.0687	148.0555

Baud rate and bit rate range for each trunk rate in the 2.4TX card

The 2.4TX card carries trunk signals at different data rates. Each trunk data rate operates in a default baud rate. However, you can customize the baud rate within the recommended baud rate range based on your deployment scenario. To customize baud rate, see.

In the *Baud Rate and Bit Rate Range for Each Trunk Rate in the 2.4TX Card* table, you can find the recommended baud rate ranges to maintain the signal health for each trunk rate in the network. The table also features the bit per second information for the respective baud rates.

Table 19: Baud rate and bit rate range for each trunk rate in the 2.4TX card

Trunk data rate per trunk (G)	Minimum baud r ate (GBd)	Maximum baud rate (GBd)	Default baud rate (GBd)	Minimum bit per second (bps)	Maximum bit per second (bps)
400	43.34518	130.4647	127.931418	2.1	4.1
500	49.61196	147.7235	137.8340588	2.5	5
600	59.53435	148.0555	137.738007	2.8	5.1
700	69.45674	147.8182	138.08166	3.2	5
800	79.37913	148.0555	137.978388	3.5	5.1
900	89.30152	147.8709	137.89817	3.8	5.2
1000	99.22392	148.0555	137.834059	4.3	5.3
1100	109.1463	148.2068	137.78165	4.7	5.3
1200	119.0687	148.0555	137.738007	5.3	5.7

Customize baud rates

Customizing baud rates is to enable users to adjust the default baud rates for the 2.4T and 2.4TX cards, particularly when operating in muxponder mode. This customization allows for optimization of network

bandwidth based on specific deployment scenarios and ensures that the baud rates align with the available bandwidth in the network.

The muxponder mode enables the 2.4T and 2.4TX cards to carry signals in default baud rates when you set up the trunk rate. However, you can customize the baud rates for each trunk rate based on the bandwidth in the network.

Use this task to customize the baud rates within the recommended range as per your deployment scenario.

Before you begin

- Install these pluggable modules as required.
  - QDD-4x100G
  - QDD-400G
- Enter the Cisco IOS XR configuration mode.

Procedure

Step 1 Locate the trunk optics controller for the 2.4T and 2.4TX cards.

Example:

```
RP/0/RP0/CPU0:ios(config)# controller optics 0/0/0/7
```

Step 2 Enter baud rate.

Example:

```
RP/0/RP0/CPU0:ios(config-Optics)# baud-rate 120.0000
```

Step 3 Save the changes.

Example:

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

Client pluggables for configuring muxponder slice modes

This section provides details about the client pluggable combinations that you need to set up the client rate for each trunk rate in slice 0 and slice 1.

Pluggable combinations in muxponder slice modes

The client data rates and ports differ for each trunk rate in the muxponder slice 0 (Trunk 0) and muxponder slice 1 (Trunk 1) configurations. However, the type of client pluggable modules stays the same for both slice modes.

Note The 2x 100 GE client rate is supported only on 2.TX cards.

Table 20: Trunk rate and client pluggable combinations for Slices 0 and Slice 1

Trunk rate (G) per trunk	Card support	Client rate	Client pluggable	Client ports	
				Slice 0	Slice 1
400	2.4T, 2.4TX	2x 100 GE 400 GE	QDD-400G	1	4
		2x 100 GE 4x 100 GE	QDD-4x100G		
500	2.4T, 2.4TX	2x 100 GE 400 GE + 1x 100 GE	QDD-400G + QDD-4x100G	1, 2	4, 5
		5x 100 GE	2x QDD-4x100G		
600	2.4T, 2.4TX	400 GE + 2x 100 GE	QDD-400G + QDD-4x100G	1, 2	4, 5
		6x 100 GE	2x QDD-4x100G		
700	2.4T, 2.4TX	2x 100 GE 400 GE + 3x 100 GE	QDD-400G + QDD-4x100G	1, 2	4, 5
		7x 100 GE	2x QDD-4x100G		
800	2.4T, 2.4TX	2x 100 GE 2x 400 GE	2x QDD-400G	1, 2	4, 5
		400 GE + 4x 100 GE	QDD-400G + QDD-4x100G		
		8x 100 GE	2x QDD-4x100G		
900	2.4T, 2.4TX	2x 100 GE 2x 400 GE + 1x 100 GE	QDD-400G + QDD-4x100G	1, 2, 3	4, 5, 6
		400 GE + 5x 100 GE	QDD-400G + QDD-4x100G		
		9x 100 GE	3x QDD-4x100G		
1000	2.4T, 2.4TX	2x 100 GE 2x 400GE + 2x 100 GE	2x QDD-400G + 2x QDD-4x100G	1, 2, 3	4, 5, 6
		10x 100 GE	3x QDD-4x100G		

Trunk rate (G) per trunk	Card support	Client rate	Client pluggable	Client ports	
				Slice 0	Slice 1
1100	2.4TX	2x 100 GE 2x 400 GE + 3x 100 GE	2x QDD-400G + QDD-4x100G	1, 2, 3	4, 5, 6
		400 GE + 7x 100 GE	2x QDD-400G + QDD-4x100G		
		11x 100 GE	3x QDD-4x100G		
1200	2.4TX	3x 400 GE	3x QDD-400G	1, 2, 3	4, 5,6
		2x 400 GE + 4x 100 GE	2x QDD-400G + QDD-4x100G		
		400 GE + 8x 100 GE	QDD-400G + 2x QDD-4x100G		
		12x 100 GE	3x QDD-4x100G		
		6x 2X100 GE	6x QDD-2X100-CWDM4-S	1, 2, 3, 4, 5, 6	—
			6x QDD-2X100-LR4-S		

[1] QDD-400G refers to QDD-400G-FR4-S, QDD-400G-LR4-S, QDD-400G-AOCxM, and QDD-400G-DR4-S pluggable modules.
[2] QDD-4x100G refers to QDD-4X100G-LR-S, QDD-4X100G-FR-S, and QDD-400G-DR4-S pluggable modules.

Make sure you use the appropriate values for client bitrate and trunk bitrate parameters when configuring the Muxponder slide mode using the **hw-module** command.

Set up the client and trunk rate in the muxponder slice mode for 2.4T and 2.4TX cards

Set up the client and trunk rate in the muxponder slice mode for the 2.4T and 2.4TX cards to configure these cards to handle specific data rates for both the aggregated trunk and individual client ports.

Use this task to set up the client and trunk rate in the muxponder slice mode for the 2.4T and 2.4TX cards.

This task considers that you are setting up the 600-G data rate in one of the trunk ports of the 2.4T and 2.4TX cards. This scenario requires you to set the client rate for the client ports. Based on the client pluggable that you use, the client rate can change to 400-GE client, 100-GE client, or mixed client.

For more information on the the data rate on each client port, see Client pluggables for configuring muxponder slice modes, on page 35 .

Before you begin

- Install these pluggables as required.
  - QDD-400G
  - QDD-4x100G

Procedure

Step 1 Specify the card location.

Example:

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
```

Step 2 Configure the 2.4T or 2.4TX line card in the muxponder slice mode.

For Trunk 0 port, enter the `muxponder-slice 0` mode.

Example:

```
RP/0/RP0/CPU0:ios(config)# mxponder-slice 0
```

For Trunk 1 port, enter the `muxponder-slice 1` mode.

Example:

```
RP/0/RP0/CPU0:ios(config)# mxponder-slice 1
```

Note
You can configure both muxponder slice 0 and slice 1 modes when needed.

For more information on how to configure muxponder slice mode with QDD-4x100GE and QDD-4x100GE pluggables, see the hw-module command.

Step 3 Set up the trunk rate for the 2.4T or 2.4TX card.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
```

Step 4 Set up the client rate based on the pluggables that you use.

For the QDD-400G pluggable, run this command.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 2 client-type 400GE
```

For the QDD-4x100G pluggable, run this command.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 1 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 1 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 1 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 1 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 2 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 2 lane 2 client-type 100GE
```

Note
Use the **lane** keyword to set up the 100-GE client rate in the client ports.

For the mixed client pluggable, use the combination of the QDD-400G and QDD-4x100G commands.

Step 5 Save the configuration and exit the muxponder slice mode.

Example:

Command

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
exit
RP/0/RP0/CPU0:ios(config)#
exit
```

Step 6 Verify the 600-G data rate that you set up.

This sample shows the 600-G data rate (Trunk Bitrate: 600G) set up in client ports 1 (FourHundredGigECtrlr0/1/0/1) and 2 with breakout lanes 1 and 2 (HundredGigECtrlr0/1/0/2/1 and HundredGigECtrlr0/1/0/2/2) using 400-GE and 100-GE client type pluggables (Client Bitrate: MIXED) in muxponder slice 0 (Slice ID: 0).

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 15:41:25.720 UTC
Location:             0/1/NXR0
Slice ID: 0 Client Bitrate: MIXED Trunk Bitrate: 600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
Traffic Split Percentage
FourHundredGigECtrlr0/1/0/1 ODU-FLEX0/1/0/0/1 100
   HundredGigECtrlr0/1/0/2/1 ODU-FLEX0/1/0/0/2/1 100
   HundredGigECtrlr0/1/0/2/2 ODU-FLEX0/1/0/0/2/2 100
```

This sample shows the 600-G data rate (Trunk Bitrate: 600G) set up in client ports 0 with breakout lanes 1 to 4 (HundredGigECtrlr0/1/0/1/1 to HundredGigECtrlr0/1/0/1/4) and 1 (HundredGigECtrlr0/1/0/2/1) using 100-GE client type pluggable (Client Bitrate: 100GE) in muxponder slice 0 (Slice ID: 0).

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 16:06:57.575 UTC
Location:             0/1/NXR0
Slice ID: 0 Client Bitrate: 100GE Trunk Bitrate: 600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port           Mapper/Trunk Port          CoherentDSP0/1/0/0
Traffic Split Percentage
HundredGigECtrlr0/1/0/1/1 ODU-FLEX0/1/0/0/1/1 100
   HundredGigECtrlr0/1/0/1/2 ODU-FLEX0/1/0/0/1/2 100
   HundredGigECtrlr0/1/0/1/3 ODU-FLEX0/1/0/0/1/3 100
   HundredGigECtrlr0/1/0/1/4 ODU-FLEX0/1/0/0/1/4 100
   HundredGigECtrlr0/1/0/2/1 ODU-FLEX0/1/0/0/2/1 100
   HundredGigECtrlr0/1/0/2/2 ODU-FLEX0/1/0/0/2/2 100
```

Set up 2x100G clients in 800G to 1200G trunk rates in the muxponder slice mode for 2.4TX card

Use this task to set up 2x100G client pluggables in 800G to 1200G trunk rates in the muxponder slice mode for the 2.4TX card.

For more information on the data rate on each client port, see Client pluggables for configuring muxponder slice modes, on page 35.

Before you begin

- Install either of these pluggables in all 6 client ports.
  - QDD-2X100-CWDM4-S
  - QDD-2X100-LR4-S

Procedure

Step 1 Specify the card location.

Example:

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
```

Step 2 Configure the 2.4TX line cards in the muxponder slice mode.

For 6x2x100pluggables in 800G to 1200G trunk modes all client ports are in slice 0. Enter the `muxponder-slice 0` mode.

Example:

```
RP/0/RP0/CPU0:ios(config)#mxponder-slice 0
```

Step 3 Set up the trunk rate for the 2.4TX cards.

These examples show how to set up various trunk rates in the muxponder slice mode.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1200G
```

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1100G
```

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
```

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 900G
```

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
```

Step 4 Set up the client rate.

For the 2X100G pluggables, run this command.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-rate 100GE
```

Step 5 Save the configuration and exit the muxponder slice mode.

Example:

Command

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#exit
RP/0/RP0/CPU0:ios(config)#exit
```

Step 6 Verify the data rate that you set up.

This sample shows the 800G data rate set up in the client ports 1, 2, 4, and 5.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Mon Aug 18 02:11:25.625 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1                          100
HundredGigECtrlr0/1/0/1/5       ODU-FLEX0/1/0/0/2                          100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/3                          100
HundredGigECtrlr0/1/0/2/5       ODU-FLEX0/1/0/0/4                          100
HundredGigECtrlr0/1/0/4/1       ODU-FLEX0/1/0/0/5                          100
HundredGigECtrlr0/1/0/4/5       ODU-FLEX0/1/0/0/6                          100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/0/7                          100
HundredGigECtrlr0/1/0/5/5       ODU-FLEX0/1/0/0/8                          100
```

This sample shows the 900G data rate set up in the client ports 1, 2, 3, 4, and 5.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1 mxponder-slice 0
Mon Aug 18 02:15:28.412 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       900G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1                          100
HundredGigECtrlr0/1/0/1/5       ODU-FLEX0/1/0/0/2                          100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/3                          100
HundredGigECtrlr0/1/0/2/5       ODU-FLEX0/1/0/0/4                          100
HundredGigECtrlr0/1/0/3/1       ODU-FLEX0/1/0/0/5                          100
HundredGigECtrlr0/1/0/4/1       ODU-FLEX0/1/0/0/6                          100
```

```
HundredGigECtrlr0/1/0/4/5       ODU-FLEX0/1/0/0/7                           100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/0/8                           100
HundredGigECtrlr0/1/0/5/5       ODU-FLEX0/1/0/0/9                           100
```

This sample shows the 1000G data rate set up in up in the client ports 1, 2, 3, 4, and 5.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1 mxponder-slice 0
Mon Aug 18 02:15:48.811 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       1000G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1                           100
HundredGigECtrlr0/1/0/1/5       ODU-FLEX0/1/0/0/2                           100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/3                           100
HundredGigECtrlr0/1/0/2/5       ODU-FLEX0/1/0/0/4                           100
HundredGigECtrlr0/1/0/3/1       ODU-FLEX0/1/0/0/5                           100
HundredGigECtrlr0/1/0/3/5       ODU-FLEX0/1/0/0/6                           100
HundredGigECtrlr0/1/0/4/1       ODU-FLEX0/1/0/0/7                           100
HundredGigECtrlr0/1/0/4/5       ODU-FLEX0/1/0/0/8                           100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/0/9                           100
HundredGigECtrlr0/1/0/5/5       ODU-FLEX0/1/0/0/10                          100
```

This sample shows the 1100G data rate set up in 1, 2, 3, 4, 5, and 6 client ports.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Mon Aug 18 02:12:03.291 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       1100G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1                           100
HundredGigECtrlr0/1/0/1/5       ODU-FLEX0/1/0/0/2                           100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/3                           100
HundredGigECtrlr0/1/0/2/5       ODU-FLEX0/1/0/0/4                           100
HundredGigECtrlr0/1/0/3/1       ODU-FLEX0/1/0/0/5                           100
HundredGigECtrlr0/1/0/3/5       ODU-FLEX0/1/0/0/6                           100
HundredGigECtrlr0/1/0/4/1       ODU-FLEX0/1/0/0/7                           100
HundredGigECtrlr0/1/0/4/5       ODU-FLEX0/1/0/0/8                           100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/0/9                           100
HundredGigECtrlr0/1/0/5/5       ODU-FLEX0/1/0/0/10                          100
HundredGigECtrlr0/1/0/6/1       ODU-FLEX0/1/0/0/11                          100
```

This sample shows the 1200G data rate set up in all 12 client ports.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0 mxponder-slice 0
Thu Nov 16 15:41:25.720 UTC
Location:             0/2/NXR0
Slice ID:             0
Client Bit100GE
Trunk  Bitrate:       1200G
Status:               Provisioned
rate:       LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                       Mapper/Trunk Port          CoherentDSP0/2/0/0
                                  Traffic Split Percentage

HundredGigECtrlr0/2/0/1/1         ODU-FLEX0/2/0/0/1                             100
HundredGigECtrlr0/2/0/1/5         ODU-FLEX0/2/0/0/2                             100
HundredGigECtrlr0/2/0/2/1         ODU-FLEX0/2/0/0/3                             100
HundredGigECtrlr0/2/0/2/5         ODU-FLEX0/2/0/0/4                             100
HundredGigECtrlr0/2/0/3/1         ODU-FLEX0/2/0/0/5                             100
HundredGigECtrlr0/2/0/3/5         ODU-FLEX0/2/0/0/6                             100
HundredGigECtrlr0/2/0/4/1         ODU-FLEX0/2/0/0/7                             100
HundredGigECtrlr0/2/0/4/5         ODU-FLEX0/2/0/0/8                             100
HundredGigECtrlr0/2/0/5/1         ODU-FLEX0/2/0/0/9                             100
HundredGigECtrlr0/2/0/5/5         ODU-FLEX0/2/0/0/10                            100
HundredGigECtrlr0/2/0/6/1         ODU-FLEX0/2/0/0/11                             100
HundredGigECtrlr0/2/0/6/5         ODU-FLEX0/2/0/0/12                            100
```

Muxponder mode for 2.4TX card

The muxponder mode enables the 2.4TX card to split wavelengths in specific client ports between the two trunk ports. In the slice mode, the client ports that support wavelength splitting act the same as other client ports. However, in the muxponder mode, the 2.4TX card activates the split client ports. The shared client ports are client port 2 for 600G and client port 3 for 1000G.

How muxponder mode splits 400GE and 4x100GE client traffic

This use case explains the wavelength splitting for 600G trunk rate.

For 600G trunk rate, you must configure client port 1, 2, and 4 as 400GE or 4x100GE. Trunk 0 receives 400GE from port 1. Trunk 7 receives 400GE from port 4. As per split client configuration, port 2 gives 200GE to Trunk 0 and another 200GE to Trunk 7. In this way, both trunk ports deliver 600G trunk rate each.

Recommended connections for point-to-point topology in muxponder mode

- Connect the port 0 and port 7 in the near end node to their respective port 0 and port 7 in the far end node.
- Make sure the optic fibers connected to trunk ports 0 and 7 are the same length. The difference must be less than 500 m; otherwise, you'll lose traffic on the split port.

Data rate capabilities for the 2.4TX card card

Table 21: Feature History

Feature Name	Release Information	Description
Additional Muxponder Mode Trunk Rates for the NCS1K14-2.4T-X-K9 Line Card	Cisco IOS XR Release 24.3.1	The NCS1K14-2.4T-X-K9 line card now supports additional trunk rates of 500G and 900G in muxponder mode, enhancing flexibility and optimizing pluggable count alongside the existing 600G and 1000G rates.

To outline the data rate capabilities of the 2.4TX card card.

The 2.4TX card card supports different trunk rates.

Table 22: Release-wise trunk rates supported by the 2.4TX card cards

Trunk rate (G)	Release introduced
500	24.3.1
600	24.1.1
900G	24.3.1
1000G	24.1.1



Note For 600G and 1000G trunk rates, in R24.1.1, the shared client port supports only 400GE client and from R24.3.1, the shared client port supports both 400GE and 4x100GE clients.

Client pluggables for configuring 2.4TX muxponder mode

Table 23: Feature History

Feature Name	Release Information	Description
100GE Channel Support for the 600G and 1000G Trunk Rate in NCS1K14-2.4T-X-K9 Muxponder Mode	Cisco IOS XR Release 24.3.1	The NCS1K14-2.4T-X-K9 line card now allows 100G breakout client support for 600G and 1000G trunk rate in muxponder mode. It features 4x100GE breakout channels in shared client ports, enabling easy integration with existing 100G networks using QDD-4X100G-LR-S, QDD-4X100G-FR-S, and QDD-400G-DR4-S pluggable modules. These channels offer high density and bandwidth efficiency without extra costs.

This section provides details about the client pluggable combinations that you need to set up the client rate for each trunk rate.

Client pluggable combinations in muxponder mode

The 2.4TX card muxponder mode supports various trunk rate per trunk with different client pluggable combinations.

Note From R24.3.1, the 2.4TX card supports 100GE client traffic in the shared client port for both 600G and 1000G trunk rates.

The client channel rate in the table refers to both the total client rate and the client rate per channel in the client ports. For example, **2x 400GE + 2x 100GE** indicates that the client traffic consists of two channels at 400GE each and two channels at 100GE each.

Table 24: 2.4TX card muxponder mode port configurations

Trunk rate (G) per trunk	Total configured trunk rate (G)	Client channel rate	Client pluggable	Shared client port	Client ports
500	1000	2x 400GE + 2x 100GE	2x QDD-400G + 1x QDD-4x100G	2	1, 4
		1x 400GE + 6x 100GE	1x QDD-400G + 2x QDD-4x100G		
		10x 100GE	3x QDD-4x100G		

Trunk rate (G) per trunk	Total configured trunk rate (G)	Client channel rate	Client pluggable	Shared client port	Client ports
600	1200	3x 400GE	3x QDD-400G	2	1, 4
		2x 400GE + 4x 100GE	2x QDD-400G + 1x QDD-4x100G		
		1x 400GE + 8x 100GE	1x QDD-400G + 2x QDD-4x100G		
900	1800	4x 400GE + 2x 100GE	4x QDD-400G + 1x QDD-4x100G	3	1, 2, 4, 5
		3x 400GE + 6x 100GE	3x QDD-400G + 2x QDD-4x100G		
		2x 400GE + 10x 100GE	2x QDD-400G + 3x QDD-4x100G		
		1x 400GE + 14x 100GE	1x QDD-400G + 4x QDD-4x100G		
		18x 100GE	5x QDD-4x100G		
1000	2000	5x 400GE	5x QDD-400G	3	1, 2, 4, 5
		4x 400GE + 4x 100GE	4x QDD-400G + 1x QDD-4x100G		
		3x 400GE + 8x 100GE	3x QDD-400G + 2x QDD-4x100G		
		2x 400GE + 12x 100GE	2x QDD-400G + 3x QDD-4x100G		
		1x 400GE + 16x 100GE	1x QDD-400G + 4x QDD-4x100G		

Understanding client rates per client port for each trunk rate

This table shows the sample client rate per client port for each trunk rate. This simplified matrix helps you understand the traffic flow in each client port. It also indicates the number of channels that each client port

uses to deliver the client traffic. The type of pluggable module inserted in the shared client port determines the traffic rate through breakout and non-breakout channels.

You can customize the configuration by mixing and matching the client pluggable modules according to your requirements.

Table 25: Client rate traffic per trunk rate and client pluggable combinations

Trunk rate (G) per trunk	Client pluggable	Client rate (GE) per trunk 0 client ports		Client rate (GE) per shared client ports		Client rate (GE) per trunk 1 client ports		
		1	2	2	3	4	5	6
500	2x QDD-400G + 1x QDD-4x100G	400	-	2x 100	-	400	-	-
	1x QDD-400G + 2x QDD-4x100G	400	-	2x 100[3]	-	4x 100	-	-
	3x QDD-4x100G	4x 100	-	2x 100[3]	-	4x 100	-	-
600	3x QDD-400G	400	-	400	-	400	-	-
	2x QDD-400G + 1x QDD-4x100G	400	-	4x 100[3]	-	400	-	-
	1x QDD-400G + 2x QDD-4x100G + 3x QDD-4x100G	400	-	4x 100[3]	-	4x 100	-	-
900	4x QDD-400G + 1x QDD-4x100G	400	400	-	2x 100[3]	400	400	-
	3x QDD-400G + 2x QDD-4x100G	400	400	-	2x 100[3]	400	4x 100	-
	2x QDD-400G + 3x QDD-4x100G	400	400	-	2x 100[3]	4x 100	4x 100	-
	1x QDD-400G + 4x QDD-4x100G	400	4x 100	-	2x 100[3]	4x 100	4x 100	-
	5x QDD-4x100G	4x 100	4x 100	-	2x 100[3]	4x 100	4x 100	-

Trunk rate (G) per trunk	Client pluggable	Client rate (GE) per trunk 0 client ports		Client rate (GE) per shared client ports		Client rate (GE) per trunk 1 client ports		
		1	2	2	3	4	5	6
1000	5x QDD-400G	400	400	-	400	400	400	-
	4x QDD-400G + 1x QDD-4x100G	400	400	-	4x 100	400	400	-
	3x QDD-400G + 2x QDD-4x100G	400	400	-	4x 100	400	4x 100	-
	2x QDD-400G + 3x QDD-4x100G	400	400	-	4x 100	4x 100	4x 100	-
	1x QDD-400G + 4x QDD-4x100G + 5x QDD-4x100G	400	4x 100	-	4x 100	4x 100	4x 100	-

[3] In this shared port, the pluggable capacity is 400GE or 4x 100GE, but, for this trunk rate, the 2.4TX card card consumes only 2x 100GE client data.

Set up the client and trunk rate in the muxponder mode for the 2.4TX card

Use this task to configure a trunk rate in muxponder mode for the 2.4TX card card.

Note This task considers that you're setting up the 600G trunk rate in the muxponder mode for the 2.4TX card card. The commands and output shown are for 600G trunk rate. The commands and output change for other trunk rates.

This task uses a mix of client pluggable modules. For this task, the card has:

- QDD-4x100G pluggable in shared client port 2, and
- QDD-400G pluggable in client ports 1 and 4

Note For the 600G trunk rate, the split port supports both 400GE and 4x100GE. For more information on required pluggable modules for other trunk rates, see Client pluggables for configuring 2.4TX muxponder mode, on page 45 .

Before you begin

- Install the pluggables as required.
  - QDD-400G
  - QDD-4x100G

Procedure

Step 1 Specify the card location.

Example:

```
RP/0/RP0/CPU0:ios(config)# hw-module location 0/1/NXR0
```

Step 2 Enter the muxponder card mode.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder
```

Step 3 Set up the trunk rate.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)# trunk-rate 600G
```

Step 4 Set up the client rate for the QDD-400G and QDD-4x100G pluggable modules.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 1 client-type 400GE
// QDD-400G pluggable in client port 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
client-port-rate 2 lane 1 client-type 100GE
// Enter lane for the QDD-4x100G pluggable in client port 2
 RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
 client-port-rate 2 lane 2 client-type 100GE
// Enter lane for the QDD-4x100G pluggable in client port 2
  RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
 client-port-rate 2 lane 3 client-type 100GE
// Enter lane for the QDD-4x100G pluggable in client port 2
 RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
 client-port-rate 2 lane 4 client-type 100GE
// Enter lane for the QDD-4x100G pluggable in client port 2
 RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
 client-port-rate 4 client-type 400GE
```

Note
Use the **lane** keyword to set up the 100GE client rate in the client ports.

Step 5 Save the configuration and exit the muxponder mode.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
exit
// Exits muxponder mode
RP/0/RP0/CPU0:ios(config)#
exit
// Exits configuration moder
```

Step 6 Verify the 600G mixed client rate configured for the 2.4TX card muxponder mode.

This sample shows the 600G data rate (Trunk Bitrate: 600G) set up in client ports 1 and 4 (FourHundredGigECtrlr0/2/0/1 and FourHundredGigECtrlr0/2/0/4) and split client port 2 with breakout lanes 1 to 4 (HundredGigECtrlr0/2/0/2/1 to HundredGigECtrlr0/2/0/2/4).

Example:

```
RP/0/RP0/CPU0:ios# show hw-module location 0/2/NXR0 mxponder
Location:             0/2/NXR0
Client Bitrate: MIXED Trunk Bitrate: 600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/2/0/0   CoherentDSP0/2/0/7

Traffic Split Percentage
FourHundredGigECtrlr0/2/0/1     ODU-FLEX0/2/0/0/1                             100
      0

HundredGigECtrlr0/2/0/2/1 ODU-FLEX0/2/0/0/2/1 100 0
HundredGigECtrlr0/2/0/2/2 ODU-FLEX0/2/0/0/2/2 100 0
HundredGigECtrlr0/2/0/2/1 ODU-FLEX0/2/0/7/2/3 0 100
HundredGigECtrlr0/2/0/2/2 ODU-FLEX0/2/0/7/2/4 0 100
FourHundredGigECtrlr0/2/0/4     ODU-FLEX0/2/0/7/4                               0
    100
```

800GE support in bundle configuration mode on 2.4TA cards

A 800GE client port is a high-bandwidth Ethernet interface that

- is internally divided into eight 100GE sub-channels (8x100GE),
- distributes the 100GE sub-channels across two trunk ports, Port 0 and Port 7, and
- is configured in muxponder mode using the **hw-module location** configuration commands on 2.4TA line cards.

Table 26: Feature history

Feature name	Release information	Feature description
800GE support in Bundle Configuration mode for NCS1K14-2.4T-A-K9 line card	Cisco IOS XR Release 26.2.1	The NCS1K14-2.4T-A-K9 line card now supports the 800GE client rate in muxponder mode, paired with the 400G, 600G, and 1200G trunk rates. In this mode, the 800GE client is split as 8x100GE and the sub-channels are carried over the two trunk ports, Port 0 and Port 7. This support increases bandwidth flexibility and optimizes port utilization for high-speed client traffic.

Supported line card and trunk rates

The 800GE client rate in muxponder mode is supported on NCS1K14-2.4T-A-K9 line card.

The 800GE client rate in muxponder mode is supported with these trunk rates:

- 400G
- 600G
- 1200G

Table 27: 2.4TA bundle mode configurations for 800GE client

Trunk rate (G) per trunk	Total configured trunk rate (G)	Client channel rate	Client pluggable	Split port	Client ports
400	800	1x 800GE (8x 100GE)	1x QDD-8X100	2	Clients: 2 Trunks: 0, 7
600	1200	1x 800GE + 1x 400GE (12x 100GE)	1x QDD-8X100 + 1x QDD-4X100	2	Clients: 2, 4 Trunks: 0, 7
1200	2400	3x 800GE	3x QDD-8X100	6	Clients: 2, 5, 6 Trunks: 0, 7

Note Ethernet PRBS and idle frame insertion are not supported for the 800GE split configuration in muxponder mode.

Configure 800GE in bundle mode on 2.4TA cards

Set up the 800GE client ports and the corresponding trunk rate in muxponder mode so that the 800GE client is split as 8x100GE and the sub-channels are carried over the two trunk ports, Port 0 and Port 7.

Use this task when configuring or reconfiguring line cards that require multiplexed client and trunk rates on 2.4TA hardware.

Before you begin

- Install the required client pluggables in the client ports that you plan to configure.

Follow these steps to configure the 800GE split client rate in muxponder mode.

Procedure

Step 1 Enter the **hw-module location** mxponder to specify the card location.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#
```

Step 2 Run the **trunk-rate** rate **client-port-rate** port **client-type** rate command and assign each client port to the 800GE or 400GE client type to configure the trunk rate.

Example:

This is a sample to configure 400G trunk rate with 800GE client in muxponder mode.

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 client-type 800GE
```

Example:

This is a sample to configure 600G trunk rate with 800GE + 400GE clients in muxponder mode.

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 client-type 800GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
```

Example:

This is a sample to configure 1200G trunk rate with three 800GE clients in muxponder mode.

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1200G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 client-type 800GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 800GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 client-type 800GE
```

Step 3 Commit the configuration to apply the changes.

Example:

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Step 4 Run the **show hw-module location** location **mxponder** to verify the configuration.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/0/NXR0 mxponder
Tue Jan 20 09:58:19.083 UTC

Location:             0/0/NXR0
Client Bitrate:       800GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                    Mapper/Trunk Port          CoherentDSP0/0/0/0   CoherentDSP0/0/0/7

                               Traffic Split Percentage

EightHundredGigECtrlr0/0/0/2   ODU-FLEX0/0/0/0/2                             50
     50


RP/0/RP0/CPU0:ios#show hw-module location 0/0/NXR0 mxponder
Tue Jan 20 10:06:37.996 UTC

Location:             0/0/NXR0
Client Bitrate:       800GE
Trunk  Bitrate:       600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
```

```
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/0/0/0   CoherentDSP0/0/0/7

                                Traffic Split Percentage

EightHundredGigECtrlr0/0/0/2    ODU-FLEX0/0/0/0/2                               75
     25


RP/0/RP0/CPU0:ios#show hw-module location 0/0/NXR0 mxponder
Tue Jan 20 10:17:04.069 UTC

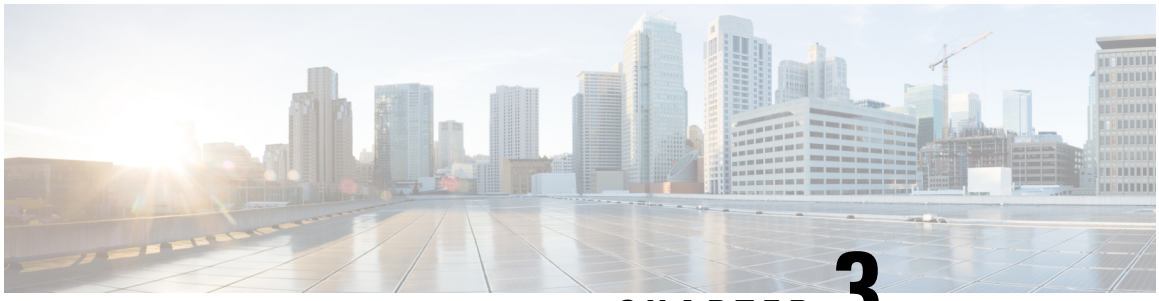
Location:             0/0/NXR0
Client Bitrate:       800GE
Trunk  Bitrate:       1200G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/0/0/0   CoherentDSP0/0/0/7

                                Traffic Split Percentage

EightHundredGigECtrlr0/0/0/6    ODU-FLEX0/0/0/7/6                               50
     50
```

Confirm that the client port shows the **800GE** client bitrate, the trunk bitrate matches the configured trunk rate, and the client-to-trunk mapping is distributed across Port 0 and Port 7 .

The line card runs in muxponder mode with the configured 800GE client ports. The 800GE client traffic is split as 8x100GE and distributed across the Port 0 and Port 7 trunks according to the selected trunk rate.



CHAPTER 3

Configuring Controllers

There are three types of controllers for the line card. The controllers are the optics controller, the ethernet controller, and the coherent DSP controller. This chapter describes the procedures used to configure these controllers.

- Automatic-In-Service, on page 56
- Display AINS configuration, on page 58
- FEC, on page 62
- Laser squelching, on page 66
- Hold-off timer, on page 70
- Idle Insertion for Ethernet controllers, on page 72
- **FlexO GID and IID**, on page 75
- Link Layer Discovery Protocol (LLDP) support on management interface, on page 76
- MAC address snooping on client ports, on page 81
- Transmit shutdown, on page 83
- Loopback, on page 85
- Loop-and-drop mechanism on Ethernet controller, on page 92
- Restore factory settings, on page 95
- Headless mode, on page 97
- Trail Trace Identifier, on page 97
- Configure chromatic dispersion, on page 100
- Configure Transmit Power, on page 104
- Configure laser bias current high threshold, on page 105
- Configure differential group delay threshold, on page 107
- Configure optical signal to noise ratio, on page 109
- Configure receive power threshold, on page 111
- Frequency, on page 113
- Baud rate, on page 114
- CCMD-16 controllers, on page 115
- Configure Controller Parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L Cards, on page 119
- Verify Rx VOA fixed ratio to zero, on page 124
- Last link flap detection, on page 127

Automatic-In-Service

Automatic-In-Service (AINS) is a controller feature that:

- allows the controller to automatically transition to the in-service state after the maintenance window,
- associates a soak time period with the automatic-in-service (AINS) state, and
- suppresses alarms from propagating to EMS/NMS monitoring systems during the AINS maintenance window.

You can configure AINS on the client ports of the QXP, 1.2T, 2.4T, and 2.4TX cards.

AINS states

If there are any service-affecting alarms when AINS is running on ethernet or optics controllers, the AINS state moves to Pending state. When the alarms are cleared, the AINS state moves to Running state.

The AINS soak time period restarts when there are line card reloads, XR reloads, line card warm reloads, power cycles, or alarm conditioning.

This table lists the AINS states.

State	Description
None	AINS is not enabled on the controller or the soak time period is complete.
Pending	AINS is configured on the controller. However, the soak time period has not started because either the primary state of the controller is in Shutdown, Admin down, or Not ready state or the secondary state is in Maintenance state. AINS can also move to Pending state if alarms are raised during the soak time period.
Running	AINS is enabled on the controller. The primary state of the controller is Up and the secondary state is AINS.

Soak time periods

A soak time period is a port timing parameter that:

- suppresses alarms during the Automatic In Service (AINS) state,
- begins when all optical and Ethernet alarms on the port are cleared, and
- specifies the time before a port transitions from AINS to In Service (IS) state.

You can configure the soak time period to be between 1 minute to 48 hours.

All alarms are suppressed during the AINS state. If optical or Ethernet alarms are raised during the soak time period, the AINS state moves to Pending. These alarms are shown in the output of the `show alarms brief card location 0/RP0/CPU0 conditions` command, not in `show alarms brief card location 0/RP0/CPU0 active`. After all alarms are cleared, the soak time period initiates, and the AINS state moves to Running. When the soak time period expires, the port transitions to the IS state.

Configure Automatic In-Service (AINS)

Use this task to enable and set the soak time for Automatic In-Service (AINS) on a controller, a muxponder slice, or globally within the system.

AINS allows specific hardware components or the entire system to automatically transition to an in-service state after a defined soak time and reduces manual intervention.

Procedure

Step 1 Run the **automatic-in-service controller** *controller rate controller* **hours** *hours* **minutes** *minutes* command to configure AINS on a controller.

Example:

This is a sample to configure AINS globally.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ains-soak hours 0 minutes 15
RP/0/RP0/CPU0:ios(config)#commit
```

Step 2 Run the **hw-module location** *location* **mxponder-slice** *slice-number* **client-port-ains-soak hours** *hours* **minutes** *minutes* command to configure AINS on a muxponder slice.

Example:

This example shows how to configure slice 0 client ports with AINS, setting the soak time to 40 minutes.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/3/nxr0 mxponder-slice 0 client-port-ains-soak hours
0 minutes 40
RP/0/RP0/CPU0:ios(config)#commit
```

Step 3 Run the **ains-soak hours** *hours* **minutes** *minutes* command to configure AINS globally.

Example:

This is a sample to configure AINS globally.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ains-soak hours 0 minutes 15
RP/0/RP0/CPU0:ios(config)#commit
```

Disable AINS

Use this task to disable the AINS on a muxponder slice, which affects all client ports within that slice.

Procedure

Step 1 Run the **hw-module location** *location* **mxponder-slice** *slice-number* **client-port-ains-soak hours** *hours* **minutes** *minutes* command to disable AINS on all muxponder client ports. Set the hours and minutes to 0.

Example:

This is a sample in which AINS is disabled on all client ports.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/3/nxr0 mxponder-slice 0 client-port-ains-soak hours
0 minutes 0
RP/0/RP0/CPU0:ios(config)#commit
```

Step 2 Run the **hw-module location** *location* **mxponder-slice** *slice-number* **client-port-ains-soak hours** *hours* **minutes** *minutes*command to disable AINS on a muxponder slice. Set the hours and minutes to 0.

Example:

This is a sample in which AINS is disabled on all client ports of slice 0.

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/3/nxr0 mxponder-slice 0 client-port-ains-soak hours
0 minutes 0
RP/0/RP0/CPU0:ios(config)#commit
```

Display AINS configuration

Use this task to view the current AINS configuration status, including the soak state and remaining duration, for various controller types.

The AINS Soak field in the output indicates the current state of AINS. The current state can be None, Pending, or Running. The Total Duration field indicates the total soak time period that is configured. The Remaining Duration field indicates the soak time that remains, after which, the AINS state moves to None.

Procedure

Run the `show controller` command for the desired interface type and location.

Example:

This example shows the Ethernet controller statistics with AINS Soak in running state.

```
RP/0/RP0/CPU0:ios#show controller HundredGigECtrlr 0/1/0/2
Thu Feb 21 19:52:55.001 UTC
Operational data for interface HundredGigECtrlr0/1/0/2:
State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
```

```
    AINS Soak: Running
    Total Duration: 0 hour(s) 15 minute(s)
    Remaining Duration: 0 hour(s) 5 minute(s) 37 second(s)
    Laser Squelch: Disabled
Phy:
    Media type: Not known
Autonegotiation disabled.
Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    BER monitoring:
        Not supported
    Holdoff Time: 0ms
```

Example:

This example shows the Ethernet controller statistics with AINS Soak in pending state.

```
RP/0/RP0/CPU0:ios#show controllers HuC 0/0/0/2
Thu Mar 12 13:52:12.129 UTC
Operational data for interface HundredGigECtrlr0/0/0/2:
State:
    Administrative state: enabled
    Operational state: Down (Reason: State undefined)
    LED state: Red On
    Maintenance: Disabled
    AINS Soak: Pending
    Total Duration: 0 hour(s) 30 minute(s)
    Remaining Duration: 0 hour(s) 30 minute(s) 0 second(s)
    Laser Squelch: Disabled
Phy:
    Media type: Not known
    Alarms:
        Current:
            Local Fault
    Statistics:
        FEC:
            Corrected Codeword Count: 0
            Uncorrected Codeword Count: 9
Autonegotiation disabled.
Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Example:

This example shows the optics controller statistics with AINS Soak in running state.

```
RP/0/RP0/CPU0:ios#show controller optics 0/1/0/3
Thu Feb 21 19:45:41.088 UTC
Controller State: Up
 Transport Admin State: Automatic In Service
 Laser State: On
 LED State: Green
 Optics Status
```

```
        Optics Type:  400G QSFP-DD DR4
        Alarm Status:
        -------------
        Detected Alarms: None
        LOS/LOL/Fault Status:
        Alarm Statistics:
        -------------
        HIGH-RX-PWR = 0            LOW-RX-PWR = 0
        HIGH-TX-PWR = 0            LOW-TX-PWR = 0
        HIGH-LBC = 0               HIGH-DGD = 0
        OOR-CD = 0                 OSNR = 0
        WVL-OOL = 0                MEA  = 0
        IMPROPER-REM = 0
        TX-POWER-PROV-MISMATCH = 0
        Performance Monitoring: Enable
        THRESHOLD VALUES
        ----------------

        Parameter                High Alarm  Low Alarm  High Warning  Low Warning
        -----------------------  ----------  ---------  ------------  -----------
        Rx Power Threshold(dBm)          4.9      -12.0           0.0          0.0
        Tx Power Threshold(dBm)          3.5      -10.1           0.0          0.0
        LBC Threshold(mA)                N/A        N/A          0.00         0.00

        LBC High Threshold = 98 %
        Polarization parameters not supported by optics

       Total TX Power = 6.39 dBm
       Total RX Power = 5.85 dBm
        Lane  Laser Bias    TX Power    RX Power  Output Frequency
        ----  ----------  ----------  ----------  ----------------
          1      75.0 %    0.59 dBm    0.63 dBm  230.43 THz
          2      68.6 %    0.06 dBm   -0.68 dBm  230.43 THz
          3      69.0 %    0.26 dBm   -0.63 dBm  230.43 THz
          4      69.1 %    0.56 dBm   -0.10 dBm  230.43 THz
 Transceiver Vendor Details
  Form Factor   : QSFP-DD
  Name    : INNOLIGHT
  Part Number   : T-DP4CNT-NGL
  Rev Number   : 1A
  Serial Number   : INLBFI940027
  PID    : T-DP4CNT-NGL
  VID    : 1A
  Date Code(yy/mm/dd)  : 21/08/21
  Fiber Connector Type: MPO
  Otn Application Code: Not Set
  Sonet Application Code: Not Set
  Ethernet Compliance Code: 400GBASE-DR4
 Transceiver Temperature : 32 Celsius
 AINS Soak               : Running
 AINS Timer              : 0h, 15m
 AINS remaining time     : 771 seconds
```

When the soak time expires, AINS state changes from Running to None. The Transport Admin State of optics controller changes from Automatic In Service to In Service.

```
RP/0/RP0/CPU0:ios# show controllers optics 0/1/0/3
Thu Feb 21 20:02:34.126 UTC
Controller State: Up
 Transport Admin State: In Service
 Laser State: On
 LED State: Green
 Optics Status
```

```
        Optics Type:  Grey optics
        Alarm Status:
        -------------
        Detected Alarms: None
        LOS/LOL/Fault Status:
        Alarm Statistics:
        -------------
        HIGH-RX-PWR = 0            LOW-RX-PWR = 0
        HIGH-TX-PWR = 0            LOW-TX-PWR = 0
        HIGH-LBC = 0               HIGH-DGD = 0
        OOR-CD = 0                 OSNR = 0
        WVL-OOL = 0                MEA  = 0
        IMPROPER-REM = 0
        TX-POWER-PROV-MISMATCH = 0
        Performance Monitoring: Enable
        THRESHOLD VALUES
        ----------------

        Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
        ------------------------  ----------  ---------  ------------  -----------
        Rx Power Threshold(dBm)          4.9      -12.0           0.0          0.0
        Tx Power Threshold(dBm)          3.5      -10.1           0.0          0.0
        LBC Threshold(mA)                N/A        N/A          0.00         0.00
        LBC High Threshold = 98 %
        Polarization parameters not supported by optics
       Total TX Power = 6.41 dBm
       Total RX Power = 5.85 dBm
        Lane  Laser Bias    TX Power    RX Power  Output Frequency
        ----  ----------  ----------  ----------  ----------------
          1      74.9 %    0.60 dBm    0.63 dBm  230.43 THz
          2      68.6 %    0.06 dBm   -0.70 dBm  230.43 THz
          3      69.0 %    0.30 dBm   -0.63 dBm  230.43 THz
          4      69.1 %    0.57 dBm   -0.11 dBm  230.43 THz
Transceiver Vendor Details
        Form Factor             : QSFP28
        Name                    : CISCO-FINISAR
        Part Number             : FTLC1152RGPL-C2
        Rev Number              : CISCO-FINISAR
        Serial Number           : FNS22150LEC
        PID                     : QSFP-100G-CWDM4-S
        VID                     : V02
        CISCO-FINISAR
        Date Code(yy/mm/dd)     : 18/04/11
        Fiber Connector Type: LC
        Sonet Application Code: Not Set
        Ethernet Compliance Code: 100GBASE-CWDM4
Transceiver Temperature : 32 Celsius
AINS Soak               : None
AINS Timer              : 0h, 0m
AINS remaining time     : 0 seconds
```

This example displays the coherentDSP controller statistics with AINS Soak in running state for a 2.4TX card.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7
Port                                          : CoherentDSP 0/0/0/7
Controller State                              : Up
Inherited Secondary State                     : Automatic-In-Service
Configured Secondary State                    : Normal
Derived State                                 : Automatic-In-Service
Loopback mode                                 : None
BER Thresholds                                : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                        : Enable
Bandwidth                                     : 800.0Gb/s
Alarm Information:
```

```
LOS = 0 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 1
FECMISMATCH = 0 FEC-UNC = 0     FLEXO_GIDM = 1
FLEXO-MM = 0    FLEXO-LOM = 0   FLEXO-RDI = 1
FLEXO-LOF = 1
Detected Alarms                                 : None
Bit Error Rate Information
PREFEC  BER                                     : 2.95E-04
POSTFEC BER                                     : 0.00E+00
Q-Factor                                        : 10.70 dB
Q-Margin                                        : 4.40 dB
Instantaneous Q-Margin                          : 4.40 dB
TTI :
        Remote hostname                         : chassisA164
        Remote interface                        : CoherentDSP 0/1/0/7
        Remote IP addr                          : 0.0.0.0
FEC mode                                        : Soft-Decision 15
Flexo-Mode                                      : Enable
Flexo Details:
        Tx GID                                  : 1
        TX IID                                  : 1, 2, 3, 4, 5, 6, 7, 8,
        Rx GID                                  : 1
        RX IID                                  : 1, 2, 3, 4, 5, 6, 7, 8,

AINS Soak                                       : Running
AINS Timer                                      : 0h, 20m
AINS remaining time                             : 1196 seconds
```

FEC

Forward Error Correction (FEC) is a method for controlling errors during data transmission that:

- adds data redundancy to the transmitted message using an algorithm, and
- enables the receiver to detect and correct a limited number of errors occurring anywhere in the message, thereby eliminating the need for the transmitter to resend the message.

This feature can be enabled on 1.2T and 2.4T cards. FEC is automatically enabled on the pluggables installed in the NCS 1014.

Note When you upgrade the software of an NCS 1014 with pluggables in the FEC disabled mode, traffic is affected.

FEC states for CoherentDSP controller

This table lists the FEC states for the coherentDSP controllers.

Table 28: FEC state for CoherentDSP controllers

State	Description
EnhancedSD15	FEC Soft-Decision 15. (Default)

Q-Margin support

Q-margin is an important optical parameter that characterizes the health of an optical link. The Q-margin value is calculated based on the average bit error rate (BER) in the optical link.

Enhanced Q-Margin support

Enhanced Q-Margin is supported for Forward Error Correction (FEC) and Performance Monitoring on CoherentDSP controllers for 2.4T cards. Enhanced Q-margin provides a better error free signal in the optical link. The enhanced Q-margin value is calculated based on the maximum number of errors per frame.

An attribute that is called instantaneous Q-margin is displayed in the output of the **show controllers coherentDSP** command. The lower the delta value between the instantaneous Q-margin value with the Q-margin value, the better the FEC performance of the NCS 1014 system.

The instantaneous Q-margin values thus help you to optimize the system with continuous error correction in subsea transport networks.

Configure FEC on the Ethernet controller

Use this task to configure FEC on the CoherentDSP controller.

Procedure

Run the **controller { HundredGigECtrlr | fourHundredGigECtrlr }** *R/S/I/P* **fec { none | standard }** command.

Example:

This sample shows how to configure FEC on the Ethernet controller.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller HundredGigECtrlr 0/1/0/6/ fec standard
RP/0/RP0/CPU0:ios(config)#commit
```

This sample shows the running FEC configuration on the Ethernet controller.

```
RP/0/RP0/CPU0:BH-SIT2#show controller HundredGigECtrlr 0/1/0/6
Tue Jul 16 15:30:30.165 IST
Operational data for interface HundredGigECtrlr0/1/0/6/2:

State:
    Administrative state: enabled
    Operational state: Down (Reason: State undefined)
    LED state: Red On
    Maintenance: Disabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled

Phy:
    Media type: Not known
    Alarms:
        Current:
            Loss of Frequency Sync Data
    Statistics:
        FEC:
```

```
            Corrected Codeword Count: 0
            Uncorrected Codeword Count: 0
Autonegotiation disabled.
Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

From Release 24.2.11, the **show controllers** command output displays post FEC BER and pre FEC BER for the Ethernet controllers of the 2.4T and 2.4TX cards.

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/1/0/2
Thu Jan 18 18:58:03.846 UTC
Operational data for interface FourHundredGigECtrlr0/1/0/2:
State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled
Phy:
    Media type: Not known
    Statistics:
        FEC:
            Corrected Codeword Count: 358543                  Valid: True       Start time: 18:54:03
 Thu Jan 18 2024
            Uncorrected Codeword Count: 0                     Valid: True       Start time: 18:54:03
 Thu Jan 18 2024
        PCS:
            Total BIP errors: 0                               Valid: True       Start time: 18:54:03
 Thu Jan 18 2024
            Total frame errors: 0                             Valid: False      Start time: 18:54:03
 Thu Jan 18 2024
            Total Bad SH: 0                                   Valid: False      Start time: 18:54:03
 Thu Jan 18 2024
Autonegotiation disabled.
Operational values:
    Speed: 400Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    Pre FEC BER: 9.1E-10
    Post FEC BER: 0.0E+00
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Configure FEC on CoherentDSP controllers

Use this task to configure FEC on the CoherentDSP controller.

Procedure

Run the **controller coherentDSP** *R/S/I/P* **fec** {**EnhancedSD15**} command.

Example:

This sample shows how to configure FEC on the CoherentDSP controller.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/0
RP/0/RP0/CPU0:ios(config-CoDSP)#fec EnhancedSD15
Tue Feb 25 11:25:52.670 UTC
WARNING! Changing FEC mode can impact traffic
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

Verify FEC on CoherentDSP controllers

This sample shows the FEC configuration on the CoherentDSP controller.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/0

Tue Feb 25 11:26:08.235 UTC


Port                                          : CoherentDSP 0/0/0/0
Controller State                              : Up
Inherited Secondary State                     : Normal
Configured Secondary State                    : Normal
Derived State                                 : In Service
Loopback mode                                 : None
BER Thresholds                                : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                        : Enable
Bandwidth                                     : 50.0Gb/s
Alarm Information:
LOS = 1 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0
Detected Alarms                               : None
Bit Error Rate Information
PREFEC  BER                                   : 0.00E+00
POSTFEC BER                                   : 0.00E+00
Q-Factor                                      : 0.00 dB
Q-Margin                                      : -5.00dB
Instantaneous Q_margin                      :  0 dB
TTI :
Remote IP addr                                : 0.0.0.0
FEC mode                                      : Soft-Decision 15

AINS Soak                                     : None
AINS Timer                                    : 0h, 0m
```

```
AINS remaining time                          : 0 seconds
```

Laser squelching

You can enable laser squelching on Ethernet controllers. Laser squelching can be enabled on QXP, 1.2T, 2.4T, and 2.4T-X cards which shuts down the laser in the event of trunk faults (LOS, LOF), and a SQUELCHED alarm is raised on the mapped client port.

Laser squelching uses an interrupt based method. Therefore, squelching occurs faster than in previous releases.

Beyond trunk faults, squelching occurs in 1.2T cards, for the client alarms Ingress LF, LOA, and CSF (ingress only).

Configure laser squelching

Enable or disable the laser squelch feature for specific optical cards, enhancing operational control and safety.

Laser squelch disables the laser output on designated ports/faults to prevent unwanted signal transmission during error conditions or maintenance.

Procedure

Step 1 Configure laser squelching for the desired card type:

For 1.2T cards, enter **controller HundredGigECtrlr** *Rack/Slot/Instance/Port/Lane* **laser-squelch** command.

For 2.4T or 2.4TX cards, enter **controller HundredGigECtrlr** | **controller fourHundredGigECtrlr** *Rack/Slot/Instance/Port/Lane* **laser-squelch** command.

Example:

This is a sample where laser squelching is enabled on the Ethernet controller of a 2.4T card and a 2.4TX card.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller fourHundredGigECtrlr 0/1/0/1
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#laser-squelch
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

This is a sample to view the laser squelch status on the controller of a 2.4T card and a 2.4TX card.

Example:

Note
In case of muxponder configuration on the 2.4T-X card, and if laser squelch is configured on the split port, whenever a fault is received on any one trunk, the corresponding split port will be squelched.

Step 2 Verify the laser squelch status on the controller of a 2.4T card or a 2.4TX card.

Example:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4
Fri Nov 17 14:26:43.213 UTC
Operational data for interface FourHundredGigECtrlr0/0/0/4:
```

```
State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: Running
      Total Duration: 0 hour(s) 5 minute(s)
      Remaining Duration: 0 hour(s) 3 minute(s) 34 second(s)
    Laser Squelch: Enabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled

Phy:
    Media type: Not known
    Statistics:
        FEC:
            Corrected Codeword Count: 580070472               Valid: False      Start time: 13:12:29
 Fri Nov 17 2023
            Uncorrected Codeword Count: 0                     Valid: False      Start time: 13:12:29
 Fri Nov 17 2023
        PCS:
            Total BIP errors: 0                               Valid: False      Start time: 13:12:29
 Fri Nov 17 2023
            Total frame errors: 0                             Valid: False      Start time: 13:12:29
 Fri Nov 17 2023
            Total Bad SH: 0                                   Valid: False      Start time: 13:12:29
 Fri Nov 17 2023

Autonegotiation disabled.

Operational values:
    Speed: 400Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Laser squelching is enabled on the specified optical card(s). The system disables laser output in response to faults or based on maintenance needs.

Protection switching use cases for Fast-Squelching

Table 29: Feature History

Feature Name	Release Information	Description
Client side PSM support for NCS1K14-2.4T-K9 and NCS1K14-2.4T-X-K9 cards configured with fast -squelching	Cisco IOS-XR Release 26.2.1	In addition to the NCS1K4-1.2T-K9 cards, now protection switching is supported on NCS1K14-2.4T-K9, and NCS1K14-2.4T-X-K9 cards with the client pluggables QDD-400G-FR4-S and the QDD-400G-LR4-S pluggables that are configured with fast-squelching. The supported datapath configurations for NCS1K14-2.4T-K9 and NCS1K14-2.4T-X-K9 cards include: • Slice mode (only for 400GE client rate) • Bundle mode configuration This update enhances network reliability by providing automated failover and signal protection for 2.4T and 2.4T-X cards, ensuring consistent uptime for high-bandwidth services.

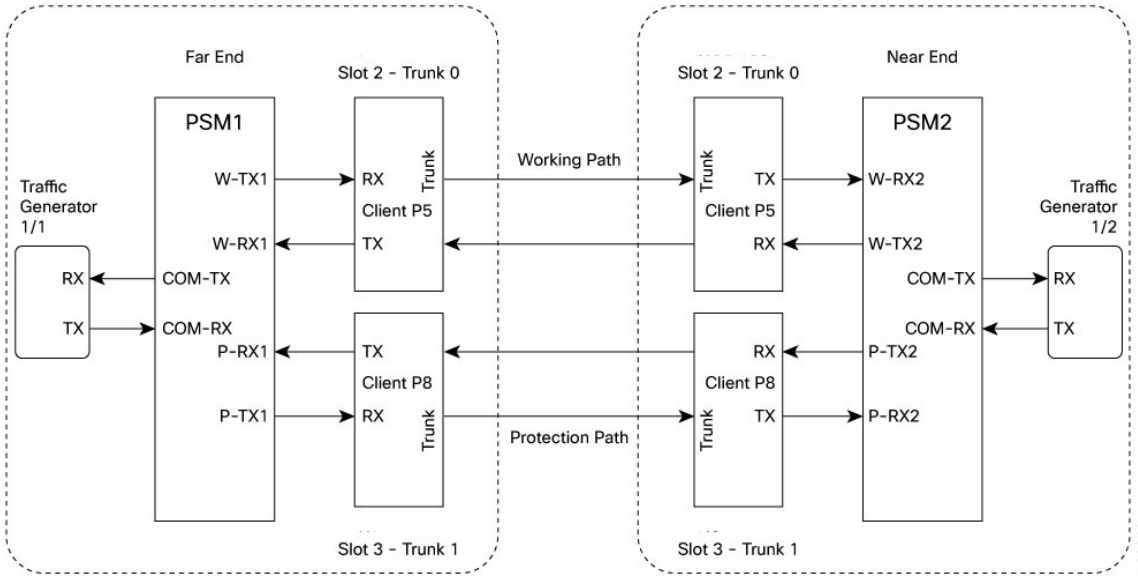
When implementing protection switching in NCS 1014 deployments that have Fast-Squelching configured, fast-squelching increases protection switching speed during trunk or client faults.



Note Protection Switching is supported on 1.2T cards and not supported on 2.4T line cards.

This sample topology includes a Far End station (FE station) and a Near End station (NE station). Each station includes an NCS 1014 node with two line cards. The nodes are connected to the respective traffic generators through a Protection Switching Module (PSM).

Figure 5: Reference topology for protection Switching



Protection switching principle (Trunk fault)

If there is a fiber cut in the trunk working path from the FE station to the NE station, an LOS alarm is raised on the NE working trunk. This results in the squelching of all client ports mapped to the working NE trunk port. As the laser of the client port is squelched, LOS is reported on the W-RX2 port of the PSM2. As the received optical power on the W-RX2 port of PSM is below the threshold, PSM2 switches to receive the optical signal in the P-RX2 port instead of the W-RX2 port. Hence switching happens for traffic from work to protect in FE station to NE station direction.

In the case of a unidirectional trunk fault, switching happens in one direction as explained above. In the other direction, when LOS is received at the W-RX2 port of PSM2, W-TX2 sends LOS for 25 milliseconds. When LOS is reported on the NE client port, fault gets propagated over the trunk, resulting in the squelching of FE station client ports. Finally, the LOS on the PSM port results in switching in this direction as well. In this way, bidirectional switching is implemented.

Protection switching principle (Client fault)

When a client failure happens on the FE station, a Client Signal Failure (CSF) alarm is raised on the NE station trunk. The CSF on the trunk results in the squelching of the corresponding client port, and the PSM switching happens. In summary, a fault on the NE station client RX port results in CSF on the FE station trunk, and the switching happens. And, a fault on the NE station client TX port results in LOS on the PSM ports, and the switching happens.

Guidelines and limitations:

- PSM must be in the standalone mode.
- PSM alarm threshold must be set to +/ –3 dBm from the actual power received in the PSM RX port.
- If line card protection is required, the working and protect path must be configured in two different line cards.
- If only client protection is required, the working and protection path can be configured in the same line card.

- If the LC trunk configuration is x50 rate, then you cannot use a single line card for work and protection due to x50 coupled mode limitations (coupled trunk)..
- Manual switch, force switch, and lock-out protection on PSM result in bidirectional switching.

Hold-off timer

When a fault occurs on the trunk port, you can hold the propagation of local faults using the hold-off time feature. This feature is can be enabled on the Ethernet controllers of 1.2T and 2.4T 1.2T, 2.4T, 2.4Tx, and QXP-K9 cards by configuring the hold-off timer.

When the fault occurs on the trunk, idles are inserted in the traffic stream from the trunk port to the client port for the duration of the configured holdoff-time. If the trunk port remains faulty beyond the configured holdoff-time, a local fault is transmitted towards the client device. If the trunk recovers from the fault before the holdoff-time expires, traffic resumes.

This feature can be used on customer deployments to prevent reset of client ports during a PSM switchover.

Configure holdoff-time on QXP Card

Use this task to configure holdoff-time on 100GE or 400GE controllers for the QXP card.

Procedure

Step 1 Run the **controller HundredGigECtrlr** *Rack/Slot/Instance/Port* **holdoff-time trunk-fault** *time-value* command to configure holdoff-time on the 100GE controllers for the QXP card.

The range of *timevalue* is from 0 ms to 3000 ms.

Example:

This is a sample where holdoff-time is enabled on the 100GE controller for the QXP card.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller HundredGigECtrlr 0/1/0/1
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#holdoff-time trunk-fault 3000
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Example:

This is a sample to view the holdoff-time status on the 100GE controller.

```
RP/0/RP0/CPU0:ios#show controller hundredGigECtrlr 0/1/0/1
Fri Jul 23 16:07:11.541 UTC
Operational data for interface HundredGigECtrlr0/1/0/1:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
```

```
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Enabled

Phy:
    Media type: Not known
Statistics:
FEC:
Corrected Codeword Count: 134967789
Uncorrected Codeword Count: 0

Autonegotiation disabled.

Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    BER monitoring:
        Not supported
        Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 3000ms
```

Step 2 Run the **controller fourHundredGigECtrlr** *Rack/Slot/Instance/Port* **holdoff-time trunk-fault** *time-value* command to configure holdoff-time on the 400GE controllers for the QXP card.

Example:

This is a sample where holdoff-time is enabled on the 400GE controller for the NCS1K4-QXP-K9 card.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller fourHundredGigECtrlr 0/0/0/10
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#holdoff-time trunk-fault 2000
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Example:

This is a sample to view the holdoff-time status on the 400GE controller.

```
RP/0/RP0/CPU0:ios#show controller fourhundredGigECtrlr 0/0/0/10
Fri Jul 23 16:07:11.541 UTC
Operational data for interface fourHundredGigECtrlr0/0/0/10:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Enabled

Phy:
    Media type: Not known
Statistics:
FEC:
Corrected Codeword Count: 134967789
Uncorrected Codeword Count: 0

Autonegotiation disabled.

Operational values:
```

```
Speed: 400Gbps
Duplex: Full Duplex
Flowcontrol: None
Loopback: None (or external)
BER monitoring:
    Not supported
    Forward error correction: Standard (Reed-Solomon)
Holdoff Time: 2000ms
```

Idle Insertion for Ethernet controllers

Idle insertion for Ethernet controllers feature allows you to perform end-to-end link verification between 100GE or 400GE or Ethernet controllers before bringing up the actual traffic. This feature enables you to perform pre-provisioning checks to isolate link errors in advance without any Ethernet testers.

Note OTU4 client rate is not supported.

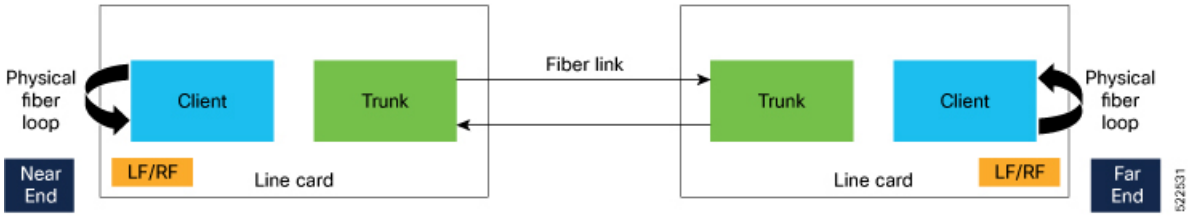
Idle frames can be inserted in both the ingress and egress directions on Ethernet controllers and the LOCAL-FAULT and REMOTE-FAULT alarms are cleared. The performance monitoring counters on the pcs layer are monitored to check for any errors on the link.

Note Do not configure the Idle insertion for Ethernet controllers feature on the link that carries live traffic.

Recommended topology for link verification

This diagram describes the recommended topology for link verification.

Figure 6: Topology for link verification



These steps describe the sequence for link verification using this topology:

1. Both the near-end and far-end clients have the LOCAL-FAULT alarm if the trunk is up on both the ends.
2. Enable idle ingress on the near-end client. The idle frame transmits toward the trunk link and reaches the far-end client. The LOCAL-FAULT alarm is then cleared on the far-end client.
3. As the far-end client has fiber loop, the idle frame is inserted again into the same client RX toward the trunk link and reaches the near-end client. The LOCAL-FAULT alarm is then cleared on the near-end client as well.

4. When you enable idle insertion on any client and in any direction, the idle frame transmits in loop similar to this topology and all the LOCAL-FAULT and the REMOTE-FAULT alarms are cleared.
5. The link can be monitored after all the alarms are cleared. The link has a problem if any alarm is reported during the link test.

Configure idle insertion on Ethernet controllers

Use this task to configure or disable idle insertion on Ethernet controllers.

Note Do not configure idle frame insertion with hold-off timer.

Procedure

Step 1 Run the **controller hundredGigECtrlr** *Rack/Slot/Instance/Port/***insert-idle ingressinsert-idle egress** command to enable this feature.

Example:

This is a sample for enabling the idle ingress and idle egress in 100GE controllers.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller hundredGigECtrlr 0/2/0/2
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#insert-idle ingress
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#insert-idle egress
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#end
```

Step 2 Run the **controller hundredGigECtrlr** *Rack/Slot/Instance/Port***no insert-idle ingressno insert-idle egress** command to disable this feature.

After disabling the idle frame insertion feature, the LOCAL-FAULT or REMOTE-FAULT alarm may not appear again because the idle frames are in loop. Hence, you must break the idle frame loop in the link by performing either one of the following.

These are the limitations:

- Perform fiber OIR on either the near-end or far-end client port.
- Perform shut and unshut operation on any client port.

Verify idle insertion configuration for Ethernet controllers

Use this task to verify idle insertion for Ethernet controllers.

Note Do not configure idle frame insertion with hold-off timer.

Procedure

Run the **show controllers hundredGigECtrlr** *Rack/Slot/Instance/Port* command to verify the *idle ingress* and *idle egress* that is configured on the Ethernet controllers of a 1.2T card.

Example:

This is a sample to verify idle insertion for 800GE controllers.

Example:

This is a sample to verify idle insertion for 100GE controllers.

```
RP/0/RP0/CPU0:ios#show controllers hundredGigECtrlr 0/2/0/2
Wed Mar 30 06:56:58.878 UTC
Operational data for interface HundredGigECtrlr0/2/0/2:
State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled

    Insert Idle Ingress: Enabled
    Insert Idle Egress: Enabled

Phy:
    Media type: Not known
    Statistics:
        FEC:
            Corrected Codeword Count: 0
            Uncorrected Codeword Count: 0
Autonegotiation disabled.
Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

FlexO GID and IID

The 2.4T and 2.4TX cards use are equipped with flexible OTN (flexO) interfaces on their trunk ports. These flexO interfaces enable the efficient and interoperable transport of OTU signals by aggregating multiple standard lower-rate interfaces.

Each group of flexO interfaces is assigned a unique flexO group identification (GID) number, which can range from 1 to 1,048,576. Within each group, individual interfaces are identified by a flexO instance identification (IID) number. Note that the IID for each member is fixed and cannot be modified.

Configure FlexO GID and IID

Use this task to configure flexO GID and IID on the coherentDSP controller.

Procedure

Step 1 Run the **configure** command to enter the global configuration mode.

Example:

This sample shows how to configure flexO GID on the CoherentDSP controller.

```
RP/0/RP0/CPU0:ios#configure terminal
Mon Feb  5 05:14:42.919 UTC
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/0
RP/0/RP0/CPU0:ios(config-CoDSP)#flexo gid 1048575
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

Step 2 Run the **show controllers** command to verify the configuration.

Example:

This sample shows the flexO GID configuration on the CoherentDSP controller.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/0
Mon Feb  5 05:20:01.660 UTC

Port                                        : CoherentDSP 0/0/0/0
Controller State                            : Up
Inherited Secondary State                   : Normal
Configured Secondary State                  : Normal
Derived State                               : In Service
Loopback mode                               : None
BER Thresholds                              : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                      : Enable
Bandwidth                                   : 1200.0Gb/s

Alarm Information:
LOS = 1 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 1     FLEXO_GIDM = 0
FLEXO-MM = 0    FLEXO-LOM = 0   FLEXO-RDI = 0
FLEXO-LOF = 0
```

```
Detected Alarms                                  : None

Bit Error Rate Information
PREFEC  BER                                      : 1.23E-02
POSTFEC BER                                      : 0.00E+00
Q-Factor                                         : 7.00 dB

Q-Margin                                         : 0.80 dB

Instantaneous Q-Margin                           : 0.80 dB

TTI :
        Remote hostname                          : ios
        Remote interface                         : CoherentDSP 0/0/0/7
        Remote IP addr                           : 0.0.0.0

FEC mode                                         : Soft-Decision 15

Flexo-Mode                                       : Enable
Flexo Details:
        Tx GID                                   : 1
        TX IID                                   : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
        Rx GID                                   : 1
        RX IID                                   : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,

AINS Soak                                        : None
AINS Timer                                       : 0h, 0m
AINS remaining time                              : 0 seconds
```

Step 3 Run the **commit** command to apply the changes.

Link Layer Discovery Protocol (LLDP) support on management interface

The LLDP can be configured on management interface of 1.2T, 2.4T, and 2.4TX cards. It requires a system to form LLDP neighborship over the system management interface, through which it advertises and learns LLDP neighbor information. This information about neighbors is used to learn about the neighbors and in turn the topology of the devices for Operations, Administration, and Maintenance (OAM) purposes.

Advantages of LLDP

- Provides support on non-Cisco devices.
- Enables neighbor discovery between non-Cisco devices.

Limitation

- When you disable LLDP globally, the LLDP gets disabled on all the interfaces.



Note By default, LLDP is enabled for NCS 1014. But when you enable and disable LLDP in the global configuration mode, LLDP gets disabled on all the interfaces.

Workaround : You must enable LLDP globally or reload the Router.

Cisco Discovery Protocol (CDP) vs LLDP

The 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 devices, such as routers, bridges, access servers, and switches. This protocol allows the network management applications to automatically discover and learn about other Cisco devices that connect to the network.

The LLDP is also a device discovery protocol that runs over Layer 2. This protocol allows the network management applications to automatically discover and learn about other non-Cisco devices that connect to the network.

Interoperability between non-Cisco devices using LLDP

LLDP is also a neighbor discovery protocol that is used by network devices to advertise information about themselves to other devices on the network. This protocol runs over the data link layer, which allows two systems running different network layer protocols to learn about each other.

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

- 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
- 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

Neighbor discovery

System advertises the LLDP TLV (Type Length Value) details over the management network using which other devices in the management network can learn about this device.

Configuring LLDP

- LLDP full stack functionality is supported on all three management interfaces supported in NCS 1014.
- You can selectively enable or disable LLDP on any of the management interfaces on demand.
- You can selectively enable or disable LLDP transmit or receive functionality at the management interface level.

- Information gathered using LLDP can be stored in the device Management Information Database (MIB) and queried with the Simple Network Management protocol (SNMP).
- LLDP operational data are available in both Command Line Interface and netconf-yang interface.

Enabling LLDP globally

When you enable LLDP globally, all interfaces that support LLDP are automatically enabled for both transmit and receive operations.

Note You can override this default operation at the interface to disable receive or transmit operations.

This table describes the global LLDP attributes that the user can configure:

Table 30: Global LLDP attributes

Attribute	Default	Range	Description
Holdtime	120	0–65535	Specifies the hold time (in sec). Hold time refers to the time or duration that an LLDP device maintains the neighbor information before discarding.
Reinit	2	2–5	Delay (in sec) for LLDP initialization on any interface
Timer	30	5-65534	Specifies the rate at which LLDP packets are sent (in sec)

Debugging LLDP issues

These commands are used for debugging issues in the LLDP functionality.

- **show lldp traffic**
- **debug lldp all**
- **debug lldp errors**
- **debug lldp events**
- **debug lldp packets**
- **debug lldp tlvs**
- **debug lldp trace**
- **debug lldp verbose**

Configure LLDP

Use this task to configure LLDP globally.

Before you begin

- LLDP full stack functionality is supported on all three management interfaces supported in NCS 1014.
- You can selectively enable or disable LLDP on any of the management interfaces on demand.
- You can selectively enable or disable LLDP transmit or receive functionality at the management interface level.
- Information gathered using LLDP can be stored in the device Management Information Database (MIB) and queried with the Simple Network Management protocol (SNMP).
- LLDP operational data are available in both Command Line Interface and netconf-yang interface.

Procedure

Run the `configure terminal` command to enter the configuration mode.

Example:

This example shows the commands to configure LLDP globally. The global LLDP configuration enables LLDP on all the three management interfaces.

```
RP/0/RP0/CPU0:regen#configure terminal
RP/0/RP0/CPU0:regen(config)#lldp management enable
RP/0/RP0/CPU0:regen(config)#lldp holdtime 30
RP/0/RP0/CPU0:regen(config)#lldp reinit 2
RP/0/RP0/CPU0:regen(config)#commit
```

Example:

This example shows the commands to configure LLDP at the management interface level.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp enable
RP/0/RP0/CPU0:ios(config-if)#commit
```

Example:

This example shows the commands to disable the LLDP transmit operations at the specified management interface.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp transmit disable
RP/0/RP0/CPU0:ios(config-if)#commit
```

Example:

This example shows the commands to disable the LLDP receive operations at the specified management interface.

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/X
RP/0/RP0/CPU0:ios(config-if)#lldp receive disable
RP/0/RP0/CPU0:ios(config-if)#commit
```

Verify LLDP configuration

Use this task to verify LLDP configuration.

Procedure

Step 1 Run the **show running-config lldp** command to verify the LLDP configuration.

Example:

```
RP/0/RP0/CPU0:regen#show running-config lldp
Tue Dec 10 10:36:11.567 UTC
lldp
timer 30
reinit 2
holdtime 120
management enable
!
```

Step 2 Run the **show lldp interface** and **show lldp neighbors** commands to verify the LLDP data.

Example:

```
RP/0/RP0/CPU0:regen#show lldp interface
Thu Nov  7 08:45:22.934 UTC
MgmtEth0/RP0/CPU0/0:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
MgmtEth0/RP0/CPU0/1:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
```

Example:

```
RP/0/RP0/CPU0:M-131#show lldp neighbors
Mon Dec  2 11:01:20.143 CET
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
[DISABLED]      MgmtEth0/RP0/CPU0/0     120        B               gi19
MYS-130         MgmtEth0/RP0/CPU0/1     120        R               MgmtEth0/RP0/CPU0/1
```

where [DISABLED] shows that the LLDP is disabled on the interface MgmtEth0/RP0/CPU0/0.

When you enable LLDP drop on the client controller ports of the muxponder or muxponder slice, the LLDP frames drop on the ports without forwarding.

Note
LLDP on 400GE is not supported on the OTN-XP card.

MAC address snooping on client ports

A MAC address snooping on client ports feature is a network monitoring capability that:

- learns the MAC address of the connected neighbor on 1.2T cards at client ports
- enables ARP snooping across all client ports through CLI, and
- helps overcome environments where LLDP cannot be used for neighbor discovery.

This feature overcomes the limitation, where LLDP (Link Layer Discovery protocol) cannot be enabled in some networks.

Limitations

- When you enable or disable MAC address snooping on any slice, few packets are dropped during configuration.
- Open config interface for enabling or disabling MAC address snooping is not supported.
- SNMP MIB is not supported for the MAC address attribute.

Note When you enable MAC address snooping on client ports, it overrides LLDP.

Configure MAC address snooping on client ports

Use this task to configure MAC address or ARP snoop on slice in muxponder slice mode.

Procedure

Run the **hw-module location** *location* **mxponder-slice** *slice-number* **client-rate 100GE trunk-rate 600G { 100G | 150G | 200G | 250G | 300G | 350G | 400G | 450G | 500G | 550G | 600G } arp-snoop** command.

Example:

This is a sample in which, MAC address or ARP snoop is configured on the client ports of slice 0 in Muxponder slice mode.

```
RP/0/RP0/CPU0:ios#configure
Mon Mar 16 19:30:33.933 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/3/nxr0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-rate 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#arp-snoop
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
Mon Mar 16 19:30:52.636 UTC
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

This is a sample in which, MAC address or ARP snoop is configured in Muxponder mode.

```
RP/0/RP0/CPU0:ios#configure
Mon Mar 16 19:08:17.154 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1 mxponder arp-snoop
RP/0/RP0/CPU0:ios(config)#commit
```

This sample shows the output of **show controllers hundredGigEctrlr** command, before configuring MAC address or ARP snoop on client ports.

```
RP/0/RP0/CPU0:ios#show controllers HundredGigECtrlr 0/1/0/2/1
Mon Mar 16 19:40:37.434 UTC
Operational data for interface HundredGigECtrlr0/1/0/2/1:
State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled
Phy:
    Media type: Not known
Autonegotiation disabled.
Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: None (or external)
    BER monitoring:
        Not supported
    Holdoff Time: 0ms
```

View neighbor MAC address

Use this task to view the neighbor's physical address after enabling MAC address or ARP snoop.

MAC address snoop output is enabled after ARP packets are received on the respective 100G client.

Procedure

Run the **show controllers hundredGigEctrlr** *R/S/I/P* command.

Example:

This sample shows the neighbor's MAC address after configuring MAC address or ARP snoop on client ports.

```
RP/0/RP0/CPU0:ios#show controllers HundredGigECtrlr 0/1/0/2/1
Mon Mar 16 19:41:08.047 UTC
Operational data for interface HundredGigECtrlr0/1/0/2/1:
State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Disabled
    AINS Soak: None
```

```
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled
    Neighbor Address:
    0010.9400.5502
Phy:
    Media type: Not known
Autonegotiation disabled.
Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
```

Transmit shutdown

Transmit shut on trunk optics controller brings down the CIM8 and PICO transmit power. You can configure transmit shut on optics controller of a 1.2T, 2.4TX, or 2.4T card.

Configure transmit shutdown on trunk optics controller

To perform transmit shutdown, enter these commands:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:chassisA164(config)#controller optics 0/1/0/0
RP/0/RP0/CPU0:chassisA164(config-Optics)#transmit-shutdown
RP/0/RP0/CPU0:chassisA164(config-Optics)#commit
RP/0/RP0/CPU0:ios(config-Optics)#exit
RP/0/RP0/CPU0:ios(config)#exit
```

Verify transmit shutdown on trunk optics controller

Use this task to verify the transmit shutdown details on the trunk optics controller.

Procedure

Run the **show controllers optics***R/S/I/P* command to view the transmit shutdown details.

Example:

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/0
Tue Dec 12 05:38:32.416 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: Off
LED State: Green
Optics Status
     Optics Type:  CIM8 DWDM
     DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
     Wavelength=1552.524nm

     Alarm Status:
```

```
    -------------
    Detected Alarms: None
    LOS/LOL/Fault Status:
    Alarm Statistics:
    -------------
    HIGH-RX-PWR = 0            LOW-RX-PWR = 0
    HIGH-TX-PWR = 0            LOW-TX-PWR = 2
    HIGH-LBC = 0               HIGH-DGD = 0
    OOR-CD = 0                 OSNR = 1
    WVL-OOL = 0                MEA  = 0
    IMPROPER-REM = 0
    TX-POWER-PROV-MISMATCH = 0
    Laser Bias Current = 0.0 %
    Actual TX Power = -40.00 dBm
    RX Power = -6.60 dBm
    RX Signal Power = -7.06 dBm
    Frequency Offset = -846 MHz
    Performance Monitoring: Enable

    THRESHOLD VALUES
    ----------------

    Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
    ------------------------  ----------  ---------  ------------  -----------
    Rx Power Threshold(dBm)         13.0      -17.0           0.0          0.0
    Tx Power Threshold(dBm)          5.0      -13.0           0.0          0.0
    LBC Threshold(mA)                N/A        N/A          0.00         0.00

    LBC High Threshold = 90 %
    Configured Tx Power = 2.00 dBm
    Configured CD High Threshold = 180000 ps/nm
    Configured CD lower Threshold = -180000 ps/nm
    Configured OSNR lower Threshold = 20.50 dB
    Configured DGD Higher Threshold = 90.00 ps
    Baud Rate =  137.9783780000 GBd
    Bits per Symbol = 3.5200000000  bits/symbol
    Modulation Type: PCS
    Chromatic Dispersion 0 ps/nm
    Configured CD-MIN -10000 ps/nm  CD-MAX 48000 ps/nm
    Polarization Mode Dispersion = 0.0 ps
    Second Order Polarization Mode Dispersion = 9.00 ps^2
    Optical Signal to Noise Ratio = 36.90 dB
    SNR = 16.10 dB
    Polarization Dependent Loss = 0.90 dB
    Polarization Change Rate = 0.00 rad/s
    Differential Group Delay = 1.00 ps
    Filter Roll Off Factor : 0.100
    Rx VOA Target Power : -2.0 dBm
    NLEQ Compensation Mode : 0
    Cross Polarization Gain Mode : 10
    Proprietary Submarine Parameters
      Type : 1        Value : 0
      Type : 2        Value : 0
      Type : 3        Value : 0
      Type : 4        Value : 0
      Type : 5        Value : 10485760


Transceiver Vendor Details

    Form Factor            : CIM8
    Name                   : CISCO-ACACIA
    Serial Number          : N/A
    PID                    : CIM8-C-K9
```

```
     VID                    : N/A
     Date Code(yy/mm/dd)    : 23/10/20
     Fiber Connector Type: LC
     Otn Application Code: Not Set
     Sonet Application Code: Not Set
     Ethernet Compliance Code: Not set

 Transceiver Temperature : 38 Celsius

AINS Soak               : None
AINS Timer              : 0h, 0m
AINS remaining time     : 0 seconds
```

Loopback

You can configure loopback on the CoherentDSP and Ethernet controllers of QXP, 1.2T, 2.4TX and 2.4T cards to identify connection problems. The loopback can be configured only in the maintenance mode. Use the **controller** ***controller-type*** and the **secondary-admin-state maintenance** commands to place the controllers in the maintenance mode.

Loopback configuration alarm details for each controller are triggered whenever there is a change in the loopback configuration. Details such as, location of the controller, severity, configuration date and time, and description are available in the output of the **show alarms brief system active** and **show alarms brief history** commands.

Configure loopback

Use this task to configure loopback on a 1.2T, 2.4T, and 2.4TX card.

Procedure

Step 1 Run the **controller***controllertype Rack/Slot/Instance/Port***sec-admin-state maintenance loopback** [**internal**] command to configure the loopback on a 1.2T card.

Example:

This example shows how a internal loopback is configured on the Ethernet controller of a 1.2T card.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller HundredGigECtrlr 0/1/0/1
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#loopback internal
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
RP/0/RP0/CPU0:ios(config)#exit
```

Note
Line loopback is not supported on CoherentDSP controller of 1.2T card.

Example:

This example shows how to verify a internal loopback configured on the Ethernet controller of 1.2T card.

```
RP/0/RP0/CPU0:ios#show controller HundredGigECtrlr 0/1/0/1
Fri Nov 17 10:01:22.840 UTC
Operational data for interface HundredGigECtrlr0/1/0/1:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Enabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled

Phy:
    Media type: Not known
    Statistics:
        FEC:
            Corrected Codeword Count: 72671614                Valid: True       Start time: 15:53:12
 Thu Nov 16 2023
            Uncorrected Codeword Count: 12                    Valid: True       Start time: 15:53:12
 Thu Nov 16 2023
        PCS:
            Total BIP errors: 0                               Valid: True       Start time: 15:53:12
 Thu Nov 16 2023
            Total frame errors: 0                             Valid: False      Start time: 15:53:12
 Thu Nov 16 2023
            Total Bad SH: 0                                   Valid: False      Start time: 15:53:12
 Thu Nov 16 2023

Autonegotiation disabled.

Operational values:
    Speed: 400Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: Internal
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Step 2 Run the **controller** {**HundredGigECtrlr** *Rack/Slot/Instance/Port/Lane* | **fourHundredGigECtrlr***Rack/Slot/Instance/Port*}**sec-admin-state maintenance loopback**[**line** | **internal**] command to configure the loopback on 2.4T card and 2.4TX card.

Example:

This example shows how a line loopback is configured on coherentDSP controller of a 2.4T card and 2.4TX card.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/1/0/0
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#loopback line
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
RP/0/RP0/CPU0:ios(config)#exit
```

Note

In case of muxponder configuration on the 2.4TX card, internal loopback must be applied on both trunk ports for the loopback behaviour to reflect on the split ports. For direct ports, the corresponding trunk internal loopback works as expected.

Example:

This example shows how to configure internal loopback on a coherent DSP controller.

```
RP/0/RP0/CPU0:ios#configure
Fri Jul 8 10:42:51.329 UTC
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/0
RP/0/RP0/CPU0:ios(config-CoDSP)#secondary-admin-state maintenance
RP/0/RP0/CPU0:ios(config-CoDSP)#loopback internal
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
Fri Jul 8 10:43:48.644 UTC
RP/0/RP0/CPU0:ios(config-CoDSP)#end
```

This example shows how to verify the internal loopback configured on a coherent DSP controller.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/0
Fri Jul 8 10:45:53.820 UTC
Port : CoherentDSP 0/0/0/0
Controller State : Down
Inherited Secondary State : Normal
Configured Secondary State : Maintenance
Derived State : Maintenance
Loopback mode : Internal
BER Thresholds : SF = 1.0E-5 SD = 1.0E-7
Performance Monitoring : Enable
Bandwidth : 400.0Gb/s
Alarm Information:
LOS = 2 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0 SF_BER = 0
SD_BER = 0 BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0 FLEXO_GIDM = 0
FLEXO-MM = 0 FLEXO-LOM = 0 FLEXO-RDI = 0
FLEXO-LOF = 0
Detected Alarms : LOS
Bit Error Rate Information
PREFEC BER : 5.00E-01
POSTFEC BER : 0.00E+00
Q-Factor : 0.00 dB
Q-Margin : 0.00dB
OTU TTI Received
FEC mode : C_FEC
Flexo-Mode : Enable
Flexo Details:
Tx GID : 0
Rx GID : 0
AINS Soak : None
AINS Timer : 0h, 0m
AINS remaining time : 0 seconds
```

Example:

This example shows how to configure line loopback on a coherent DSP controller.

```
RP/0/RP0/CPU0:ios#configure
Fri Jul 8 10:48:48.577 UTC
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/0
RP/0/RP0/CPU0:ios(config-CoDSP)#secondary-admin-state maintenance
RP/0/RP0/CPU0:ios(config-CoDSP)#loopback line
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

```
Fri Jul 8 10:49:26.809 UTC
RP/0/RP0/CPU0:ios(config-CoDSP)#end
```

This example shows how to verify the line loopback configured on a coherent DSP controller.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/0
Fri Jul 8 10:49:44.073 UTC
Port : CoherentDSP 0/0/0/0
Controller State : Down
Inherited Secondary State : Normal
Configured Secondary State : Maintenance
Derived State : Maintenance
Loopback mode : Line
BER Thresholds : SF = 1.0E-5 SD = 1.0E-7
Performance Monitoring : Enable
Bandwidth : 400.0Gb/s
Alarm Information:
LOS = 2 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0 SF_BER = 0
SD_BER = 0 BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0 FLEXO_GIDM = 0
FLEXO-MM = 0 FLEXO-LOM = 0 FLEXO-RDI = 0
FLEXO-LOF = 0
Detected Alarms : LOS
Bit Error Rate Information
PREFEC BER : 5.00E-01
POSTFEC BER : 0.00E+00
Q-Factor : 0.00 dB
Q-Margin : 0.00dB
OTU TTI Received
FEC mode : C_FEC
Flexo-Mode : Enable
Flexo Details:
Tx GID : 0
Rx GID : 0
AINS Soak : None
AINS Timer : 0h, 0m
AINS remaining time : 0 seconds
```

Example:

This example shows how to configure internal loopback on the 400GE controller.

```
RP/0/RP0/CPU0:ios#configure
Fri Jul 8 11:19:26.286 UTC
RP/0/RP0/CPU0:ios(config)#controller FourHundredGigECtrlr 0/0/0/3
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#loopback internal
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
Fri Jul 8 11:19:47.496 UTC
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#end
```

This example shows how to verify the internal loopback configured on the 400GE controller.

```
RP/0/RP0/CPU0:ios#show controllers FourHundredGigECtrlr 0/0/0/3
Fri Jul 8 11:19:59.597 UTC
Operational data for interface FourHundredGigECtrlr0/0/0/3:
State:
Administrative state: enabled
Operational state: Down (Reason: State undefined)
LED state: Red On
Maintenance: Enabled
AINS Soak: None
```

```
Total Duration: 0 hour(s) 0 minute(s)
Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
Laser Squelch: Disabled
Insert Idle Ingress: Disabled
Insert Idle Egress: Disabled
Phy:
Media type: Not known
Alarms:
Current:
Loss of Signal
Statistics:
FEC:
Corrected Codeword Count: 702710
Uncorrected Codeword Count: 1147
Autonegotiation disabled.
Operational values:
Speed: 400Gbps
Duplex: Full Duplex
Flowcontrol: None
Loopback: Internal
BER monitoring:
Not supported
Forward error correction: Standard (Reed-Solomon)
Holdoff Time: 0ms
```

Example:

This example shows how to configure line loopback on the 4X100GE MXP.

```
RP/0/RP0/CPU0:ios(config)#controller hundredGigECtrlr 0/3/0/1/1
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#loopback line
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

This example shows how to verify the line loopback configured on the 4X100GE MXP.

```
RP/0/RP0/CPU0:ios#sh controllers hundredGigECtrlr 0/3/0/1/1
Fri Jul 22 10:34:39.730 UTC
Operational data for interface HundredGigECtrlr0/3/0/1/1:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Enabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled

Phy:
    Media type: Not known
    Statistics:
        FEC:
            Corrected Codeword Count: 6110368                Valid: True       Start time: 13:10:41
 Thu Jul 21 2022
            Uncorrected Codeword Count: 2771                 Valid: True       Start time: 13:10:41
 Thu Jul 21 2022
        PCS:
            Total BIP errors: 63700992                       Valid: True       Start time: 13:10:41
 Thu Jul 21 2022
            Total frame errors: 0                            Valid: False      Start time: 13:10:41
 Thu Jul 21 2022
            Total Bad SH: 0                                  Valid: False      Start time: 13:10:41
```

```
 Thu Jul 21 2022

Autonegotiation disabled.

Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: Line
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Example:

This example shows how to configure internal loopback on the 4X100GE MXP.

```
RP/0/RP0/CPU0:ios#conf
RP/0/RP0/CPU0:ios(config)#controller hundredGigECtrlr 0/3/0/7/1
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#loopback internal
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

This example shows how to verify the internal loopback configured on the 4X100GE MXP.

```
RP/0/RP0/CPU0:ios#show controller HundredGigECtrlr 0/3/0/7/1
Fri Jul 22 10:40:34.928 UTC

Operational data for interface HundredGigECtrlr0/3/0/7/1:

State:
    Administrative state: enabled
    Operational state: Down (Reason: State undefined)
    LED state: Red On
    Maintenance: Enabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled

Phy:
    Media type: Not known
    Alarms:
        Current:
            Loss of Signal
    Statistics:
        FEC:
            Corrected Codeword Count: 31426046
            Uncorrected Codeword Count: 2187

Autonegotiation disabled.

Operational values:
    Speed: 100Gbps
    Duplex: Full Duplex
    Flowcontrol: None
    Loopback: Internal
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Step 3 Run the **show alarms brief system active** command to view the loopback configuration alarms.

Example:

```
RP/0/RP0/CPU0:ios#show alarms brief system active
Tue Sep 13 17:43:35.212 UTC

------------------------------------------------------------------------------------
Active Alarms
------------------------------------------------------------------------------------
Location        Severity     Group            Set Time                    Description


------------------------------------------------------------------------------------


                                                                        0/2             Minor
  Controller       09/13/2022 17:34:32 UTC    HundredGigECtrlr0/2/0/2 - Internal Loopback Configured
0/2             Minor        Controller       09/13/2022 17:34:32 UTC    HundredGigECtrlr0/2/0/2 -
Internal Loopback Configured
0/2             Minor        Controller       09/13/2022 17:34:32 UTC    HundredGigECtrlr0/2/0/2 -
Line Loopback Configured

0/2             Major        Ethernet         09/13/2022 17:34:31 UTC    HundredGigECtrlr0/2/0/1/2
- Loss of Synchronization The Data Interface

0/2            Minor       Controller       09/13/2022 17:39:19 UTC    CoherentDSP0/2/0/0 - Internal
 Loopback Configured
```

Loop-and-drop mechanism on Ethernet controller

Table 31: Feature History

Feature Name	Release Information	Description
Loop-and-drop mechanism on Ethernet controller	Cisco IOS XR Release 24.4.1	

Feature Name	Release Information	Description
		The internal and line loopback configurations on the Ethernet controller have been enhanced to activate a loop-and-drop mechanism. This mechanism prevents traffic from flowing beyond the client port to the near end router in case of internal loopback, and far end router in case of the line loopback. This loop-and-drop mechanism is implemented by: • Internal loopback: Shutting down the transmit power at the client Ethernet controller. • Line loopback: Inserting a local fault alarm signal toward the trunk. This feature is supported on these cards: • NCS1K14-2.4T-K9 • NCS1K14-2.4T-X-K9 • NCS1K4-QXP-K9 • NCS1K4-1.2T-K9 (supports only internal loop-and-drop) • NCS1K4-1.2TL-K9 (supports only internal loop-and-drop) This enhancement isolates traffic during loopback, preventing it from reaching the router. A practical use case for this is when using the PRBS pattern for testing. By effectively containing the PRBS traffic, the mechanism ensures that PRBS traffic does not cause unwanted events such as error bits, which could otherwise lead to misinterpretation of network status and trigger false alarms at the router ports.

From Release 24.4.1, configuring client loopback on Ethernet controllers activates a loop-and-drop mechanism, replacing the previous loop-and-continue mechanism. When you set up internal and line loopbacks under Ethernet controllers, these loopback types are automatically applied for both 100GE and 400GE traffic.

- Internal loop-and-drop
- Line loop-and-drop

Advantage

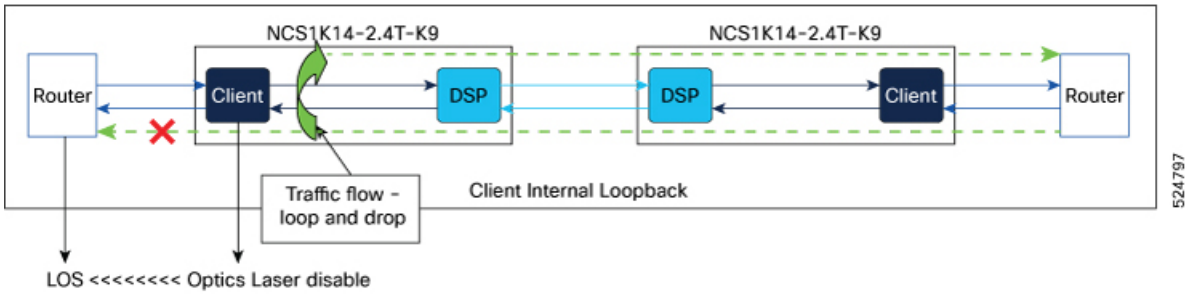
This enhancement provides a significant advantage by isolating traffic during loopback operations. It prevents traffic from reaching the router, thereby avoiding potential undesired events and alarms at the router ports. This is particularly beneficial in maintaining network stability and integrity during diagnostic activities.

A practical example of this advantage is seen when using the Pseudo-Random Bit Sequence (PRBS) pattern to evaluate network link performance and reliability. By effectively containing PRBS traffic within the loopback, the mechanism prevents unwanted events, such as error bits, from reaching the router. This containment ensures that test data does not interfere with the router's regular functions, avoiding misinterpretation of network status and preventing false alarms at the router ports.

Internal loop-and-drop

When you apply an internal loopback on a client Ethernet controller, the transmit power is disabled. This mechanism is applicable for both breakout and non-breakout setups. In a breakout setup, power is turned off on the specific lane where the loopback is configured, whereas in a non-breakout setup, power is shut down on all the lanes. This power shutdown prevents traffic from traveling beyond the client port to the near-end router as indicated by the cross mark in this image.

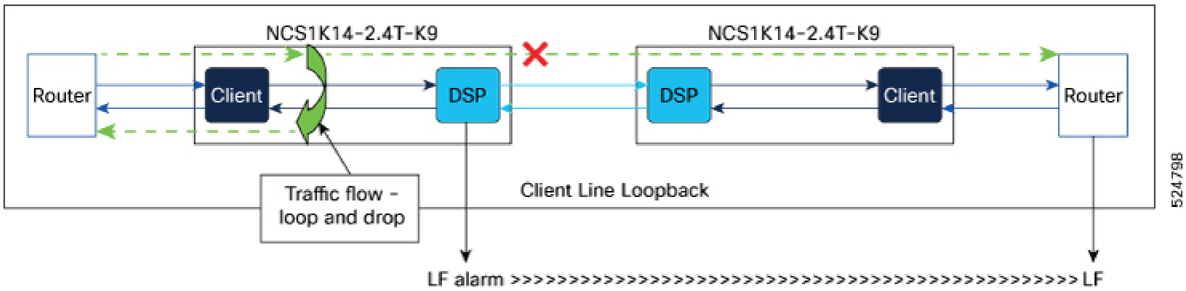
Figure 7: Internal loop-and-drop



Line loop-and-drop

When you apply a line loopback on a specific client Ethernet controller, the system forces a local fault (LF) signal on the near-end in the egress direction. This LF signal then propagates towards the far-end, preventing the traffic from flowing towards the far-end router as indicated by the cross mark in this image.

Figure 8: Line loop-and-drop



Supported cards

This mechanism is supported on these cards:

- NCS1K14-2.4T-K9
- NCS1K14-2.4T-X-K9
- NCS1K4-QXP-K9
- NCS1K4-1.2T-K9 (supports only internal loop-and-drop)
- NCS1K4-1.2TL-K9 (supports only internal loop-and-drop)

Restore factory settings

Note Perform this operation only on the console port.

Use this task to restore NCS 1014 to factory settings.

You can perform this operation only through the console port and not on the management interface.

Procedure

Run the **commit replace** command.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#commit replace
Tue Sep 24 09:36:59.430 UTC

This commit will replace or remove the entire running configuration. This
operation can be service affecting.
Do you wish to proceed? [no]: yes
RP/0/RP0/CPU0:ios(config)#exit
RP/0/RP0/CPU0:ios#reload
Tue Sep 24 09:38:12.881 UTC
```

```
Standby card not present or not Ready for failover. Proceed? [confirm]

Preparing system for backup. This may take a few minutes especially for large configurations.
        Status report: node0_RP0_CPU0: BACKUP INPROGRESS

        Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
[Done]

Proceed with reload?  [confirm]
Reloading node  0/RP0/CPU0

RL: Reboot initiated with code 1, cause User initiated graceful reload reboot_timeout 30 shutdown
delay 0
RL: Shutdown initiated
Query the node to be reloaded
 NODE_IP of noded to be reloaded 198.51.100.1
sending stop hb
Cause: User initiated graceful reload
VM IP addr sent for reload 198.51.100.1

Received ack from sdrmgr for reload request.Returncode:0
successful disconnection from service
wd_disconnect_cb 548 CMP-WD disconnected successfully
Invmgr successful disconnection from service

RP/0/RP0/CPU0:ios#
Disconnecting from 'default-sdr--1' console. Continue(Y/N)?


Connecting to 'default-sdr--1' console
ÿûÿûÿûÿýbootlogd: ioctl(/dev/pts/2, TIOCCONS): Device or resource busy
/sbin/restorecon:  lstat(/etc/adjtime) failed:  No such file or directory
Configuring network interfaces... done.
Starting system message bus: dbus.
Starting OpenBSD Secure Shell server: sshd
sshd start/running, process 1739
Starting rpcbind daemon...done.
Starting random number generator daemonUnable to open file: /dev/tpm0
.
Starting system log daemon...0
Starting kernel log daemon...0
tftpd-hpa disabled in /etc/default/tftpd-hpa
Starting internet superserver: xinetd.
net.ipv4.ip_forward = 1
Libvirt not initialized for container instance
Starting crond: OK
SIOCADDRT: File exists

DBG_MSG: platform type is 0
[*] ima_policy have loaded, or IMA policy file does not exist
Start serial incoming on , Clearing ..
RP/0/RP0/CPU0:Sep 24 09:38:44.284 UTC: fpd-serv[256]: %PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR
:FPD-NEED-UPGRADE :DECLARE :0/PM0:

This (D)RP Node is not ready or active for login /configuration
.......
........
........

ios con0/RP0/CPU0 is now available

Press RETURN to get started.
```

```
!!!!!!!!!!!!!!!!!!! NO root-system username is configured. Need to configure root-system username.
 !!!!!!!!!!!!!!!!!!!
```

After the **commit replace** operation completes, you must perform the IOS XR reload operation.

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.

Headless mode

A headless mode is a system operation mode that

- enables Cisco NCS 1014 to continue forwarding traffic when the controller card is absent, failed, or being restarted
- allows fault propagation for client and trunk ports, and
- maintains errorless operation unless there is any line-level activity or the traffic is encrypted and the keys have expired beyond a predefined maximum duration.

During headless mode, the control plane will be unavailable. You cannot provision new configurations, view operational data, or perform performance monitoring without a functional controller.

Trail Trace Identifier

A trail trace identifier (TTI) is a feature that:

- helps identify a signal from its source to its destination within a network,
- supports configuration of sent or expected identifiers in both ASCII and HEX formats, and
- triggers a controller shutdown and a TIM alarm when an expected TTI string does not match the received trace string.

Configure Trail Trace Identifier

Use this task to configure TTI on the coherent DSP controllers of a 1.2T, 2.4T, or 2.4TX card.

Procedure

Step 1 Run the **controller coherentDSP** *R/S/I/P* **tti** {**sent | expected**} **ascii | hex***tti-string* command.

Note
The *tti-string* can have a maximum of 64 characters for ASCII and 128 characters for HEX.

Example:

This sample shows how to configure TTI on a coherent DSP controller with the sent and expected strings set to the same ASCII string. The state of the controller is up.

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#controller coherentDSP
RP/0/RP0/CPU0:ios(config-CoDSP)#tti sent ascii hello
RP/0/RP0/CPU0:ios(config-CoDSP)#tti expected ascii hello cisco
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
Thu Dec 7 14:25:43.391 IST
RP/0/RP0/CPU0:ios(config-CoDSP)#end
```

This is a sample to view the TTI details on a coherentDSP controller.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/3/0/0
Thu Dec 7 14:26:37.345 IST

Port : CoherentDSP 0/3/0/0
Controller State : Down
Inherited Secondary State : Normal
Configured Secondary State : Normal
Derived State : In Service
Loopback mode : None
BER Thresholds : SF = 1.0E-5 SD = 1.0E-7
Performance Monitoring : Enable
Bandwidth : 800.0Gb/s

Alarm Information:
LOS = 0 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0 SF_BER = 0
SD_BER = 0 BDI = 0 TIM = 1
FECMISMATCH = 0 FEC-UNC = 0 FLEXO_GIDM = 0
FLEXO-MM = 0 FLEXO-LOM = 0 FLEXO-RDI = 0
FLEXO-LOF = 0
Detected Alarms : TIM

Bit Error Rate Information
PREFEC BER : 2.57E-04
POSTFEC BER : 0.00E+00
Q-Factor : 10.80 dB

Q-Margin : 4.50 dB

Instantaneous Q-Margin : 4.50 dB

OTU TTI Sent

FULL TTI ASCII STRING : hello
OTU TTI Received
FULL TTI ASCII STRING : hello
OTU TTI Expected
FULL TTI ASCII STRING : hello cisco

FEC mode : Soft-Decision 15

Flexo-Mode : Enable
Flexo Details:
Tx GID : 1
TX IID : 1, 2, 3, 4, 5, 6, 7, 8,
Rx GID : 1
RX IID : 1, 2, 3, 4, 5, 6, 7, 8,

AINS Soak : None
AINS Timer : 0h, 0m
AINS remaining time : 0 seconds
```

This sample shows how to configure TTI on a coherent DSP controller with the sent and expected strings set to HEX strings.

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ne(config)#controller coherentDSP 0/0/0/0
RP/0/RP0/CPU0:ne(config-CoDSP)#tti sent hex 6E6E6E2A2A2A
RP/0/RP0/CPU0:ne(config-CoDSP)#tti  expected hex 3F4B4B4B3D3E3A
RP/0/RP0/CPU0:ne(config-CoDSP)#commit
RP/0/RP0/CPU0:ios(config)#exit
```

Step 2 Run the **commit** command to apply the changes.

Configure Trail Trace Identifier on QXP card

Use this task to configure the TTI sent or expected string in the full ASCII format, or Source Access Point Identifier (SAPI)/Destination Access Point Identifier (DAPI) format on ODU-flex, ODU4, and coherentDSP controllers for the QXP card.

Note TTI operates only in **trunk mode OR**.

This table lists the ASCII format that is supported for TTI:

ASCII with character string	Controller
Full ASCII 64-character	CoherentDSP,odu4,odu-flex
SAPI ASCII 15-character	CoherentDSP,odu4,odu-flex
DAPI ASCII 15-character	CoherentDSP,odu4,odu-flex
Operator-specific ASCII 32-character	CoherentDSP,odu4,odu-flex

Procedure

Step 1 Run the **controller** *controller-type R/S/I/P* **tti** {**sent | expected**} {**ascii | sapi ascii | dapi ascii | operator-specific ascii** } *tti-string* to configure TTI.

Example:

This is a sample configuration for FULL TTI for coherentDSP controller.

```
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/8
RP/0/RP0/CPU0:ios(config-CoDSP)#tti sent ascii cisco
```

```
RP/0/RP0/CPU0:ios(config-CoDSP)#tti  expected ascii cisco123
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

This is a sample configuration for TTI HEX for coherentDSP controller.

```
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/8
RP/0/RP0/CPU0:ios(config-CoDSP)#tti sent hex 6E6E6E2A2A2A
RP/0/RP0/CPU0:ios(config-CoDSP)#tti  expected hex 3F4B4B4B3D3E3A
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

This is a sample configuration for Operator specific TTI for coherentDSP controller.

```
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/8
RP/0/RP0/CPU0:ios(config-CoDSP)#tti sent operator-specific ascii hellooo
RP/0/RP0/CPU0:ios(config-CoDSP)#tti expected operator-specific ascii hellooo
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

This is a sample configuration for Operator specific TTI HEX for coherentDSP controller.

```
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/8
RP/0/RP0/CPU0:ios(config-CoDSP)#tti  sent operator-specific hex 6E6E6E2A2A2A3D3E3A3A6E6E6E2A2A2A3D
RP/0/RP0/CPU0:ios(config-CoDSP)#tti  expected operator-specific hex 5A5A6D3A3B3C3F4B4B4B3D3E3A
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

This a sample configuration for SAPI for coherentDSP controller.

```
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/8
RP/0/RP0/CPU0:ios(config-CoDSP)#tti sent operator-specific ascii hellooo
RP/0/RP0/CPU0:ios(config-CoDSP)#tti expected operator-specific ascii hellooo
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

This is a sample configuration for DAPI for coherentDSP controller.

```
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/8
RP/0/RP0/CPU0:ios(config-CoDSP)#tti sent dapi ascii cisco123
RP/0/RP0/CPU0:ios(config-CoDSP)#tti  expected dapi ascii hello
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

Step 2 Run the **commit** command to apply the changes.

Configure chromatic dispersion

Use this task to configure chromatic dispersion on optics controllers of 1.2T, 2.4TX, and 2.4T cards.

When you configure the maximum and minimum values for chromatic dispersion for any data rate, ensure the minimum difference between the configured values is equal to or greater than 1500 ps/nm.

This table lists the default CD search range for a 1.2T card.

Data rate	BPS	Card support	Default CD search range
200G to 500G	BPS < = 3	1.2T, 1.2TL	-10,000 to 100,000 ps/nm
	3 < BPS <= 4	1.2T, 1.2TL	-10,000 to 80,000 ps/nm
	4 < BPS <=5	1.2T	-5,000 to 20,000 ps/nm

Data rate	BPS	Card support	Default CD search range
600G	BPS=5.2578125	1.2T	-2000 to 2,000 ps/nm

This table lists the default CD search range for 2.4T, 2.4TX, cards.

Line rate				
	138GBd	128GBd	118GBd	108GBd
1200G	4000 to -5000	4000 to -5000	—	—
1000G	20000 to -5000	20000 to -5000	20000 to -5000	—
800G	48000 to -10000	48000 to -10000	48000 to -10000	35000 to -1000
600G	74000 to -10000	74000 to -10000	74000 to -10000	60000 to -1000
500G	90000 to -10000	90000 to -10000	90000 to -10000	72000 to -1000
400G	—	90000 to -10000	90000 to -10000	72000 to -1000

Note The cd-min and cd-max values must be set for BPS values that are greater than 4 in the 1.2T card.

Note When the user provisions the cd-min and cd-max values that are outside the range through CLI, the provisioned values are accepted; however, only the actual values supported by the hardware are applied.

Procedure

Step 1 Enter the `controller optics` command in the global configuration mode.

Example:

This is a sample where chromatic dispersion is configured on the optics controller.

```
RP/0/RP0/CPU0:ios#configure
Mon Aug 19 19:31:42.115 UTC
RP/0/RP0/CPU0:ios(config)#controller  optics 0/1/0/1
RP/0/RP0/CPU0:ios(config-Optics)#cd-max 4000
RP/0/RP0/CPU0:ios(config-Optics)#cd-min -1000
RP/0/RP0/CPU0:ios(config-Optics)#commit
Mon Aug 19 19:35:24.697 UTC
RP/0/RP0/CPU0:ios(config-Optics)#exit
RP/0/RP0/CPU0:ios(config)#exit


RP/0/RP0/CPU0:ios#show run controller optics 0/1/0/*
Mon Aug 19 19:57:41.859 UTC
controller Optics0/1/0/0
 transmit-power -15
 dwdm-carrier 50GHz-grid itu-ch 55
```

```
 enh-sop-tol-mode 1
 cross-pol-gain-mode 10
 lbc-high-threshold 5
!
controller Optics0/1/0/1
 description trunk power UP
 cd-min -1000
 cd-max 4000
 enh-colorless-mode 2
 enh-sop-tol-mode 3
 nleq-comp-mode 4
 cross-pol-gain-mode 2
 cross-pol-weight-mode 3
 cpr-win-mode 3
 cpr-ext-win-mode 8
 rx-voa fixed-ratio 1200
 filter-roll-off-factor 0.035
!
controller Optics0/1/0/5
 soak-time 10
!
```

Step 2 Run the **commit** command to apply the changes.

Configure chromatic dispersion threshold

Use this task to configure the minimum and maximum acceptable chromatic dispersion for the trunk optics controllers.

The CD alarm is raised if the chromatic dispersion goes below the minimum or exceeds the maximum value.

Procedure

Step 1 Enter the **controller optics***R/S/I/P***cd-high-threshold***cd-high***cd-low-threshold***cd-low** command in the global configuration mode to configure the minimum and maximum chromatic dispersion threshold.

Example:

This sample configures the maximum and minimum acceptable CD on the controller optics:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/7
RP/0/RP0/CPU0:ios(config-Optics)#cd-high-threshold 2400
RP/0/RP0/CPU0:ios(config-Optics)#cd-low-threshold -2400
RP/0/RP0/CPU0:ios(config-Optics)#commit
```

This sample shows the maximum and minimum acceptable CD configured on the controller optics:

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/7
Fri Nov 12 10:58:50.595 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Yellow
Optics Status
         Optics Type:  CIM8 DWDM
```

```
        DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
        Wavelength=1552.524nm

        Alarm Status:
        -------------
        Detected Alarms:
                HIGH-RX-PWR   LOW-TX-PWR
                HIGH-DGD

        LOS/LOL/Fault Status:

        Alarm Statistics:

        -------------
        HIGH-RX-PWR = 1            LOW-RX-PWR = 0
        HIGH-TX-PWR = 0            LOW-TX-PWR = 1
        HIGH-LBC = 0               HIGH-DGD = 6
        OOR-CD = 0                 OSNR = 0
        WVL-OOL = 0                MEA  = 0
        IMPROPER-REM = 1
        TX-POWER-PROV-MISMATCH = 0
        Laser Bias Current = 0.0 %
        Actual TX Power = 0.97 dBm
        RX Power = -0.53 dBm
        RX Signal Power = -1.20 dBm
        Frequency Offset = 63 MHz

        Performance Monitoring: Enable

        THRESHOLD VALUES
        ----------------

        Parameter                High Alarm  Low Alarm  High Warning  Low Warning
        -----------------------  ----------  ---------  ------------  -----------
        Rx Power Threshold(dBm)         -2.0       -3.0           0.0          0.0
        Tx Power Threshold(dBm)          4.0        2.0           0.0          0.0
        LBC Threshold(mA)                N/A        N/A          0.00         0.00

        LBC High Threshold = 55 %
        Configured Tx Power = 1.00 dBm
        Configured CD High Threshold = 2400 ps/nm
        Configured CD lower Threshold = -2400 ps/nm
        Configured OSNR lower Threshold = 0.40 dB
        Configured DGD Higher Threshold = 0.30 ps
        Baud Rate =  63.1394679230 GBd
        Bits per Symbol = 3.0000000000  bits/symbol
        Modulation Type: 8QAM
        Chromatic Dispersion 0 ps/nm
        Configured CD-MIN -48000 ps/nm  CD-MAX 48000 ps/nm
        Polarization Mode Dispersion = 0.0 ps
        Second Order Polarization Mode Dispersion = 29.00 ps^2
        Optical Signal to Noise Ratio = 36.10 dB
        SNR = 17.50 dB
        Polarization Dependent Loss = 0.50 dB
        Polarization Change Rate = 0.00 rad/s
        Differential Group Delay = 1.00 ps

Transceiver Vendor Details

        Form Factor            : CIM8
        Name                   : CISCO-ACACIA
        Part Number            : 10-3500-01
        Rev Number             : 01
        Serial Number          : ACA24480037
```

```
        PID                   : CIM8-C-K9
        VID                   : VES1
        Date Code(yy/mm/dd)   : 23/11/10
        Fiber Connector Type: LC
        Otn Application Code: Not Set
        Sonet Application Code: Not Set
        Ethernet Compliance Code: Not set

Transceiver Temperature : 46 Celsius
AINS Soak               : None
AINS Timer              : 0h, 0m
AINS remaining time     : 0 seconds
```

Step 2 Run the **commit** command to apply the changes.

Configure Transmit Power

Use this task to configure transmit power on the trunk (CIM8 and PICO) optics within the -190 to +50 range (in units of 0.1dBm)

Procedure

Enter the `controller optics` command in the global configuration mode.

Example:

```
RP/0/RP0/CPU0:ios#configure
Mon Aug 19 19:31:42.115 UTC
RP/0/RP0/CPU0:ios(config)#controller optics 0/1/0/7
RP/0/RP0/CPU0:ios(config-Optics)#transmit-power -1.50
RP/0/RP0/CPU0:ios(config-Optics)#commit
Mon Aug 19 19:35:24.697 UTC
RP/0/RP0/CPU0:ios(config-Optics)#exit
RP/0/RP0/CPU0:ios(config)#exit
```

This is a sample in which transmit power of -1.50 dBm is configured on the CIM8 optics.

```
RP/0/RP0/CPU0:ios#show  controllers optics 0/0/0/7
 Controller State: Up
 Transport Admin State: Automatic In Service
 Laser State: On
 LED State: Green
 Optics Status
        Optics Type:  CIM8 DWDM
        DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
        Wavelength=1552.524nm
        Alarm Status:
        -------------
        Detected Alarms: None
        LOS/LOL/Fault Status:
        Alarm Statistics:
        HIGH-RX-PWR = 0            LOW-RX-PWR = 2
        HIGH-TX-PWR = 0            LOW-TX-PWR = 1
        HIGH-LBC = 0               HIGH-DGD = 0
        OOR-CD = 0                 OSNR = 1
```

```
WVL-OOL = 0                MEA  = 0
IMPROPER-REM = 0
TX-POWER-PROV-MISMATCH = 0
Laser Bias Current = 0.0 %
Actual TX Power = 1.98 dBm
RX Power = -0.68 dBm
RX Signal Power = -1.25 dBm
Frequency Offset = -41 MHz
Performance Monitoring: Enable
THRESHOLD VALUES
Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
------------------------  ----------  ---------  ------------  -----------
Rx Power Threshold(dBm)         13.0      -17.0           0.0          0.0
Tx Power Threshold(dBm)          5.0      -13.0           0.0          0.0
LBC Threshold(mA)                N/A        N/A          0.00         0.00
LBC High Threshold = 90 %
Configured Tx Power = 2.00 dBm
Configured CD High Threshold = 180000 ps/nm
Configured CD lower Threshold = -180000 ps/nm
Configured OSNR lower Threshold = 20.50 dB
Configured DGD Higher Threshold = 90.00 ps
Baud Rate =  137.9783940000 GBd
Bits per Symbol = 3.5200000000  bits/symbol
Modulation Type: PCS
Chromatic Dispersion -1 ps/nm
Configured CD-MIN -10000 ps/nm  CD-MAX 48000 ps/nm
Polarization Mode Dispersion = 0.0 ps
Second Order Polarization Mode Dispersion = 13.00 ps^2
Optical Signal to Noise Ratio = 38.10 dB
SNR = 17.40 dB
Polarization Dependent Loss = 2.50 dB
Polarization Change Rate = 0.00 rad/s
Differential Group Delay = 1.00 ps
Filter Roll Off Factor : 0.100
Rx VOA Target Power : -2.0 dBm
NLEQ Compensation Mode : 0
Cross Polarization Gain Mode : 10
Proprietary Submarine Parameters
  Type : 1        Value : 0
  Type : 2        Value : 0
  Type : 3        Value : 0
  Type : 4        Value : 0
  Type : 5        Value : 0
  Type : 6        Value : 1000
  Type : 7        Value : 0
  Type : 8        Value : 0
  Type : 9        Value : 0
  Type : 10       Value : 0
```

Configure laser bias current high threshold

Use this task to laser bias current high threshold.

Procedure

Step 1 Run the **controller optics***R/S/I/P* **lbc-high-threshold***lbc-value* command to configure the laser bias current threshold.

Example:

This is a sample which configures the high laser bias threshold on the controller optics.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/7
RP/0/RP0/CPU0:ios(config-Optics)#lbc-high-threshold 55
RP/0/RP0/CPU0:ios(config-Optics)#commit
```

This is a sample which shows the high rlaser bias threshold configured on the controller optics.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/7
Fri Nov 12 10:58:50.595 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Yellow
Optics Status
         Optics Type:  CIM8 DWDM
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm

         Alarm Status:
         -------------
         Detected Alarms:
                 HIGH-RX-PWR   LOW-TX-PWR
                 HIGH-DGD

         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 1            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 1
         HIGH-LBC = 0               HIGH-DGD = 6
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 1
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = 0.97 dBm
         RX Power = -0.53 dBm
         RX Signal Power = -1.20 dBm
         Frequency Offset = 63 MHz

         Performance Monitoring: Enable

         THRESHOLD VALUES
         ----------------

         Parameter                High Alarm  Low Alarm  High Warning  Low Warning
         -----------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         -2.0       -3.0           0.0          0.0
         Tx Power Threshold(dBm)          4.0        2.0           0.0          0.0
         LBC Threshold(mA)                N/A        N/A          0.00         0.00

         LBC High Threshold = 55 %
         Configured Tx Power = 1.00 dBm
         Configured CD High Threshold = 2400 ps/nm
         Configured CD lower Threshold = -2400 ps/nm
         Configured OSNR lower Threshold = 0.40 dB
         Configured DGD Higher Threshold = 0.30 ps
         Baud Rate =  63.1394679230 GBd
         Bits per Symbol = 3.0000000000  bits/symbol
```

```
          Modulation Type: 8QAM
          Chromatic Dispersion 0 ps/nm
          Configured CD-MIN -48000 ps/nm  CD-MAX 48000 ps/nm
          Polarization Mode Dispersion = 0.0 ps
          Second Order Polarization Mode Dispersion = 29.00 ps^2
          Optical Signal to Noise Ratio = 36.10 dB
          SNR = 17.50 dB
          Polarization Dependent Loss = 0.50 dB
          Polarization Change Rate = 0.00 rad/s
          Differential Group Delay = 1.00 ps

Transceiver Vendor Details

          Form Factor            : CIM8
          Name                   : CISCO-ACACIA
          Part Number            : 10-3500-01
          Rev Number             : 01
          Serial Number          : ACA24480037
          PID                    : CIM8-C-K9
          VID                    : VES1
          Date Code(yy/mm/dd)    : 23/11/10
          Fiber Connector Type: LC
          Otn Application Code: Not Set
          Sonet Application Code: Not Set
          Ethernet Compliance Code: Not set

Transceiver Temperature : 46 Celsius
AINS Soak               : None
AINS Timer              : 0h, 0m
AINS remaining time     : 0 seconds
```

Step 2 Run the **commit** command to apply the changes.

Configure differential group delay threshold

Use this task to configure the threshold value for the maximum acceptable differential group delay (DGD) on the trunk optics controllers.

- The DGD alarm is raised if DGD exceeds this value.
- The range is 0–18000 (in the units of 0.01 ps).

Procedure

Step 1 Run the **controller optics***R/S/I/P***dgd-high-threshold***dgd-value* command to configure to configure the maximum acceptable DGD.

Example:

This sample configures the minimum acceptable DGD on the controller optics.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/7
RP/0/RP0/CPU0:ios(config-Optics)#dgd-high-threshold 30
RP/0/RP0/CPU0:ios(config-Optics)#commit
```

This sample shows the maximum acceptable DGD configured on the controller optics.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/7
Fri Nov 12 10:58:50.595 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Yellow
Optics Status
         Optics Type:  CIM8 DWDM
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm

         Alarm Status:
         -------------
         Detected Alarms:
                 HIGH-RX-PWR   LOW-TX-PWR
                 HIGH-DGD

         LOS/LOL/Fault Status:

         Alarm Statistics:

         -------------
         HIGH-RX-PWR = 1            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 1
         HIGH-LBC = 0               HIGH-DGD = 6
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 1
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = 0.97 dBm
         RX Power = -0.53 dBm
         RX Signal Power = -1.20 dBm
         Frequency Offset = 63 MHz

         Performance Monitoring: Enable

         THRESHOLD VALUES
         ----------------

         Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
         ------------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         -2.0       -3.0           0.0          0.0
         Tx Power Threshold(dBm)          4.0        2.0           0.0          0.0
         LBC Threshold(mA)                N/A        N/A          0.00         0.00

         LBC High Threshold = 55 %
         Configured Tx Power = 1.00 dBm
         Configured CD High Threshold = 2400 ps/nm
         Configured CD lower Threshold = -2400 ps/nm
         Configured OSNR lower Threshold = 0.40 dB
         Configured DGD Higher Threshold = 0.30 ps
         Baud Rate =  63.1394679230 GBd
         Bits per Symbol = 3.0000000000  bits/symbol
         Modulation Type: 8QAM
         Chromatic Dispersion 0 ps/nm
         Configured CD-MIN -48000 ps/nm  CD-MAX 48000 ps/nm
         Polarization Mode Dispersion = 0.0 ps
         Second Order Polarization Mode Dispersion = 29.00 ps^2
         Optical Signal to Noise Ratio = 36.10 dB
         SNR = 17.50 dB
```

```
         Polarization Dependent Loss = 0.50 dB
         Polarization Change Rate = 0.00 rad/s
         Differential Group Delay = 1.00 ps

Transceiver Vendor Details

         Form Factor            : CIM8
         Name                   : CISCO-ACACIA
         Part Number            : 10-3500-01
         Rev Number             : 01
         Serial Number          : ACA24480037
         PID                    : CIM8-C-K9
         VID                    : VES1
         Date Code(yy/mm/dd)    : 23/11/10
         Fiber Connector Type: LC
         Otn Application Code: Not Set
         Sonet Application Code: Not Set
         Ethernet Compliance Code: Not set

Transceiver Temperature : 46 Celsius

AINS Soak               : None
AINS Timer              : 0h, 0m
AINS remaining time     : 0 seconds
```

Step 2 Run the **commit** command to apply the changes.

Configure optical signal to noise ratio

Use this task to configure the minimum acceptable Optical Signal to Noise ratio (OSNR) value on the 1.2T, 2.4TX, and 2.4T cards.

The OSNR alarm is raised if OSNR goes below this value.

The range is 0–4000 (in units of 0.01db).

Procedure

Step 1 Enter the **controller optics***R/S/I/P***osnr-low-threshold***osnr-valuecd-low* command to configure the minimum acceptable OSNR.

Example:

This sample configures the minimum acceptable OSNR on the controller optics.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/7
RP/0/RP0/CPU0:ios(config-Optics)#osnr-low-threshold 40
RP/0/RP0/CPU0:ios(config-Optics)#commit
```

This sample shows the minimum acceptable OSNR configured on the controller optics.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/7
Fri Nov 12 10:58:50.595 UTC
Controller State: Up
```

```
Transport Admin State: In Service
Laser State: On
LED State: Yellow
Optics Status
         Optics Type:  CIM8 DWDM
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm

         Alarm Status:
         -------------
         Detected Alarms:
                 HIGH-RX-PWR   LOW-TX-PWR
                 HIGH-DGD

         LOS/LOL/Fault Status:
         Alarm Statistics:
         -------------
         HIGH-RX-PWR = 1            LOW-RX-PWR = 0
         HIGH-TX-PWR = 0            LOW-TX-PWR = 1
         HIGH-LBC = 0               HIGH-DGD = 6
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 1
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = 0.97 dBm
         RX Power = -0.53 dBm
         RX Signal Power = -1.20 dBm
         Frequency Offset = 63 MHz

         Performance Monitoring: Enable

         THRESHOLD VALUES
         ----------------

         Parameter                High Alarm  Low Alarm  High Warning  Low Warning
         -----------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         -2.0       -3.0           0.0          0.0
         Tx Power Threshold(dBm)          4.0        2.0           0.0          0.0
         LBC Threshold(mA)                N/A        N/A          0.00         0.00

         LBC High Threshold = 55 %
         Configured Tx Power = 1.00 dBm
         Configured CD High Threshold = 2400 ps/nm
         Configured CD lower Threshold = -2400 ps/nm
         Configured OSNR lower Threshold = 0.40 dB
         Configured DGD Higher Threshold = 0.30 ps
         Baud Rate =  63.1394679230 GBd
         Bits per Symbol = 3.0000000000  bits/symbol
         Modulation Type: 8QAM
         Chromatic Dispersion 0 ps/nm
         Configured CD-MIN -48000 ps/nm  CD-MAX 48000 ps/nm
         Polarization Mode Dispersion = 0.0 ps
         Second Order Polarization Mode Dispersion = 29.00 ps^2
         Optical Signal to Noise Ratio = 36.10 dB
         SNR = 17.50 dB
         Polarization Dependent Loss = 0.50 dB
         Polarization Change Rate = 0.00 rad/s
         Differential Group Delay = 1.00 ps
Transceiver Vendor Details
         Form Factor            : CIM8
         Name                   : CISCO-ACACIA
         Part Number            : 10-3500-01
         Rev Number             : 01
```

```
          Serial Number          : ACA24480037
          PID                    : CIM8-C-K9
          VID                    : VES1
          Date Code(yy/mm/dd)    : 23/11/10
          Fiber Connector Type: LC
          Otn Application Code: Not Set
          Sonet Application Code: Not Set
          Ethernet Compliance Code: Not set

Transceiver Temperature : 46 Celsius
AINS Soak               : None
AINS Timer              : 0h, 0m
AINS remaining time     : 0 seconds
```

Step 2 Run the **commit** command to apply the changes.

Configure receive power threshold

Use this task to configure the high and low threshold of the total optical signal power of the received signal on the 1.2T, 2.4TX, and 2.4T cards.

The range is –400 to 300 (in the units of 0.1 dBm).

Procedure

Step 1 Enter the **rx-high-threshold***rx-high***rx-low-threshold***rx-low**R/S/I/P* command to configure the high and low receive power threshold.

Example:

This sample configures the high receive power threshold on the controller optics.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/7
RP/0/RP0/CPU0:ios(config-Optics)#rx-high-threshold -20
RP/0/RP0/CPU0:ios(config-Optics)#rx-low-threshold -30
RP/0/RP0/CPU0:ios(config-Optics)#commit
```

This sample shows the high receive power threshold configured on the controller optics.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/7
Fri Nov 12 10:58:50.595 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Yellow
Optics Status
          Optics Type:  CIM8 DWDM
          DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
          Wavelength=1552.524nm

          Alarm Status:
          -------------
          Detected Alarms:
                  HIGH-RX-PWR   LOW-TX-PWR
```

```
        HIGH-DGD

LOS/LOL/Fault Status:

Alarm Statistics:

-------------
HIGH-RX-PWR = 1            LOW-RX-PWR = 0
HIGH-TX-PWR = 0            LOW-TX-PWR = 1
HIGH-LBC = 0               HIGH-DGD = 6
OOR-CD = 0                 OSNR = 0
WVL-OOL = 0                MEA  = 0
IMPROPER-REM = 1
TX-POWER-PROV-MISMATCH = 0
Laser Bias Current = 0.0 %
Actual TX Power = 0.97 dBm
RX Power = -0.53 dBm
RX Signal Power = -1.20 dBm
Frequency Offset = 63 MHz

Performance Monitoring: Enable

THRESHOLD VALUES
----------------

Parameter                High Alarm  Low Alarm  High Warning  Low Warning
-----------------------  ----------  ---------  ------------  -----------
Rx Power Threshold(dBm)         -2.0       -3.0           0.0          0.0
Tx Power Threshold(dBm)          4.0        2.0           0.0          0.0
LBC Threshold(mA)                N/A        N/A          0.00         0.00

LBC High Threshold = 55 %
Configured Tx Power = 1.00 dBm
Configured CD High Threshold = 2400 ps/nm
Configured CD lower Threshold = -2400 ps/nm
Configured OSNR lower Threshold = 0.40 dB
Configured DGD Higher Threshold = 0.30 ps
Baud Rate =  63.1394679230 GBd
Bits per Symbol = 3.0000000000  bits/symbol
Modulation Type: 8QAM
Chromatic Dispersion 0 ps/nm
Configured CD-MIN -48000 ps/nm  CD-MAX 48000 ps/nm
Polarization Mode Dispersion = 0.0 ps
Second Order Polarization Mode Dispersion = 29.00 ps^2
Optical Signal to Noise Ratio = 36.10 dB
SNR = 17.50 dB
Polarization Dependent Loss = 0.50 dB
Polarization Change Rate = 0.00 rad/s
Differential Group Delay = 1.00 ps

Transceiver Vendor Details

        Form Factor            : CIM8
        Name                   : CISCO-ACACIA
        Part Number            : 10-3500-01
        Rev Number             : 01
        Serial Number          : ACA24480037
        PID                    : CIM8-C-K9
        VID                    : VES1
        Date Code(yy/mm/dd)    : 23/11/10
        Fiber Connector Type: LC
        Otn Application Code: Not Set
        Sonet Application Code: Not Set
        Ethernet Compliance Code: Not set
```

```
Transceiver Temperature : 46 Celsius
AINS Soak               : None
AINS Timer              : 0h, 0m
AINS remaining time     : 0 seconds
```

Step 2 Run the **commit** command to apply the changes.

Frequency

Use this task to configure the frequency on trunk ports of the transponder line cards.

This table lists the frequency range with grid spacing supported on the line card.

Line card	Frequency range (THz)	Grid spacing
1.2T	191.25 to 196.1	50GHz and 100MHz
1.2TL [4]	186.1 to 190.85	100MHz
2.4T	191.25 to 196.1	50GHz and 100MHz
2.4TX using C band trunk pluggables	191.25 to 196.1	50GHz and 100MHz
2.4TX using L band trunk pluggables	186.15 to 190.925	100MHz
QXP	191.275 to 196.125	50GHz and 100MHz

[4] Only non-ITU channels are supported

Procedure

Step 1 Run the **controller optics***Rack/Slot/Instance/Port***dwdm-carrier** {**100MHz-grid frequency** *frequency*} | {**50GHz-grid** [**frequency** *frequency*} command to configure the wavelength.

Step 2 Run the **commit** command to apply the changes.

Change the trunk port frequency on a transponder card

Use this task when you need to modify the optical signal frequency on the trunk port on the transponder cards.

Procedure

Step 1 Run the **config** command to enter the global configuration mode.

Step 2 Run the **shutdown** command in optics configuration mode to shutdown the trunk port.

Example:

```
RP/0/RP0/CPU0:ios(config)#controller optics 0/2/0/0
RP/0/RP0/CPU0:ios(config-Optics)#shutdown
RP/0/RP0/CPU0:ios(config-Optics)#commit
Fri Jul 11 07:19:30.624 UTC
RP/0/RP0/CPU0:ios(config-Optics)#
```

The interface at trunk port 0 is shut down.

Step 3 Run the **dwdm-carrier 100MHz-grid frequency** <*frequency*> command to set the new frequency.

Example:

```
RP/0/RP0/CPU0:ios(config-Optics)#dwdm-carrier 100MHz-grid frequency 1911500
Fri Jul 11 07:20:17.234 UTC
WARNING! Changing channel/wavelength/frequency can impact traffic
RP/0/RP0/CPU0:ios(config-Optics)#commit
Fri Jul 11 07:20:19.905 UTC
RP/0/RP0/CPU0:ios(config-Optics)#
```

Note
Changing the trunk port frequency can impact traffic.

The frequency for the 100 MHz grid channel is set to 1911500.

Step 4 Run the **no shutdown** command in optics configuration mode to bring the port back up.

Example:

```
RP/0/RP0/CPU0:ios(config-Optics)#no shutdown
RP/0/RP0/CPU0:ios(config-Optics)#commit
Fri Jul 11 07:20:57.198 UTC
RP/0/RP0/CPU0:ios(config-Optics)#
```

Note
If you configure the frequency and use the no shutdown command in the same batch, the no shutdown command is processed first, followed by the new frequency. This means the trunk interface transmits at the previous frequency for a shorter time.

The channel resumes operation at the new frequency. The channel on the trunk port now operates at the configured frequency.

Baud rate

You can configure the baud rate on trunk ports of the 2.4T and 2.4TX line cards.

During baud rate configuration, any changes or configurations must differ by at least 0.5 from the current baud rate. For example, if the current baud rate on the node is 130.525, the new configuration must be set to a value greater than 131.

This table lists the baud range supported on the line card.

Line rate	Min BR [GBd]	Min BR [GBd]	Approx.max net bpS	Approx.min net bpS	Default baud rate
400G	62	128	4.1	2.1	127.931418
500G	62	138	5	2.5	137.834059
600G	72	138	5.1	2.8	137.738007
700G	88	138	5	3.2	138.081665
800G	98	138	5.1	3.5	137.978388
900G	108	138	5.2	3.8	137.898168
1000G	118	138	5.3	4.3	137.834059
1100G	128	138	5.3	4.7	137.78165
1200G	128	138	5.7	5.3	137.738007

CCMD-16 controllers

The NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards have two types of controllers.

- OCH controllers
- OMS controllers

When the NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards become operational, both the OMS and OCH controllers are automatically created by default.

View OCH controller and parameters

Use this task to view an OCH contollers and parameters.

When you bring up the nodes with NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards, the OCH controllers are automatically created by default. The OCH controllers for the NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards are:

Table 32: Supported interfaces

Card	Port type	OCH ports	Interfaces
NCS1K14-CCMD-16-C	LC Ports	16	och R/S/I/1 to och R/S/I/16

Card	Port type	OCH ports	Interfaces
NCS1K14-CCMD-16-L	LC Ports	16	och R/S/I/1 to och R/S/I/16

Note R/S/I/P stands for Rack/Slot/Instance/Port.

Procedure

Step 1 Run the **show controllers och** command to view the OCH controllers.

Example:

This output shows the active OCH controllers in *Slot 0* from *port 1 to 16*.

Output Example

```
0/0/0/1               Och Interface Instance
  0/0/0/10              Och Interface Instance
  0/0/0/11              Och Interface Instance
  0/0/0/12              Och Interface Instance
  0/0/0/13              Och Interface Instance
  0/0/0/14              Och Interface Instance
  0/0/0/15              Och Interface Instance
  0/0/0/16              Och Interface Instance
  0/0/0/2               Och Interface Instance
  0/0/0/3               Och Interface Instance
  0/0/0/4               Och Interface Instance
  0/0/0/5               Och Interface Instance
  0/0/0/6               Och Interface Instance
  0/0/0/7               Och Interface Instance
  0/0/0/8               Och Interface Instance
  0/0/0/9               Och Interface Instance
```

Step 2 Run the **show controllers** command to view the parameters of an OCH controller.

Example:

This output shows the parameters of the OCH controller.

Output Example

```
RP/0/RP0/CPU0:ios#show controllers och 0/1/0/1
Thu Oct 12 09:26:38.555 UTC

 Controller State: Up

 Transport Admin State: In Service

 LED State: Green

         Alarm Status:
         -------------
         Detected Alarms: None
```

```
Alarm Statistics:
-----------------
RX-LOS-P = 0
TX-POWER-FAIL-LOW = 0

Parameter Statistics:
---------------------
Tx Power = -2.30 dBm
Rx Power = -0.70 dBm
```

View OMS controller and parameters

Use this task to view an OMS controller and parameters.

When you bring up the nodes with NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards, the OMS controllers are automatically created by default.

The OMS controllers for the NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards are:

Table 33: Supported interfaces

Card	Port type	OMS ports	Interfaces
NCS1K14-CCMD-16-C	LC Ports	1	oms R/S/I/0
NCS1K14-CCMD-16-L	LC Ports	1	oms R/S/I/0

Note R/S/I/P stands for Rack/Slot/Instance/Port.

Procedure

Step 1 Run the **show controllers oms ?** command to view the OMS controllers.

Example:

This output shows the active OMS controller in *Slot 0*.

Step 2 Run the **show controllers** command to view the parameters of an OMS controller.

Example:

This output shows the parameters of the an OMS controller.

Output Example

```
RP/0/RP0/CPU0:ios#show controllers oms 0/1/0/0
Thu Oct 12 09:23:35.297 UTC
 Controller State: Up
 Transport Admin State: In Service
 LED State: Red
         Alarm Status:
```

```
-------------
Detected Alarms:
        HI-TX-BR-PWR

Alarm Statistics:
-----------------
RX-LOS-P = 3
TX-POWER-FAIL-LOW = 0
INGRESS-AMPLI-GAIN-LOW = 0
INGRESS-AMPLI-GAIN-HIGH = 0
EGRESS-AUTO-LASER-SHUT = 0
EGRESS-AMPLI-GAIN-LOW = 0
EGRESS-AMPLI-GAIN-HIGH = 0
HI-TX-BR-PWR = 2


Parameter Statistics:
---------------------
Tx Power = -3.40 dBm
Rx Power = -2.30 dBm
Rx Voa Attenuation = 5.0 dB
Tx Voa Attenuation = 5.0 dB
Ingress Ampli Mode = Gain
Ingress Ampli Gain = 5.0 dB
Ingress Ampli Tilt = 0.0 dB
Ingress Ampli OSRI = OFF
Egress Ampli Mode = Gain
Egress Ampli Gain = 2.0 dB
Egress Ampli Tilt = 0.0 dB
Egress Ampli Gain Range = Normal
Egress Ampli OSRI = OFF
Egress Ampli BR Power = -12.90 dBm
Egress Ampli BR Ratio = -9.39 dB


Configured Parameters:
-------------
Rx Voa Attenuation = 5.0 dB
Tx Voa Attenuation = 5.0 dB
Ingress Ampli Mode = Gain
Ingress Ampli Gain = 5.0 dB
Ingress Ampli Power = -9.0 dBm
Ingress Ampli OSRI = OFF
Egress Ampli Mode = Gain
Egress Ampli Gain = 2.0 dB
Egress Ampli Power = 0.0 dBm
Egress Ampli Gain Range = Normal
Egress Ampli OSRI = OFF
BR High Threshold = -17.0 dBm
```

Configure Controller Parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L Cards

Table 34: Feature History

Feature Name	Release Information	Feature Description
Supported Functionalities of CCMD-16-C and CCMD-16-L Line Cards	Cisco IOS XR Release 7.11.1	Supported Functionalities of CCMD-16-C and CCMD-16-L Line Cards: The software supports Variable Optical Attenuator (VoA), power monitoring and reporting of parameters to the controllers at the OCH and OMS level. It helps in configuring the amplifier parameters for optimizing signal transmissions. The software also supports in-band and out-of-band tone detection and monitoring and reporting of alarms.

This chapter describes the controller configuration using EDFA, VoA, optical safety, and photodiode parameters which are supported on NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L line cards.

Table 35: Basic Controller Configuration Parameters

Parameter	CLI	Description
EDFA-ADD Control Mode	`controller Oms R/S/I/P` `egress-ampli-mode power-control`	Here the default mode is gain-control. Use this command to change the mode to power-control mode. You can delete this configuration to revert to gain-control mode.
EDFA-ADD Power	`controller Oms R/S/I/P` `egress-ampli-power <val in 0.1 dBm>`	This is the target output power configuration for the EDFA located on COM-Tx. If the output power is not configured, then a default value is used by the system. You can find the applied target output power by using the **show controller oms** output.

EDFA-ADD Gain Range	`controller Oms R/S/I/P` `egress-ampli-gain-range <normal | extended>`	

		If you change the gain range from Normal to Extended or the opposite way, without updating the proper gain value for the new gain range, then the following may happen:
		• The EDFA switches to the preconfigured or default value of the gain causing a mismatch between the operational and configured gain.
		• The gain configuration is lost during the reload of software or line card, as the configured gain mismatches with the latest gain-range. This may result in traffic interruption during these reload operations.
		Hence, we recommend that you explicitly configure the gain range mode as normal or extended, and the corresponding gain values for each mode to get the expected results.
		The following are a few example scenarios that may not work as expected:
		• Scenario 1:
		Current running config:
		`controller Oms R/S/I/P`
		` egress-ampli-gain-range extended`
		` egress-ampli-gain <gain value in extended mode>`
		New applied config:
		`controller Oms R/S/I/P`
		` no egress-ampli-gain-range extended`
		` commit`
		• Scenario 2:
		Current running config:
		`controller Oms R/S/I/P`
		` egress-ampli-gain-range extended`
		` egress-ampli-gain <gain value in extended mode>`
		New applied config:
		`controller Oms R/S/I/P`
		` egress-ampli-gain-range Normal`
		` commit`
		• Scenario 3:
		Current running config:
		`controller Oms R/S/I/P`
		`egress-ampli-gain <gain value in normal mode>`

		New applied config:
		`controller Oms R/S/I/P`
		`egress-ampli-gain-range extended`
		`commit`
		The following is another example scenario that involves commit-replace command where you replace the existing gain configuration that does not have explicitly configured gain-range, with new gain-range and gain value.
		Scenario 4:
		Current running config:
		`controller Oms R/S/I/P`
		`egress-ampli-gain <gain value in normal mode>`
		New applied config:
		`……..`
		`controller Oms R/S/I/P`
		`egress-ampli-gain-range extended`
		`egress-ampli-gain <gain value in extended mode>`
		`commit replace`

EDFA-ADD Gain	`controller Oms R/S/I/P` `egress-ampli-gain <val in 0.1 dB>`	This is used for configuring gain for the EDFA located on the COM-Tx port. If the gain is not configured, then a default value is used by the system. The **show controller oms** output can be used for finding the applied gain.
EDFA-DROP Control Mode	`controller Oms R/S/I/P` `ingress-ampli-mode power-control`	The default mode is gain-control. Use this command to change the mode to power-control mode. You can delete this configuration to revert to gain-control mode.
EDFA-DROP Power	`controller Oms R/S/I/P` `ingress-ampli-power <vale in 0.1 dBm>`	This is the target output power configuration for the EDFA located on COM-Rx. If the output power is not configured, then a default value is used by the system. You can find the applied target output power by using the **show controller oms** output.
EDFA-DROP Gain	`controller Oms R/S/I/P` `ingress-ampli-gain <val in 0.1 dB>`	This is used for configuring gain for the EDFA located on the COM-Rx port. If the gain is not configured, then a default value is used by the system. The **show controller oms** output can be used for finding the applied gain.

VoA Attenuation	`controller Oms R/S/I/P`
	`tx-voa-attenuation <val in 0.1 dB>`
	`rx-voa-attenuation <val in 01. dB>`
		This is used to configure VoA attenuation for COM-Tx and COM-Rx ports. The system picks a default value if the gain is not configured. The **show controller oms** output can be used for finding the applied attenuation.

Shutdown (COM Port)	`controller Oms R/S/I/P` `shutdown`	COM-Tx and Rx ports are put in Out of Service (OOS) when this command is executed. This turns off the EDFA devices on both COM-Tx and COM-Rx ports and the respective alarms are masked. Here the default value is 'unshut'.
Shutdown (Ch Ports)	`controller Och R/S/I/P` `shutdown`	The channel port will be marked as out-of-service.

Configure Operational Parameters

The different operational parameters supported are given below.

Table 36: Operational Parameters

Parameter	CLI	Description
Photo Diodes (COM Port)	**OMS Controller** `Tx Power and Rx Power`	Rx Power = Inband power + OOB Tx Power = Inband power + OOB It reports the power transmitted and received on the OMS port. The OOB power received on COM-Rx is looped back to COM-Tx with some loss due to the insertion-loss on the loop back path.
Photo Diodes (CH Ports)	**OCH Controller** `Tx Power and Rx Power`	It reports the power transmitted and received on the channel ports.

Amplifier Parameters	**OMS Controller**
	`Ingress Ampli Mode`
	`Ingress Ampli Gain`
	`Ingress Ampli Tilt`
	`Ingress Ampli OSRI`
	`Egress Ampli Mode`
	`Egress Ampli Gain`
	`Egress Ampli Tilt`
	`Egress Ampli Gain Range`
	`Egress Ampli OSRI`
	`Egress Ampli BR Power`
	`Egress Ampli BR Ratio`
		Egress and ingress amplifier parameters.

VoA Parameters	**OMS Controller** `Tx VoA Attenuation and Rx VoA Attenuation`	VoA attenuation parameters.

Configure Optical Safety Parameters

Use the following parameters for configuring optical safety.

Table 37: Optical Safety Parameters

Parameter	CLI	Description

OSRI	`controller Oms R/S/I/P`
	`egress-ampli-osri`
	`ingress-ampli-osri`
		Use this configuration to enable or Disable Optical Safety Remote Interlock (OSRI) on the amplifiers located on COM-Tx and COM-Rx ports. The default value is 'Off'. The **show controller oms** output can be used for finding the OSRI configuration and status.

Verify Rx VOA fixed ratio to zero

Use this task to configure the Rx VOA attenuation to a fixed ratio of zero, effectively disabling the Rx VOA entirely.

Warning Disabling Rx VOA attenuation in certain scenarios may destabilize the network. Practice caution and contact the Cisco TAC team for more information.

Procedure

Run the **configure** command to enter the global configuration mode.

Example:

This is a sample configuration which sets the rx-voa fixed-ratio to zero.

```
RP/0/RP0/CPU0:NE80#configur
Tue Aug 6 07:44:28.125 UTC
Current Configuration Session  Line       User     Date                     Lock
00001000-00001569-00000000 vty0 cisco Tue Aug 6 05:33:16 2024
RP/0/RP0/CPU0:NE80(config)#controller optics 0/2/0/0 rx-voa fixed-ratio 0
RP/0/RP0/CPU0:NE80(config)#controller optics 0/2/0/7 rx-voa fixed-ratio 0
RP/0/RP0/CPU0:NE80(config)#commit
Tue Aug 6 07:44:48.545 UTC
RP/0/RP0/CPU0:NE80(config)#end
```

Verify Rx VOA fixed ratio to zero

Use this task to verify the Rx VOA attenuation to a fixed ratio of zero, effectively disabling the Rx VOA entirely.



Warning Disabling Rx VOA attenuation in certain scenarios may destabilize the network. Practice caution and contact the Cisco TAC team for more information.

Procedure

Step 1 Run the `show running-config controller` command to verify the rx-voa fixed ratio is set to fixed ratio zero.

Example:

```
RP/0/RP0/CPU0:N112#show running-config controller optics 0/2/0/0
Thu May 16 09:19:51.223 UTC
description optics0/2/0/0
dwdm-carrier 100MHz-grid frequency 1909000
rx-voa fixed-ratio 0
```

Step 2 Run the `show controllers optics` command to verify whether the rx-voa fixed-ratio is set to fixed ratio zero.

Example:

```
RP/0/RP0/CPU0:123#show controllers optics 0/2/0/0
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Green
Optics Status
Optics Type: CIM8 DWDM
DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
Wavelength=1552.524nm
Alarm Status:
-------------
Detected Alarms: None
LOS/LOL/Fault Status:
Alarm Statistics:
-------------
HIGH-RX-PWR = 0 LOW-RX-PWR = 0
HIGH-TX-PWR = 0 LOW-TX-PWR = 0
HIGH-LBC = 0 HIGH-DGD = 0
OOR-CD = 0 OSNR = 0
WVL-OOL = 0 MEA = 0
IMPROPER-REM = 0
TX-POWER-PROV-MISMATCH = 0
Laser Bias Current = 0.0 %
Actual TX Power = 2.00 dBm
RX Power = -0.01 dBm
RX Signal Power = 1.94 dBm
Frequency Offset = 0 MHz
Performance Monitoring: Enable
THRESHOLD VALUES
----------------
Parameter High Alarm Low Alarm High Warning Low Warning
----------------------- ---------- --------- ------------ -----------
```

```
Rx Power Threshold(dBm) 13.0 -17.0 0.0 0.0
Tx Power Threshold(dBm) 5.0 -13.0 0.0 0.0
LBC Threshold(mA) N/A N/A 0.00 0.00
LBC High Threshold = 90 %
Configured Tx Power = 2.00 dBm
Configured CD High Threshold = 180000 ps/nm
Configured CD lower Threshold = -180000 ps/nm
Configured OSNR lower Threshold = 20.40 dB
Configured DGD Higher Threshold = 88.00 ps
Baud Rate = 137.9783940000 GBd
Bits per Symbol = 3.5200000000 bits/symbol
Modulation Type: PCS
Chromatic Dispersion 0 ps/nm
Configured CD-MIN -10000 ps/nm CD-MAX 48000 ps/nm
Polarization Mode Dispersion = 0.0 ps
Second Order Polarization Mode Dispersion = 44.00 ps^2
Optical Signal to Noise Ratio = 36.60 dB
SNR = 18.20 dB
Polarization Dependent Loss = 0.40 dB
Polarization Change Rate = 4639.00 rad/s
Differential Group Delay = 2.00 ps
Filter Roll Off Factor : 0.100
Rx VOA Fixed Ratio : 0.00 dB
NLEQ Compensation Mode : 0
Cross Polarization Gain Mode : 10
Proprietary Submarine Parameters
Type : 1 Value : 0
Type : 2 Value : 0
Type : 3 Value : 0
Type : 4 Value : 0
Type : 5 Value : 10485760
Transceiver Vendor Details
Form Factor : CIM8
Name : ACACIA
Part Number : 10-100508-01
Rev Number : 10
Serial Number : N/A
PID : CIM8-CE-K9
VID : VES1
Firmware Version : Major.Minor.Build
Active : 80.130.19
Inactive : 80.130.19
Date Code(yy/mm/dd) : 24/06/18
Fiber Connector Type: LC
Otn Application Code: Not Set
Sonet Application Code: Not Set
Ethernet Compliance Code: Not set
Transceiver Temperature : 36 Celsius
AINS Soak : None
AINS Timer : 0h, 0m
AINS remaining time : 0 seconds
```

Last link flap detection

Table 38: Feature History

Feature Name	Release Information	Feature Description
Last Link Flap Detection for Transponder Controllers	Cisco IOS XR Release 24.4.1	This feature introduces the detection of the last link flap for various transponder controllers. It provides the elapsed time since the occurrence of the last flap for each controller in the `show controllers` output. Link flap refers to the state transition of controllers from UP state to another state or from other states to UP state. This information enables network administrators to effectively monitor and troubleshoot signal stability and performance across the network. Parameter added: • **Last link flapped***value*

Link flap refers to the state transition of controllers from UP state to another state or from other states to UP state. The common cause is related to bad, unsupported, or nonstandard cable or Small Form-Factor Pluggable (SFP) or related to other link synchronization issues. The cause for link flapping can be intermittent or permanent.

NCS 1014 node detects the flapping of link in a signal for various transponder controllers. It provides the time that is elapsed from the occurrence of the last link flap through the *last link flapped* parameter in the **show controllers** output.

The **show controller** output displays the time-elapsed value for the last link flap in seconds, minutes, hours, days, weeks, and years based on the occurrence of the event.

Examples of last link flap value	Meaning of the value
00:00:10	Last link flapped 10 seconds ago.
1d12h	Last link flapped one day and 12 hours ago.
1y00w	Last link flapped one year ago.

The transponder controllers supporting the last link flap parameters are:

- Ethernet
- CoherentDSP
- ODU
- OTU
- Optics

Output examples for each controller

Ethernet controller

The output provides information about a link flap on Ethernet controller that happened one year ago. It highlights the *Last link flapped* value in *1y00w* format for **show controllers TenGigECtrlr 0/1/0/0**.

```
RP/0/0/CPU0:ios#show controllers TenGigECtrlr 0/1/0/0
Fri Sep 27 14:38:16.645 IST
Operational data for interface TenGigECtrlr0/1/0/0:
State:
    Administrative state: enabled
    Operational state: Up
    Maintenance: Enabled
    AINS Soak: None
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled
    Last link flapped: 1y00w
```

Optics controller

The output provides information about a link flap on optics controller that happened four seconds ago. It highlights the *Last link flapped* value in the *HH:MM:SS* format for **show controllers Optics 0/2/0/2**.

```
RP/0/0/CPU0:ios#show controllers Optics 0/2/0/2
Thu Oct 10 12:39:37.767 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Green
Last link flapped: 00:00:04
         Optics not present
         Optics Type: Unavailable
        DWDM Carrier Info: Unavailable, MSA ITU Channel= Unavailable, Frequency= Unavailable
 , Wavelength= Unavailable
         TX Power = Unavailable
         RX Power = Unavailable
```

OTU controller

The output provides information about a link flap on OTU controller that happened ten seconds ago. It highlights the *Last link flapped* value in the *HH:MM:SS* format **show controllers OTU4 0/0/0/6**.

```
RP/0/0/CPU0:ios#show controllers OTU4 0/0/0/6
Thu Oct 10 12:57:13.777 UTC
Port                                        : OTU4 0/0/0/6
Controller State                            : Up
LED state                                   : Not Applicable
Last link flapped                           : 00:00:10
Inherited Secondary State                   : Normal
Configured Secondary State                  : Normal
Derived State                               : In Service
Loopback mode                               : None
BER Thresholds                              : SF = 1.0E-6  SD = 1.0E-7
Performance Monitoring                      : Enable
Alarm Information:
LOS = 0 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0     FLEXO_GIDM = 0
FLEXO-MM = 0    FLEXO-LOM = 0   FLEXO-RDI = 0
FLEXO-LOF = 0
Detected Alarms                             : None
```

ODU controller

The output provides information about a link flap on ODU controller that happened 52 seconds ago. It highlights the *Last link flapped* value in the *HH:MM:SS* format **show controllers ODU4 0/0/0/6**.

```
RP/0/0/CPU0:ios#show controllers ODU4 0/0/0/6
Thu Oct 10 12:57:55.717 UTC
Port                                        : ODU4 0/0/0/6
Controller State                            : Up
LED state                                   : Not Applicable
Last link flapped                           : 00:00:52
Inherited Secondary state                   : Normal
Configured Secondary state                  : Normal
Derived State                               : In Service
Loopback mode                               : None
BER Thresholds                              : SF = 1.0E-6  SD = 1.0E-7
Performance Monitoring                      : Disable
Path Monitoring Mode                        : Non-Intrusive Monitor
PM TIM-CA state                             : Disable
Alarm Information:
AIS = 0 IAE = 0 BIAE = 0
SF_BER = 0      SD_BER = 0      BDI = 0
OCI = 0 LCK = 0 PTIM = 0
TIM = 0 CSF = 0 GFP LFD = 0
GFP LOCS = 0    GFP LOCCS = 0   GFP UPM = 0
Detected Alarms                             : None
```

ODU-FLEX controller

The output provides information about a link flap on ODU-FLEX controller that happened one minute and 35 seconds ago. It highlights the *Last link flapped* value in the *HH:MM:SS* format **show controllers ODU-FLEX 0/3/0/0/*** .

```
RP/0/RP0/CPU0:ios#show controllers ODU-FLEX 0/3/0/0/*
Thu Oct 10 16:25:20.008 IST
Port                                        : ODU-FLEX 0/3/0/0/1
Controller State                            : Up
Last link flapped                           : 00:01:35
Inherited Secondary state                   : Normal
Configured Secondary state                  : Normal
Derived State                               : In Service
Loopback mode                               : None
BER Thresholds                              : SF = 1.0E-6  SD = 1.0E-7
Performance Monitoring                      : Enable
Path Monitoring Mode                        : Operational
PM TIM-CA state                             : Disable
```

CoherentDSP controller

The output provides information about a link flap on coherentDSP controller that happened one day and 12 hours ago. It highlights the *Last link flapped* value in *1d12h* format for **show controllers coherentDSP 0/2/0/0**.

```
RP/0/RP0/CPU0:ios#show  controllers coherentDSP 0/2/0/0
Tue Nov 26 06:20:08.673 UTC

Port                                        : CoherentDSP 0/2/0/0
Controller State                            : Up
Last link flapped                           : 1d12h

Inherited Secondary State                   : Normal
Configured Secondary State                  : Normal
Derived State                               : In Service
Loopback mode                               : None
BER Thresholds                              : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                      : Enable
```

```
Bandwidth                                          : 800.0Gb/s
Client Network mapping Type                        : Ethernet

Alarm Information:
LOS = 16        LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 2     FLEXO_GIDM = 0
FLEXO-MM = 0    FLEXO-LOM = 0   FLEXO-RDI = 15
FLEXO-LOF = 96
Detected Alarms                                    : None

Bit Error Rate Information
PREFEC  BER                                        : 1.52E-04
POSTFEC BER                                        : 0.00E+00
Q-Factor                                           : 11.20 dB

Q-Margin                                           : 4.80 dB

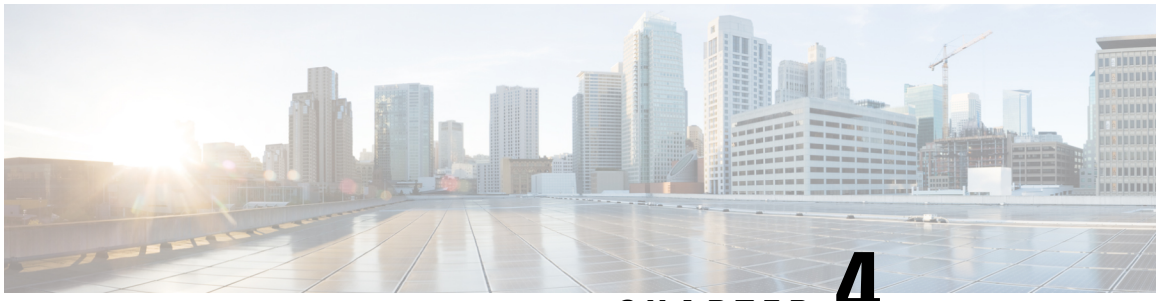
Instantaneous Q-Margin                             : 4.80 dB

TTI :
Remote hostname                          : Node126
Remote interface                         : CoherentDSP 0/2/0/0
Remote IP addr                           : 0.0.0.0

FEC mode                                           : Soft-Decision 15

Flexo-Mode                                         : Enable
Flexo Details:
Tx GID                                   : 1
TX IID                                   : 1, 2, 3, 4, 5, 6, 7, 8,
Rx GID                                   : 1
RX IID                                   : 1, 2, 3, 4, 5, 6, 7, 8,

AINS Soak                                          : None
AINS Timer                                         : 0h, 0m
AINS remaining time                                : 0 seconds
```



CHAPTER 4

Performance Monitoring

This chapter describes the procedures to configure and view the performance monitoring parameters.

- Performance monitoring , on page 131
- Performance monitoring for NCS1K14-2.4T-X-K9 card , on page 153
- Performance monitoring for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards , on page 153

Performance monitoring

Table 39: Feature History

Feature Name	Release Information	Feature Description
Performance Monitoring Enhancements	Cisco IOS XR Release 24.4.1	You can now view historical Performance Monitoring (PM) parameters for the past 1 to 7 days for the 24-hour interval on these controllers: • CoherentDSP • Ethernet • ODU-layer This feature enhances data collection directly from the equipment, providing a comprehensive view of performance over time. Parameter added: • **bucket** *value*

Performance monitoring (PM) parameters help service providers gather and store performance data, set thresholds, and report results. This process enables early detection of network issues.

You can configure and retrieve PM counters for various controllers in 30-second, 15-minute, or 24-hour flex-bin intervals.

These parameters simplify troubleshooting operations and enhance the data collected directly from the equipment.

Note Downgrade to 7.11.1 or earlier releases from 24.1.1 can lead to a restart of the pm_collector process. To avoid this, clear the PM historical data before the downgrade using these commands:

```
process shutdown pm_collector
                run
                cd /misc/config
                rm chkpt_pm_collector_*
                exit
```

Use these commands if you are already experiencing a continuous pm_collector restart.

```
process shutdown pm_collector
                run
                cd /misc/config
                rm chkpt_pm_collector_*
                exit
                process start pm_collector
```

Configure the PM parameters

Use this task to configure the performance monitoring (PM) parameters for the optics, Ethernet, odu-flex, and coherent DSP controllers.

Follow these steps to configure PM parameters:

Procedure

Run the **show controller** *controllertype R/S/I/P* { **pm** { **current | history**} { **30-sec 15-min** | | **24-hour** } { **optics** | **ether** | **fec** | **otn | prbs| flex-bin**} **linenumber** command to configure the controllers.

Example:

This sample configures the performance monitoring parameters for the optics controller at 24-hour intervals.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/1/5 pm 24-hour optics threshold osnr max
345
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:RINode1#configure
Tue Feb 25 17:52:54.320 IST
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 24-hour optics threshold opr min
-30
RP/0/RP0/CPU0:RINode1(config)#commit
```

Example:

This sample configures the performance monitoring parameters for the Ethernet controller at 15-minute intervals.

```
RP/0/RP0/CPU0:chassisA164(config)#controller fourHundredGigECtrlr 0/1/0/4 pm 15-min ether
threshold rx-pkt 1
RP/0/RP0/CPU0:RINode1#configure
Tue Feb 25 17:51:18.353 IST
```

```
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 15-min optics threshold opr min -30
RP/0/RP0/CPU0:RINode1(config)#commit
Tue Feb 25 17:52:02.504 IST
```

Example:

This sample configures the performance monitoring parameters for the Coherent DSP controller at 30-second intervals.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/7 pm 30-sec fec threshold post-fec-ber
max OE-15
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:RINode1#configur
Tue Feb 25 17:48:00.789 IST
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 ?
apply-group Apply configuration from a group
exclude-group Exclude apply-group configuration from a group
pm Configure pm parameters
rx-low-threshold Configure transponder low receive power threshold
sec-admin-state Configure the secondary admin state of och controller
shutdown Disable och controller processing
tone-detect-oob Configure tone detect oob
tone-pattern-expected Configure Tone Pattern Expected
tone-rate Configure bit rate
tx-low-threshold Configure transponder low transmit power threshold
<cr>
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm ?
15-min Configure pm parameters of 15 minute interval
24-hour Configure pm parameters of 24 hour interval
30-sec Configure pm parameters of 30 second interval
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 30-sec ?
optics Configure och optics layer performance monitoring
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 30-sec optics ?
report set och optics layer TCA reporting status
threshold Configure threshold on och optics layer parameters
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 30-sec optics threshold ?
opr set opr threshold in dBm
opt set opt threshold in dBm
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 30-sec optics threshold opr ?
max set opr max threshold in dBm
min set opr min threshold in dBm
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 30-sec optics threshold opr min ?
<-5000,+6633> Enter 4 digit value ;Input value 3000 for 30.00dbm
RP/0/RP0/CPU0:RINode1(config)#controller och 0/5/0/8 pm 30-sec optics threshold opr min -30
RP/0/RP0/CPU0:RINode1(config)#commit
Tue Feb 25 17:50:01.632 IST
```

View the PM parameters

Use this task to view the performance monitoring parameters for Optics, Ethernet, and Coherent DSP controllers.

Follow these steps to configure the PM parameters:

Procedure

Step 1 Run the **show controllers** *controllertype R/S/I/P* **{ pm { current | history } { 30 sec | 15-min | 24-hour } { optics | ether | fec | otn | prbs}** *linenumber* } command to display the current performance monitoring parameters of the controller with 15-minute intervals.

Example:

This sample displays the current performance monitoring parameters of the Optics controller with 15-minute intervals.

```
RP/0/RP0/CPU0:ios#show controller optics 0/1/0/3 pm current 15-min optics 3
Fri Sep 22 13:53:37.120 IST

Optics in the current interval [13:45:00 - 13:53:37 Fri Sep 22 2023]

Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                          Threshold(min)   Threshold(min)  (min) Threshold(max)
Threshold(max)  (max)
LBC[% ]      : 56.8      56.8      56.8     0.0              NA              NO   100.0
 NA              NO
OPT[dBm]     : -40.00    -40.00    -40.00   -30.00           NA              NO   63.32
 NA              NO
OPR[dBm]     : -40.00    -40.00    -40.00   -30.00           NA              NO   63.32
 NA              NO
```

```
RP/0/RP0/CPU0:RINode1#show controllers och 0/5/0/8 pm current 15-min optics 1
Tue Feb 25 17:55:28.915 IST

Optics in the current interval [17:45:00 - 17:55:28 Tue Feb 25 2025]

Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                          Threshold(min)   Threshold(min)  (min) Threshold(max)
Threshold(max)  (max)
OPT[dBm]     : -50.00    -50.00    -50.00   -30.00           NA              NO   15.00
 NA              NO
OPR[dBm]     : -2.59     -2.59     -2.59    -28.00           NA              NO   8.00
 NA              NO

Last clearing of "show controllers OPTICS" counters never
```

Example:

This sample displays the current performance monitoring parameters of the client Optics controller with 15-minute intervals.

```
RP/0/RP0/CPU0:ios#show controller optics 0/2/0/1 pm current 15-min optics 1
Fri Sep 22 13:56:52.123 IST

Optics in the current interval [13:45:00 - 13:56:52 Fri Sep 22 2023]

Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                          Threshold(min)   Threshold(min)  (min) Threshold(max)
Threshold(max)  (max)
LBC[% ]      : 24.8      25.7      26.7     0.0              NA              NO   100.0
 NA              NO
OPT[dBm]     : -0.12     -0.00     0.11     -30.00           NA              NO   63.32
```

```
 NA              NO
OPR[dBm]     : -0.67     -0.46     -0.24    -30.00            NA              NO    63.32
 NA              NO
```

Example:

This sample displays the current performance monitoring parameters of the client Ethernet controller with 15-minute intervals.

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4 pm current 15-min ether
ETHER in the current interval [16:15:00 - 16:18:44 Fri Nov 17 2023]
ETHER current bucket type : Valid
 RX-UTIL[%]                : 0.00               Threshold : 0.00         TCA(enable) : NO
 TX-UTIL[%]                : 0.00               Threshold : 0.00         TCA(enable) : NO
 RX-PKT                    : 0                  Threshold : 0            TCA(enable) : NO
 STAT-PKT                  : 0                  Threshold : 0            TCA(enable) : NO
 OCTET-STAT                : 0                  Threshold : 0            TCA(enable) : NO
 OVERSIZE-PKT              : 0                  Threshold : 0            TCA(enable) : NO
 FCS-ERR                   : 0                  Threshold : 0            TCA(enable) : NO
 LONG-FRAME                : 0                  Threshold : 0            TCA(enable) : NO
 JABBER-STATS              : 0                  Threshold : 0            TCA(enable) : NO
 64-OCTET                  : 0                  Threshold : 0            TCA(enable) : NO
 65-127-OCTET              : 0                  Threshold : 0            TCA(enable) : NO
 128-255-OCTET             : 0                  Threshold : 0            TCA(enable) : NO
 256-511-OCTET             : 0                  Threshold : 0            TCA(enable) : NO
 512-1023-OCTET            : 0                  Threshold : 0            TCA(enable) : NO
 1024-1518-OCTET           : 0                  Threshold : 0            TCA(enable) : NO
 IN-UCAST                  : 0                  Threshold : 0            TCA(enable) : NO
 IN-MCAST                  : 0                  Threshold : 0            TCA(enable) : NO
 IN-BCAST                  : 0                  Threshold : 0            TCA(enable) : NO
 OUT-UCAST                 : 0                  Threshold : 0            TCA(enable) : NO
 OUT-BCAST                 : 0                  Threshold : 0            TCA(enable) : NO
 OUT-MCAST                 : 0                  Threshold : 0            TCA(enable) : NO
 TX-PKT                    : 0                  Threshold : 0            TCA(enable) : NO
 OUT-OCTET                 : 0                  Threshold : 0            TCA(enable) : NO
 IFIN-ERRORS               : 0                  Threshold : 0            TCA(enable) : NO
 IFIN-OCTETS               : 0                  Threshold : 0            TCA(enable) : NO
 STAT-MULTICAST-PKT        : 0                  Threshold : 0            TCA(enable) : NO
 STAT-BROADCAST-PKT        : 0                  Threshold : 0            TCA(enable) : NO
 STAT-UNDERSIZED-PKT       : 0                  Threshold : 0            TCA(enable) : NO
 IN_GOOD_BYTES             : 0                  Threshold : 0            TCA(enable) : NO
 IN_GOOD_PKTS              : 0                  Threshold : 0            TCA(enable) : NO
 IN_DROP_OTHER             : 0                  Threshold : 0            TCA(enable) : NO
 OUT_GOOD_BYTES            : 0                  Threshold : 0            TCA(enable) : NO
 OUT_GOOD_PKTS             : 0                  Threshold : 0            TCA(enable) : NO
 IN_PKT_64_OCTET           : 0                  Threshold : 0            TCA(enable) : NO
 IN_PKTS_65_127_OCTETS     : 0                  Threshold : 0            TCA(enable) : NO
 IN_PKTS_128_255_OCTETS    : 0                  Threshold : 0            TCA(enable) : NO
 IN_PKTS_256_511_OCTETS    : 0                  Threshold : 0            TCA(enable) : NO
 IN_PKTS_512_1023_OCTETS   : 0                  Threshold : 0            TCA(enable) : NO
 IN_PKTS_1024_1518_OCTETS  : 0                  Threshold : 0            TCA(enable) : NO
 OUT_PKT_64_OCTET          : 0                  Threshold : 0            TCA(enable) : NO
 OUT_PKTS_65_127_OCTETS    : 0                  Threshold : 0            TCA(enable) : NO
 OUT_PKTS_128_255_OCTETS   : 0                  Threshold : 0            TCA(enable) : NO
 OUT_PKTS_256_511_OCTETS   : 0                  Threshold : 0            TCA(enable) : NO
 OUT_PKTS_512_1023_OCTETS  : 0                  Threshold : 0            TCA(enable) : NO
 OUT_PKTS_1024_1518_OCTETS : 0                  Threshold : 0            TCA(enable) : NO
 TX_UNDERSIZED_PKT         : 0                  Threshold : 0            TCA(enable) : NO
 TX_OVERSIZED_PKT          : 0                  Threshold : 0            TCA(enable) : NO
 TX_JABBER                 : 0                  Threshold : 0            TCA(enable) : NO
 TX_BAD_FCS                : 0                  Threshold : 0            TCA(enable) : NO
```

Example:

This sample displays the current performance monitoring for FEC on the Coherent DSP controller 15-minute intervals:

```
RP/0/RP0/CPU0:ios#show controller coherentDSP 0/2/0/0 pm current 15-min fec
Fri Sep 22 14:02:19.236 IST

g709 FEC in the current interval [14:00:00 - 14:02:19 Fri Sep 22 2023]

FEC current bucket type : Valid
    EC-BITS   : 545156378205            Threshold : 5400000000000          TCA(enable)  : YES
    UC-WORDS  : 0                       Threshold : 5                      TCA(enable)  : YES

                                      MIN       AVG        MAX      Threshold      TCA     Threshold
     TCA
                                                                        (min)      (enable)    (max)
    (enable)
PreFEC BER                     :   5.19E-03   5.36E-03   6.09E-03      0E-15        NO       0E-15
       NO
PostFEC BER                    :     0E-15     0E-15     0E-15      0E-15        NO       0E-15
    NO
Q[dB]                          :      8.10      8.10      8.10          0.00        NO        0.00
       NO
Q_Margin[dB]                   :      2.10      2.10      2.10          0.00        NO        0.00
       NO
Instantaneous Q_Margin [dB]    :      1.70      1.77      1.80          0.00        NO        0.00
       NO
```

```
RP/0/RP0/CPU0:RINode1#show controllers oms 0/5/0/33 pm current 30-sec optics 1
Tue Feb 25 17:56:39.462 IST

Optics in the current interval [17:56:30 - 17:56:39 Tue Feb 25 2025]

Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                         Threshold(min)   Threshold(min)  (min)  Threshold(max)
Threshold(max)  (max)
OPT[dBm]     : -7.90     -7.90     -7.90    -30.00            NA               NO   15.00
 NA              NO
OPR[dBm]     : -0.60     -0.60     -0.60    -28.00            NA               NO   8.00
 NA              NO

Last clearing of "show controllers OPTICS" counters never
```

Example:

This sample displays the current performance monitoring parameters for PRBS of the Coherent DSP controller with 15-minute intervals:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 pm current 15-min prbs
Mon Feb 13 00:58:48.327 UTC
PRBS in the current interval [00:45:00 - 00:58:48 Mon Feb 13 2019]
PRBS current bucket type : Valid
EBC : 40437528165
FOUND-COUNT : 1 FOUND-AT-TS : 00:51:22 Mon Feb 13 2019
LOST-COUNT : 1 LOST-AT-TS : 00:52:52 Mon Feb 13 2019
CONFIG-PTRN : PRBS_PATTERN_PN31
Last clearing of "show controllers OTU" counters never
```

Example:

This sample displays the current performance monitoring FEC parameters of the coherentDSP OTN with 15-minute intervals:

```
show controllers coherentDSP 0/0/0/7 pm current 15-min otn
Fri Nov 17 16:33:50.820 UTC
g709 OTN in the current interval [16:30:00 - 16:33:50 Fri Nov 17 2023]
OTN current bucket type : Valid
    ES-NE   : 0         Threshold : 500      TCA(enable)  : YES
```

```
    ESR-NE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    SES-NE  : 0         Threshold : 500      TCA(enable)  : YES
    SESR-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    UAS-NE  : 0         Threshold : 500      TCA(enable)  : YES
    BBE-NE  : 0         Threshold : 10000    TCA(enable)  : YES
    BBER-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    FC-NE   : 0         Threshold : 10       TCA(enable)  : YES

    ES-FE   : 0         Threshold : 500      TCA(enable)  : YES
    ESR-FE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    SES-FE  : 0         Threshold : 500      TCA(enable)  : YES
    SESR-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    UAS-FE  : 0         Threshold : 500      TCA(enable)  : YES
    BBE-FE  : 0         Threshold : 10000    TCA(enable)  : YES
    BBER-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    FC-FE   : 0         Threshold : 10       TCA(enable)  : YES
```

Example:

This sample displays the current performance monitoring for OTN parameters of the ODU-Flex with 15-minute intervals:

```
RP/0/RP0/CPU0:ios#show controllers odu-flEX 0/0/0/7/4 pm current 15-min otn pathmonitor
Fri Nov 17 16:44:34.849 UTC
g709 OTN in the current interval [16:30:00 - 16:44:34 Fri Nov 17 2023]
OTN current bucket type : Valid
    ES-NE   : 0         Threshold : 87       TCA(enable)  : YES
    ESR-NE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    SES-NE  : 0         Threshold : 1        TCA(enable)  : YES
    SESR-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    UAS-NE  : 0         Threshold : 3        TCA(enable)  : YES
    BBE-NE  : 0         Threshold : 85040    TCA(enable)  : YES
    BBER-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    FC-NE   : 0         Threshold : 10       TCA(enable)  : YES

    ES-FE   : 0         Threshold : 87       TCA(enable)  : YES
    ESR-FE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    SES-FE  : 0         Threshold : 1        TCA(enable)  : YES
    SESR-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    UAS-FE  : 0         Threshold : 3        TCA(enable)  : YES
    BBE-FE  : 0         Threshold : 85040    TCA(enable)  : YES
    BBER-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    FC-FE   : 0         Threshold : 10       TCA(enable)  : YES
```

Example:

This sample displays the current performance monitoring parameters of the coherentDSP with 15-minute intervals FEC:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 pm current 15-min fec
Fri Nov 17 16:16:05.276 UTC

g709 FEC in the current interval [16:15:00 - 16:16:05 Fri Nov 17 2023]

FEC current bucket type : Valid
    EC-BITS   : 19795040790            Threshold : 5400000000000          TCA(enable)  : YES
    UC-WORDS  : 0                      Threshold : 5                      TCA(enable)  : YES

                                      MIN       AVG        MAX      Threshold      TCA     Threshold
     TCA
                                                                        (min)      (enable)     (max)
    (enable)
PreFEC BER                    :   2.70E-04   2.79E-04   2.88E-04      0E-15         NO        0E-15
       NO
PostFEC BER                   :     0E-15     0E-15     0E-15      0E-15         NO        0E-15
    NO
Q[dB]                         :     10.70     10.70     10.70          0.00         NO         0.00
       NO
```

```
Q_Margin[dB]                    :      4.40      4.45      4.50          0.00        NO        0.00
       NO
Instantaneous Q_Margin [dB]     :      4.40      4.45      4.50          0.00        NO        0.00
       NO
```

Example:

This sample displays the current performance monitoring parameters of the Ethernet controller with 15-minute intervals for FEC.

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/1 pm current 15-min fec
Ethernet FEC in the current interval [11:30:00 - 11:31:00 Mon Oct 30 2023]
FEC current bucket type : Valid
    EC-WORDS  : 8406                    Threshold : 0                     TCA(enable)  : NO
    UC-WORDS  : 0                       Threshold : 0                     TCA(enable)  : NO
```

Example:

This sample displays the current performance monitoring parameters of the trunk optics with 15-minute intervals.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/7 pm current 15-min optics 1
Optics in the current interval [16:00:00 - 16:11:43 Fri Nov 17 2023]
Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                          Threshold(min)   Threshold(min)  (min) Threshold(max)
Threshold(max)  (max)
LBC[% ]       : 0.0       0.0       0.0      5.0              NA              NO   85.0
 NA               NO
OPT[dBm]      : 1.96      2.01      2.04     -12.01           NA              NO   4.00
 NA               NO
OPR[dBm]      : -0.55     -0.46     -0.35    -14.09           NA              NO   11.00
 NA               NO
CD[ps/nm]     : -1        0         0        -9700            NA              NO   46560
 NA               NO
DGD[ps ]      : 0.00      1.00      1.00     0.00             NA              NO   81.00
 NA               NO
SOPMD[ps^2]   : 2.00      24.45     93.00    0.00             NA              NO   60000.00
 NA               NO
OSNR[dB]      : 37.90     39.11     40.70    21.50            NA              NO   99.00
 NA               NO
PDL[dB]       : 1.70      1.91      2.10     0.00             NA              NO   3.00
 NA               NO
PCR[rad/s]    : 0.00      0.00      0.00     0.00             NA              NO   2500000.00
 NA               NO
RX_SIG[dBm]   : -1.07     -0.78     -0.64    -15.09           NA              NO   3.00
 NA               NO
FREQ_OFF[Mhz]: -112       -51       14       -3200            NA              NO   3200
 NA               NO
SNR[dB]       : 17.20     17.48     17.70    0.00             NA              NO   100.00
 NA               NO
SNR-X[dB]    : 17.40     17.67     18.00    0.00             NA              NO   300.00
NA               NO
SNR-Y[dB]    : 17.00     17.31     17.60    0.00             NA              NO   300.00
NA               NO
SOP-S1        : 0.00      0.00      0.00     -1.00            NA              NO   1.00
 NA               NO
SOP-S2        : 0.00      0.00      0.00     -1.00            NA              NO   1.00
 NA               NO
SOP-S3        : 0.00      0.00      0.00     -1.00            NA              NO   1.00
 NA               NO
```

Example:

This sample displays the current performance monitoring parameters of the client optics with 15-minute intervals.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/4 pm current 15-min optics 1
Fri Nov 17 16:13:38.671 UTC

Optics in the current interval [16:00:00 - 16:13:38 Fri Nov 17 2023]

Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                          Threshold(min)   Threshold(min)  (min) Threshold(max)
Threshold(max)  (max)
LBC[% ]      : 83.3      83.3      83.3     0.0              NA              NO    100.0
 NA              NO
OPT[dBm]     : 1.23      1.23      1.23     -2.01            NA              NO    4.00
 NA              NO
OPR[dBm]     : 1.19      1.21      1.24     -5.00            NA              NO    4.00
 NA              NO
```

Example:

This sample displays the current performance monitoring PCS parameters of the client with 15-minute intervals.

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4 pm current 15-min pcs
Ethernet PCS in the current interval [16:15:00 - 16:26:15 Fri Nov 17 2023]
Ethernet PCS current bucket type : Valid
 BIP[00]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[01]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[02]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[03]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[04]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[05]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[06]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[07]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[08]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[09]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[10]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[11]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[12]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[13]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[14]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[15]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[16]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[17]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[18]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[19]                    : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[00]                : 0                          Threshold : 0
```

```
TCA(enable) : NO
 FRM-ERR[01]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[02]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[03]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[04]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[05]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[06]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[07]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[08]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[09]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[10]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[11]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[12]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[13]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[14]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[15]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[16]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[17]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[18]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[19]                : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[00]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[01]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[02]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[03]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[04]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[05]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[06]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[07]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[08]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[09]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[10]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[11]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[12]                 : 0                          Threshold : 0
```

```
TCA(enable) : NO
 BAD-SH[13]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[14]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[15]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[16]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[17]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[18]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[19]                 : 0                          Threshold : 0
TCA(enable) : NO
 ES                         : 0                          Threshold : 0
TCA(enable) : NO
 SES                        : 0                          Threshold : 0
TCA(enable) : NO
 UAS                        : 0                          Threshold : 0
TCA(enable) : NO
 ES-FE                      : 0                          Threshold : 0
TCA(enable) : NO
 SES-FE                     : 0                          Threshold : 0
TCA(enable) : NO
 UAS-FE                     : 0                          Threshold : 0
TCA(enable) : NO
```

Example:

This sample displays the FEC performance monitoring parameters of the Ethernet controllers for various time intervals.

From Release 24.2.11, the **show controllers** command output displays post FEC BER and pre FEC BER for the Ethernet controllers of the 2.4T and 2.4TX cards.

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/1/0/4 pm current flex-bin fec
Mon Dec 18 11:30:29.101 UTC

Ethernet FEC in the current interval [11:30:20 - 11:30:28 Mon Dec 18 2023]

Flexible bin interval size: 10 seconds

FEC current bucket type : Valid
    EC-WORDS  : 11174                  Threshold : 0                  TCA(enable)  : NO
    UC-WORDS  : 0                      Threshold : 0                  TCA(enable)  : NO

               MIN        AVG        MAX      Threshold   TCA       Threshold      TCA
                                              (min)       (enable)  (max)       (enable)
PreFEC BER:   2.8E-09    2.9E-09    3.1E-09   0E-15        NO       0E-15          NO
PostFEC BER:  0E-15      0E-15      0E-15     0E-15        NO       0E-15          NO


RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/1/0/4 pm history flex-bin fec 1
Mon Dec 18 11:48:04.229 UTC

Ethernet FEC in interval 1 [11:47:50 - 11:48:00 Mon Dec 18 2023]

Flexible bin interval size: 10 seconds

FEC history bucket type : Valid
EC-WORDS   : 12128
UC-WORDS   : 0

               MIN                        AVG                        MAX
PreFEC BER :   2.6E-09                      2.9E-09                        3.0E-09
```

```
PostFEC BER:    0E-15                    0E-15                    0E-15
```

Step 2 Run the **show controllers** *controllertype R/S/I/P* **{ pm { current | history } { 30 sec | 15-min | 24-hour } { optics | ether | fec | otn | prbs}** *linenumber* } command to display the current performance monitoring parameters of the controller with 30-minute intervals.

Example:

This sample displays the current performance monitoring of PCS of the Ethernet controller with 30-second intervals:

```
RP/0/RP0/CPU0:ios#show controllers hundredGigECtrlr 0/1/0/2/1 pm current 30-sec pcs
Fri Sep 22 14:04:33.676 IST
Ethernet PCS in the current interval [14:04:30 - 14:04:33 Fri Sep 22 2023]
Ethernet PCS current bucket type : Valid
 BIP[00]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[01]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[02]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[03]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[04]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[05]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[06]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[07]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[08]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[09]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[10]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[11]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[12]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[13]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[14]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[15]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[16]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[17]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[18]                    : 0                          Threshold : 0
TCA(enable) : NO
 BIP[19]                    : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[00]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[01]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[02]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[03]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[04]                : 0                          Threshold : 0
```

```
TCA(enable) : NO
 FRM-ERR[05]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[06]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[07]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[08]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[09]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[10]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[11]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[12]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[13]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[14]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[15]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[16]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[17]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[18]                : 0                          Threshold : 0
TCA(enable) : NO
 FRM-ERR[19]                : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[00]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[01]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[02]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[03]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[04]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[05]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[06]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[07]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[08]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[09]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[10]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[11]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[12]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[13]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[14]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[15]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[16]                 : 0                          Threshold : 0
```

```
TCA(enable) : NO
 BAD-SH[17]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[18]                 : 0                          Threshold : 0
TCA(enable) : NO
 BAD-SH[19]                 : 0                          Threshold : 0
TCA(enable) : NO
 ES                         : 0                          Threshold : 0
TCA(enable) : NO
 SES                        : 0                          Threshold : 0
TCA(enable) : NO
 UAS                        : 0                          Threshold : 0
TCA(enable) : NO
 ES-FE                      : 0                          Threshold : 0
TCA(enable) : NO
 SES-FE                     : 0                          Threshold : 0
TCA(enable) : NO
 UAS-FE                     : 0                          Threshold : 0
TCA(enable) : NO
```

Example:

This sample displays the history performance monitoring of PCS of the Ethernet controller with 30-second intervals:

```
RP/0/RP0/CPU0:ios#show controllers hundredGigECtrlr 0/1/0/2/1 pm history 30-sec pcs 1
Fri Sep 22 14:06:14.193 IST
Ethernet PCS in the current interval [14:05:30 - 14:06:00 Fri Sep 22 2023]
Ethernet PCS current bucket type : Valid
 BIP[00]                    : 0
 BIP[01]                    : 0
 BIP[02]                    : 0
 BIP[03]                    : 0
 BIP[04]                    : 0
 BIP[05]                    : 0
 BIP[06]                    : 0
 BIP[07]                    : 0
 BIP[08]                    : 0
 BIP[09]                    : 0
 BIP[10]                    : 0
 BIP[11]                    : 0
 BIP[12]                    : 0
 BIP[13]                    : 0
 BIP[14]                    : 0
 BIP[15]                    : 0
 BIP[16]                    : 0
 BIP[17]                    : 0
 BIP[18]                    : 0
 BIP[19]                    : 0
 FRM-ERR[00]                : 0
 FRM-ERR[01]                : 0
 FRM-ERR[02]                : 0
 FRM-ERR[03]                : 0
 FRM-ERR[04]                : 0
 FRM-ERR[05]                : 0
 FRM-ERR[06]                : 0
 FRM-ERR[07]                : 0
 FRM-ERR[08]                : 0
 FRM-ERR[09]                : 0
 FRM-ERR[10]                : 0
 FRM-ERR[11]                : 0
 FRM-ERR[12]                : 0
 FRM-ERR[13]                : 0
 FRM-ERR[14]                : 0
 FRM-ERR[15]                : 0
 FRM-ERR[16]                : 0
```

```
 FRM-ERR[17]                : 0
 FRM-ERR[18]                : 0
 FRM-ERR[19]                : 0
 BAD-SH[00]                 : 0
 BAD-SH[01]                 : 0
 BAD-SH[02]                 : 0
 BAD-SH[03]                 : 0
 BAD-SH[04]                 : 0
 BAD-SH[05]                 : 0
 BAD-SH[06]                 : 0
 BAD-SH[07]                 : 0
 BAD-SH[08]                 : 0
 BAD-SH[09]                 : 0
 BAD-SH[10]                 : 0
 BAD-SH[11]                 : 0
 BAD-SH[12]                 : 0
 BAD-SH[13]                 : 0
 BAD-SH[14]                 : 0
 BAD-SH[15]                 : 0
 BAD-SH[16]                 : 0
 BAD-SH[17]                 : 0
 BAD-SH[18]                 : 0
 BAD-SH[19]                 : 0
 ES                         : 0
 SES                        : 0
 UAS                        : 0
 ES-FE                      : 0
 SES-FE                     : 0
 UAS-FE                     : 0
```

Step 3 Run the **show controllers** *controllertype R/S/I/P* **{ pm { current | history } { 30 sec | 15-min | 24-hour } { optics | ether | fec | otn | prbs}** *linenumber* } command to display the current performance monitoring parameters of the controller with 10-second intervals as flexi-bin.

Example:

This sample displays the current performance monitoring parameters of the trunk optics controller with 10-second intervals as flexi-bin:

```
RP/0/RP0/CPU0:ios#show controllers optics 0/1/0/0 pm current flex-bin optics 1
Fri Sep 22 14:08:37.001 IST
Optics in the current interval [14:08:30 - 14:08:36 Fri Sep 22 2023]
Flexible bin interval size: 10 seconds
Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                          Threshold(min)   Threshold(min)  (min) Threshold(max)
Threshold(max)  (max)
LBC[% ]       : 0.0       0.0       0.0      0.0              NA              NO   0.0
 NA              NO
OPT[dBm]      : -1.53     -1.49     -1.45    0.00             NA              NO   0.00
 NA              NO
OPR[dBm]      : -1.62     -1.61     -1.57    0.00             NA              NO   0.00
 NA              NO
CD[ps/nm]     : 2         2         3        0                NA              NO   0
 NA              NO
DGD[ps ]      : 3.00      3.00      3.00     0.00             NA              NO   0.00
 NA              NO
SOPMD[ps^2]   : 9.00      21.57     40.00    0.00             NA              NO   0.00
 NA              NO
OSNR[dB]      : 37.90     37.90     37.90    0.00             NA              NO   0.00
 NA              NO
PDL[dB]       : 1.10      1.10      1.10     0.00             NA              NO   0.00
 NA              NO
```

```
PCR[rad/s]    : 0.00      26.29     93.00     0.00              NA              NO    0.00
 NA               NO
RX_SIG[dBm]   : -2.14     -2.09     -2.05     0.00              NA              NO    0.00
 NA               NO
FREQ_OFF[Mhz]: 873        902       938       0                 NA              NO    0
 NA               NO
SNR[dB]       : 20.90     20.97     21.10     0.00              NA              NO    0.00
 NA               NO
SNR-AX[dB]    : 20.90     21.00     21.10     0.00              NA              NO    0.00
 NA               NO
SNR-AY[dB]    : 20.90     20.99     21.00     0.00              NA              NO    0.00
 NA               NO
SNR-BX[dB]    : 19.20     19.40     19.60     0.00              NA              NO    0.00
 NA               NO
SNR-BY[dB]    : 19.30     19.40     19.50     0.00              NA              NO    0.00
 NA               NO
SOP-S1        : 0.00      1.09      2.55      0.00              NA              NO    0.00
 NA               NO
SOP-S2        : 0.31      0.32      0.33      0.00              NA              NO    0.00
 NA               NO
SOP-S3        : 0.94      0.94      0.94      0.00              NA              NO    0.00
 NA               NO
```

Example:

This sample displays the history performance monitoring parameters of the trunk optics controller with 10-second intervals as flexi-bin:

```
RP/0/RP0/CPU0:ios#show controllers optics 0/1/0/0 pm history flex-bin optics 1 bucket 1
Fri Sep 22 14:09:54.425 IST
Optics in interval 1 [14:09:40 - 14:09:50 Fri Sep 22 2023]
Flexible bin interval size: 10 seconds
Optics history bucket type : Valid
              MIN       AVG       MAX
LBC[% ]     : 0.0       0.0       0.0
OPT[dBm]    : -1.52     -1.49     -1.47
OPR[dBm]    : -1.63     -1.59     -1.55
CD[ps/nm]   : 1         1         2
DGD[ps ]    : 2.00      2.70      3.00
SOPMD[ps^2]: 4.00       14.00     27.00
OSNR[dB]    : 37.90     37.90     37.90
PDL[dB]     : 1.10      1.10      1.10
PCR[rad/s]  : 0.00      16.00     96.00
RX_SIG[dBm] : -2.13      -2.08     -2.02
FREQ_OFF[Mhz]: 833        870       916
SNR[dB]    : 20.80     20.94     21.10
SNR-AX[dB]: 20.80      20.97     21.10
SNR-AY[dB]: 20.90      20.93     21.10
SNR-BX[dB]: 19.30      19.42     19.50
SNR-BY[dB]: 19.20      19.42     19.50
SOP-S1     : 0.00      1.53      2.55
SOP-S2     : 0.30      0.32      0.33
SOP-S3     : 0.94      0.94      0.95
```

Example:

This sample displays the current performance monitoring parameters of the coherentDSP controller as flexi-bin:

```
RP/0/0/CPU0:ios#show controllers coherentDSP 0/1/0/0 pm current flex-bin fec
Fri Sep 22 14:11:11.213 IST
g709 FEC in the current interval [14:11:10 - 14:11:10 Fri Sep 22 2023]
Flexible bin interval size: 10 seconds
FEC current bucket type : Valid
    EC-BITS   : 2532544513              Threshold : 0                     TCA(enable)   : NO
    UC-WORDS  : 0                       Threshold : 0                     TCA(enable)   : NO
```

```
                                        MIN       AVG       MAX      Threshold      TCA      Threshold
     TCA
                                                                        (min)      (enable)     (max)
    (enable)
PreFEC BER                      :   3.39E-03   3.44E-03   3.59E-03      0E-15        NO       0E-15
       NO
PostFEC BER                     :     0E-15     0E-15     0E-15      0E-15        NO       0E-15
    NO
Q[dB]                           :      8.60      8.60      8.60          0.00        NO        0.00
       NO
Q_Margin[dB]                    :      2.60      2.60      2.60          0.00        NO        0.00
       NO
Instantaneous Q_Margin [dB]     :      2.30      2.30      2.30          0.00        NO        0.00
       NO
```

Step 4 Run the **show controllers** *controllertype R/S/I/P* **{ pm { current | history } { 30 sec | 15-min | 24-hour } { optics | ether | fec | otn | prbs}** *linenumber* } command to display the current performance monitoring FEC parameters of the coherentDSP OTN.

Example:

This sample displays the current performance monitoring FEC parameters of the coherentDSP OTN with 15-minute intervals:

```
show controllers coherentDSP 0/0/0/7 pm current 15-min otn
Fri Nov 17 16:33:50.820 UTC
g709 OTN in the current interval [16:30:00 - 16:33:50 Fri Nov 17 2023]
OTN current bucket type : Valid
    ES-NE   : 0         Threshold : 500      TCA(enable)  : YES
    ESR-NE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    SES-NE  : 0         Threshold : 500      TCA(enable)  : YES
    SESR-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    UAS-NE  : 0         Threshold : 500      TCA(enable)  : YES
    BBE-NE  : 0         Threshold : 10000    TCA(enable)  : YES
    BBER-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    FC-NE   : 0         Threshold : 10       TCA(enable)  : YES

    ES-FE   : 0         Threshold : 500      TCA(enable)  : YES
    ESR-FE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    SES-FE  : 0         Threshold : 500      TCA(enable)  : YES
    SESR-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    UAS-FE  : 0         Threshold : 500      TCA(enable)  : YES
    BBE-FE  : 0         Threshold : 10000    TCA(enable)  : YES
    BBER-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    FC-FE   : 0         Threshold : 10       TCA(enable)  : YES
```

Example:

This sample displays the current performance monitoring for OTN parameters of the ODU-Flex with 15-minute intervals:

```
RP/0/RP0/CPU0:ios#show controllers odu-flEX 0/0/0/7/4 pm current 15-min otn pathmonitor
Fri Nov 17 16:44:34.849 UTC
g709 OTN in the current interval [16:30:00 - 16:44:34 Fri Nov 17 2023]
OTN current bucket type : Valid
    ES-NE   : 0         Threshold : 87       TCA(enable)  : YES
    ESR-NE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    SES-NE  : 0         Threshold : 1        TCA(enable)  : YES
    SESR-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    UAS-NE  : 0         Threshold : 3        TCA(enable)  : YES
    BBE-NE  : 0         Threshold : 85040    TCA(enable)  : YES
    BBER-NE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
    FC-NE   : 0         Threshold : 10       TCA(enable)  : YES

    ES-FE   : 0         Threshold : 87       TCA(enable)  : YES
    ESR-FE  : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
```

```
SES-FE  : 0         Threshold : 1        TCA(enable)  : YES
SESR-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
UAS-FE  : 0         Threshold : 3        TCA(enable)  : YES
BBE-FE  : 0         Threshold : 85040    TCA(enable)  : YES
BBER-FE : 0.00000   Threshold : 0.00000  TCA(enable)  : NO
FC-FE   : 0         Threshold : 10       TCA(enable)  : YES
```

Clear the PM parameters

Procedure

Run the **clear controller** *controllertype R/S/I/P* **pm** command to clear performance monitoring parameters for Ethernet and CoherentDSP controllers.

Example:

This sample shows how to clear the PM parameters on the CoherentDSP controller.

```
RP/0/RP0/CPU0:ios#show controller coherentDSP 0/0/0/0 pm current 15-min fec
Fri Sep 22 14:28:12.100 IST
g709 FEC in the current interval [14:15:00 - 14:28:12 Fri Sep 22 2023]
FEC current bucket type : Valid
Configuration Guide for Cisco NCS 1014, IOS XR Releases 25.x.x
212
Configure NCS 1014 Transponder Cards
Configuring PM Parameters
EC-BITS : 1159814176244 Threshold : 5400000000000 TCA(enable) :
YES
UC-WORDS : 0 Threshold : 5 TCA(enable) :
YES
MIN AVG MAX Threshold TCA
Threshold TCA
(min) (enable)
(max) (enable)
PreFEC BER : 0E-15 2.14E-03 2.28E-02 0E-15 NO
0E-15 NO
PostFEC BER : 0E-15 1.37E-10 6.59E-08 0E-15 NO
0E-15 NO
Q[dB] : 0.00 4.14 8.60 0.00 NO
0.00 NO
Q_Margin[dB] : -6.00 -1.89 2.60 0.00 NO
0.00 NO
Instantaneous Q_Margin [dB] : -21474836.48 -28144.25 2.30 0.00
NO 0.00 NO
Last clearing of "show controllers OTU" counters never

RP/0/RP0/CPU0:ios#clear controller coherentDSP 0/0/0/0 pm
Mon Jun 10 11:44:31.650 UTC
RP/0/RP0/CPU0:ios#show controller coherentDSP 0/0/0/0 pm current 15-min fec
Fri Sep 22 14:30:06.833 IST
g709 FEC in the current interval [14:30:00 - 14:30:06 Fri Sep 22 2023]
FEC current bucket type : Valid
EC-BITS : 17889249955 Threshold : 5400000000000 TCA(enable) :
YES
UC-WORDS : 0 Threshold : 5 TCA(enable) :
YES
```

```
MIN AVG MAX Threshold TCA
Threshold TCA
(min) (enable)
(max) (enable)
PreFEC BER : 3.38E-03 3.49E-03 3.85E-03 0E-15 NO
0E-15 NO
PostFEC BER : 0E-15 0E-15 0E-15 0E-15 NO
0E-15 NO
Q[dB] : 8.60 8.60 8.60 0.00 NO
0.00 NO
Q_Margin[dB] : 2.50 2.50 2.60 0.00 NO
0.00 NO
Instantaneous Q_Margin [dB] : 2.20 2.20 2.20 0.00 NO
0.00 NO
Last clearing of "show controllers OTU" counters 00:00:07
```

Example:

This sample shows how to clear the PM parameters on the Ethernet controller.

```
RP/0/RP0/CPU0:ios#clear controller HundredGigECtrlr 0/0/0/2/1 pm
```

PM history persistence

PM history parameters for Optics, Ethernet, and coherent DSP controllers are retained even after a line card cold reload, line card warm reload, XR reload, Calvados reload, RP reload, Hw-module all reload, power cycle, or upgrade of the NCS 1014 chassis.

After a software upgrade to the latest release, you can view the history performance monitoring parameters from the previous release. The PM history persistence is supported for 30-second, 15-minute, and 24-hour bucket types.

However, this list describes the time required to fill all historical buckets for each bucket type. After the buckets are full, no errors appear when you fetch PM historical data.

- For 30-second bucket type, 15 minutes is required to fill 30 historical buckets.
- For 15-minute bucket type, 8 hours is required to fill 32 historical buckets.
- For 24-hour bucket type, 24 hours are required to fill 7 historical bucket.

PM counters are updated continuously in the current bucket for all bucket types: flex, 30-second, 15-minute, and 24-hour. When the timer expires for a bucket type, the current PM data moves to the historical PM bucket. This process is called rollover. After rollover, the current PM data becomes available as historical PM data.

If the controller is deleted or removed, PM data remains persistent for three hours. If the controller does not return within three hours, the PM data is cleared because the controller is considered no longer in use.

Limitations of PM history persistence

If the NCS 1014 reloads during the rollover period, one of these scenarios occurs:

- A complete PM bucket may be missing, and the system marks the next PM bucket as Invalid.
- PM bucket expiry message appears as follows:

```
RP/0/RP0/CPU0:ios#show controllers hundredGigECtrlr 0/3/0/2/2 pm history 30-sec ether
29
```

```
Fri Apr  1 01:32:20.646 UTC
History data is empty, Verify at least one collection period is expired
```

- PM bucket interval is marked as Invalid and counters are updated as zero.
- PM bucket interval is marked as Invalid and counters are updated as nonzero.

Instantaneous Q-Margin

Scenarios on Instantaneous Q-margin

In these scenarios, the initial few PM buckets are displayed as valid although the instantaneous Q-margin values are displayed as invalid in those buckets. The PM is performed for 30 sec, 15 mins, and 24 hours, respectively.

Scenarios where this issue occurs include:

- Shutdown or enable optics
- Trunk rate change
- Fiber cuts

To address these situations, avoid considering the initial PM bucket readings when monitoring the instantaneous Q-margin values for these scenarios.

This sample shows that, in certain scenarios, the initial PM bucket readings are invalid. Later, the PM bucket readings become valid, even when the instantaneous Q-margin value remains invalid.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/2/0/0 pm history flex-bin fec 1
Fri Sep 22 14:17:01.008 IST
g709 FEC in interval 1 [14:16:50 - 14:17:00 Fri Sep 22 2023]
Flexible bin interval size: 10 seconds
FEC history bucket type : Valid
 EC-BITS   : 25615718133              UC-WORDS  : 0
                                 MIN                          AVG                          MAX
PreFEC BER                :    3.37E-03                     3.49E-03                     3.90E-03
PostFEC BER               :    0E-15                         0E-15                        0E-15
Q                         :    8.60                          8.60                         8.60
Q_margin                  :    2.50                          2.56                         2.60
Instantaneous Q_margin    :    2.20                          2.20                         2.20
```

Now, the PM buckets are valid although the instantaneous Q-margin value is invalid.

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/2/0/0 pm history 30-sec fec 1
Sep 22 08:52:03.750 UTC
g709 FEC in interval 1 [08:51:50 - 08:52:00 Fri Sep 22 2023]
FEC history bucket type : Invlid
 EC-BITS   : 35072302421                   UC-WORDS  : 0

                                   MIN                          AVG                          MAX
PreFEC BER                :    5.20E-03                     5.30E-03                     5.64E-03
PostFEC BER               :    0E-15                         0E-15                        0E-15
Q                         :    8.10                          8.10                         8.10
Q_margin                  :    2.10                          2.10                         2.10
Instantaneous Q_margin    :    1.80                          1.80                         1.80
```

View Ethernet statistics

You can access and view the Performance Monitoring statistics for the Ethernet controllers.

Procedure

Run the **show controller** *controllertype R/S/I/P* **pm** command to view the performance monitoring statistics for the Ethernet controllers.

Example:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4 stats
Fri Nov 17 16:28:34.138 UTC
Statistics for interface FourHundredGigECtrlr0/0/0/4 (cached values):
Ingress:
    Input total bytes           = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input good bytes            = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input total packets         = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input 802.1Q frames         = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pause frames          = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pkts 64 bytes         = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pkts 65-127 bytes     = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pkts 128-255 bytes    = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pkts 256-511 bytes    = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pkts 512-1023 bytes   = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pkts 1024-1518 bytes  = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input pkts 1519-Max bytes   = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input good pkts             = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input unicast pkts          = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input multicast pkts        = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input broadcast pkts        = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input drop overrun          = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input drop abort            = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input drop invalid VLAN     = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input drop invalid DMAC     = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input drop invalid encap    = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input drop other            = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input error giant           = 0                     Valid = False        Start time = 13:12:29
```

```
 Fri Nov 17 2023
    Input error runt            = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input error jabbers         = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input error fragments       = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input error CRC             = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input error collisions      = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input error symbol          = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input error other           = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input MIB giant             = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input MIB jabber            = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Input MIB CRC               = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
Egress:
    Output total bytes          = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output good bytes           = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output total packets        = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output 802.1Q frames        = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pause frames         = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pkts 64 bytes        = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pkts 65-127 bytes    = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pkts 128-255 bytes   = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pkts 256-511 bytes   = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pkts 512-1023 bytes  = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pkts 1024-1518 bytes = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output pkts 1519-Max bytes  = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output good pkts            = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output unicast pkts         = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output multicast pkts       = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output broadcast pkts       = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output drop underrun        = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output drop abort           = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output drop other           = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
    Output error other          = 0                     Valid = False        Start time = 13:12:29
 Fri Nov 17 2023
```

Note

Performance monitoring statistics are not supported for the input unicast packets, output unicast packets, and input error fragments counters for Ethernet clients.

Performance monitoring for NCS1K14-2.4T-X-K9 card

Service providers use performance monitoring (PM) parameters to gather performance data, store it, set thresholds, and report it. This helps with early detection of network issues.

You can configure and retrieve PM counters at intervals of 30 seconds, 15 minutes, or 24 hours. These parameters simplify troubleshooting operations and increase the amount of data that can be collected directly from the equipment.

Performance monitoring for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards

Table 40: Feature History

Feature Name	Release Information	Feature Description
Supported Functionalities of CCMD-16-C and CCMD-16-L Line Cards	Cisco IOS XR Release 7.11.1	Supported Functionalities of CCMD-16-C and CCMD-16-L Line Cards: The software supports Variable Optical Attenuator (VoA), power monitoring and reporting of parameters to the controllers at the OCH and OMS level. It helps in configuring the amplifier parameters for optimizing signal transmissions. The software also supports in-band and out-of-band tone detection and monitoring and reporting of alarms.

Service providers use performance monitoring (PM) parameters to gather and store data, set thresholds, and report performance information for early detection of network issues.

You can configure and retrieve PM counters for the OCH and OMS controllers in 30-second, 15-minute, or 24-hour intervals, or in a flexible 10-second bin interval. These parameters simplify troubleshooting and enhance the value of data collected directly from the equipment.

PM parameters supported on OMS controller

These are the PM parameters supported on the OMS controller:

Table 41: PM parameters supported on OMS controller

Controller	Supported PM parameters	Description
OMS	OPT (dBm)	Transmitted power
	OPR (dBm)	Received Power
	OPBR (dBm)	Back Reflection Power
	OPBRR (dB)	Back Reflection Ratio
	EAGN (dB)	Egress Ampli Gain
	EATL (dB)	Egress Ampli Tilt
	IAGN (dB)	Ingress Ampli Gain
	IATL (dB)	Ingress Ampli Tilt

PM parameters supported on OCH controller

These are the PM parameters supported on the OCH controller:

Controller	Supported PM parameters	Description
OCH	OPT (dBm)	Transmitted Power
	OPR (dBm)	Received Power

Configure PM parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards

Follow these steps to configure the minimum and maximum thresholds for individual parameters.

Procedure

Step 1 Run the **controller** *controllertype R/S/I/P* **pm** {**30-sec** | **15-min** | **24-hour**} **optics threshold** { **parameter-name**} {**max**|**min**} {**value**} command to configure minimum and maximum thresholds for individual performance monitoring parameters.

Example:

This sample displays the performance monitoring parameters of the OMS controller.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:(config)#controller oms 0/1/0/0 pm 30-sec optics threshold opt min < value >
RP/0/RP0/CPU0:ios(config)#commit
```

This sample displays the performance monitoring parameters of the OCH controller.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:(config)#controller och 0/1/0/1 pm 30-sec optics threshold opt min < value >
RP/0/RP0/CPU0:ios(config)#commit
```

Step 2 Run the **controller** *controllertype R/S/I/P* **pm** {**30-sec** | **15-min** | **24-hour**} **optics report** { **parameter-name**} {**min-tca**|**max-tca**} command to enable reporting of threshold crossing alarms for individual parameters.

Step 3 Run the **commit** command to save the changes.

View PM parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards

Use this task to view the performance monitoring parameters for OMS and OCH controllers.

Procedure

Step 1 Run the **show controllers** *controllertype R/S/I/P* **pm { current | history } { 30 sec | 15-min | 24-hour | flex-bin } optics** { *linenumber* } **bucket***value* command to view PM parameters for the OMS and OCH controllers.

Example:

This is a sample to view the PM parameters of the OMS controller.

```
RP/0/RP0/CPU0:ios_P2A_DT_03#show controllers oms 0/1/0/0 pm current 30-sec optics 1
Optics in the current interval [15:02:30 - 15:02:36 Mon Nov 20 2023]
Optics current bucket type : Valid
             MIN       AVG       MAX      Operational      Configured      TCA   Operational
Configured     TCA
                                          Threshold(min)   Threshold(min)  (min)  Threshold(max)
Threshold(max)  (max)
OPT[dBm]     : -8.30     -8.24     -8.20    -50.00            NA              NO   30.00
 NA               NO
OPR[dBm]     : -1.80     -1.76     -1.60    -50.00            NA              NO   30.00
 NA               NO
OPBR[dBm]     : -11.61    -11.61    -11.61   -50.00            NA              NO   -10.00
  NA               NO
OPBRR[dB]     : -3.30     -3.30     -3.30    -50.00            NA              NO   0.00
  NA               NO
EAGN[dB]      : 2.00      2.00      2.00     -3.00             NA              NO   22.00
  NA               NO
EATL[dB]      : 0.00      0.00      0.00     -6.50             NA              NO   6.50
  NA               NO
IAGN[dB]      : 5.00      5.00      5.00     0.00              NA              NO   10.00
  NA               NO
IATL[dB]      : 0.00      0.00      0.00     -6.50             NA              NO   6.50
  NA               NO
Last clearing of "show controllers OPTICS" counters never
```

This is a sample to view the PM parameters of the OCH controller.

```
RP/0/RP0/CPU0:ios_P2A_DT_03#show controllers och 0/1/0/2 pm current 30-sec optics 1
Optics in the current interval [15:04:30 - 15:04:39 Mon Nov 20 2023]
Optics current bucket type : Valid
MIN AVG MAX Operational Configured TCA Operational Configured TCA
Threshold(min) Threshold(min) (min) Threshold(max) Threshold(max) (max)
OPT[dBm] : -1.40 -1.36 -1.30 -50.00 NA NO 30.00 NA NO
OPR[dBm] : -5.80 -5.71 -5.70 -50.00 NA NO 30.00 NA NO


RP/0/RP0/CPU0:ios_P2A_DT_03#show controllers och 0/1/0/2 pm current 15-min optics 1
Optics in the current interval [15:00:00 - 15:05:03 Mon Nov 20 2023]
Optics current bucket type : Valid
MIN AVG MAX Operational Configured TCA Operational Configured TCA
```

```
Threshold(min) Threshold(min) (min) Threshold(max) Threshold(max) (max)
OPT[dBm] : -1.80 -1.50 -1.30 -50.00 NA NO 30.00 NA NO
OPR[dBm] : -5.80 -5.75 -5.70 -50.00 NA NO 30.00 NA NO
Last clearing of "show controllers OPTICS" counters never
```

Step 2 Run the **show controllers** *controllertype R/S/I/P* **pm history { 30 sec | 15-min | 24-hour }**[**optics** { *lanenumber* }]**bucket** *value* to view the PM history parameters for OMS and OCH controllers.

Example:

This is a sample to view the historical PM parameters on a Coherent DSP controller for the past 2 days using the 24-hour interval.

```
RP/0/RP0/CPU0:Node164#show controllers coherentDSP 0/3/0/0 pm history 24-hour fec bucket 2
Sat Oct 19 13:55:13.872 IST

g709 FEC in interval 2 [00:00:00 - 24:00:00 Thu Oct 17 2024]

FEC history bucket type : valid
EC-BITS   : 10121314105194            UC-WORDS  : 0

                              MIN                          AVG                          MAX

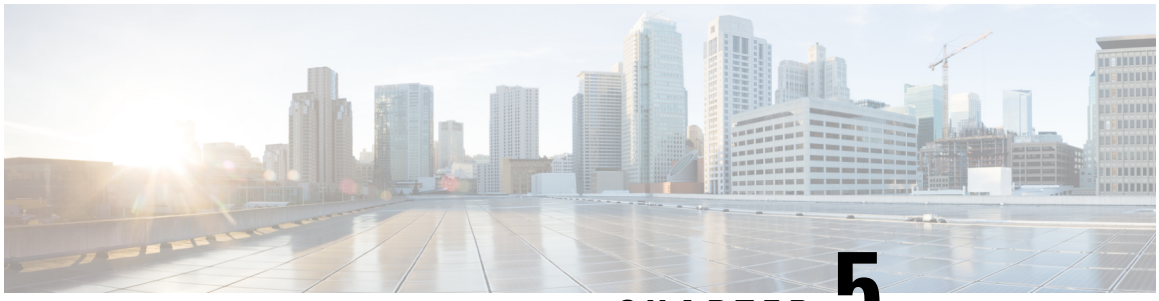
PreFEC BER              :    1.43E-04                      1.51E-04                     1.59E-04
PostFEC BER             :    0E-15                         0E-15                        0E-15
Q                       :    11.10                         11.19                        11.20
Q_margin                :    4.70                          4.79                         4.80
Instantaneous Q_margin  :    4.69                          4.79                         4.80

Last clearing of "show controllers OTU" counters never
```

Example:

This is a sample to view the historical PM parameters on an OMS controller for the 30-second interval.

```
RP/0/RP0/CPU0:ios_P2A_DT_02#show controllers oms 0/3/0/0 pm history 30-sec optics 1 bucket 1
Wed Dec 6 11:04:50.821 UTC
Optics in interval 1 [11:04:00 - 11:04:30 Wed Dec 6 2023]
Optics history bucket type : Valid
MIN AVG MAX
OPT[dBm]     : -8.30     -8.27     -8.20
OPR[dBm]     : -3.00     -1.62     -0.20
OPBR[dBm]     : -11.61    -11.61    -11.51
OPBRR[dB]     : -3.40     -3.31     -3.30
EAGN[dB]      : 2.00      2.00      2.00
EATL[dB]      : 0.00      0.00      0.10
IAGN[dB]      : 5.00      5.00      5.00
IATL[dB]      : 0.00      0.00      0.00
```



CHAPTER 5

Pseudo Random Binary Sequence

- Pseudo Random Binary Sequence , on page 157
- Trunk PRBS, on page 164
- Client PRBS , on page 166

Pseudo Random Binary Sequence

The Pseudo Random Binary Sequence (PRBS) feature enables data integrity checks between NCS 1014 trunk and client links without generating client traffic.

The PRBS feature must be enabled on both the transmitting and receiving ports to ensure error-free traffic during link bring-up. The transmitting port creates and sends a bit pattern to the peer device. The NCS 1014 device verifies that the correct pattern is received. Additionally, PRBS is supported on the ODU4 controller by the 1.2T card.

You can configure PRBS on a coherentDSP or Ethernet controller of a 1.2T, 2.4T, or 2.4TX cards. PRBS can also be configured on a ODU controller of the 1.2T card. Before enabling PRBS, the secondary admin state of the controllers must be set to *maintenance*.

PRBS modes

PRBS operates in several modes, each offering unique functionality:

- **Source mode:** The NCS 1014 trunk port generates a PRBS signal on the line continuously according to the configured PRBS pattern
- **Sink mode:** The NCS 1014 trunk port locks onto the ingress signal according to the configured pattern, analyzes, and reports the errors.
- **Source-Sink mode:** The NCS 1014 trunk port acts as both the PRBS transmitter and receiver, meaning it generates a PRBS signal according to the configured pattern, locks onto the ingress signal with the same pattern, and reports the errors.

Configure PRBS on CoherentDSP controller

Use this task to configure PRBS on the trunk port of the coherentDSP controller of a 1.2T, 2.4T, or 2.4TXcard.

Procedure

Step 1 Run the **controller coherentDSP** *R/S/I/P* command to enter the coherentDSP controller configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)#controller CoherentDSP 0/0/0/7
```

Step 2 Run the **secondary-admin-state { normal | maintenance}** command to set the secondary admin state.

Example:

```
RP/0/RP0/CPU0:ios(config-CoDSP)#secondary-admin-state maintenance
```

Step 3 Run the **prbs mode {source | sink | source-sink} pattern {pn31 | pn23 | pn15 | pn7}** command to configure the PRBS mode and pattern.

Example:

```
RP/0/RP0/CPU0:ios(config-CoDSP)#prbs mode source-sink pattern pn15
```

Step 4 Commit the changes.

Example:

```
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

Configure PRBS on ODU controller

You can configure PRBS in these combinations for an ODU controller on a 1.2T card.

The client ports need to have a physical loopback in all the combinations:

- Near end client and near end trunk ODU4
- Near end client and far end client ODU4
- Near end client and far end trunk ODU4
- Near end trunk and far end trunk ODU4

Use this task to enable PRBS on ODU4 controller's trunk ports of a 1.2T card in configuration mode.

Procedure

Step 1 Run the **controller odu4** *R/S/I/P* command to enter the odu4 controller configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)#controller odu4 0/0/0/7
```

Step 2 Run the **secondary-admin-state { normal | maintenance}** command to set the secondary admin state.

Example:

```
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#secondary-admin-state maintenance
```

Step 3 Run the **opu prbs mode {source | sink | source-sink} pattern {pn31 | pn23 | pn15 | pn7}** command to configure the PRBS mode and pattern.

Example:

```
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern pn23
```

Step 4 Commit the changes.

Example:

```
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Configure PRBS on Ethernet controller

Use this task to configure PRBS on the client ports of an Ethernet controller of the 2.4T and 2.4TX cards.

Procedure

Step 1 Run the **controller** {**fourHundredGigECtrlr**} *R/S/I/P* command to enter the ethernet controller configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)#controller FourHundredGigECtrlr 0/3/0/1
```

Step 2 Run the **secondary-admin-state { normal | maintenance}** command to set the secondary admin state.

Example:

```
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
```

Step 3 Run the **opu prbs mode {source | sink | source-sink} pattern {pn31 | pn23 }** command to configure the PRBS mode and pattern.

Example:

```
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern pn31
```

Step 4 Commit the changes.

Example:

```
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Verify PRBS on controllers

Use this task to verify PRBS details on the controllers.

Procedure

Step 1 Run the **show controllers fourHundredGigEctrlr** *R/S/I/P* **prbs-details** command to display PRBS details configured on an Ethernet controller.

Example:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4
Fri Sep 25 09:45:18.222 UTC
Operational data for interface FourHundredGigECtrlr0/0/0/4:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Enabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    PRBS:
      Status: Locked
      Mode: Source-sink
      Pattern: PN31
      Direction: Line
      Framing: Framed
      Configured Time: 25 Sep 09:44:56 (22 seconds elapsed)
      First Lock Established Time: 25 Sep 09:44:59 (19 seconds elapsed)
      Lock Time(in seconds): 19
      Bit errors: 0
      Lock Found count: 1
      Lock Lost count: 0
      Result: PASS
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled
    State transition count: 1
    Last link flapped: 00:29:47
```

Step 2 Run the **show controllers coherentDSP** *R/S/I/P* **prbs-details** command to display PRBS details configured on a coherentDSP controller.

Example:

```
RP/0/RP0/CPU0:ios#show  controllers coherentDSP 0/0/0/7 prbs-details
Wed Nov 15 18:13:35.210 UTC

----------------------PRBS details------------------
PRBS Test               : Enable
PRBS Mode               : Source-Sink
PRBS Pattern            : PN15
PRBS Status             : Locked
```

Step 3 Run the **show controllers ODU** *R/S/I/P* **prbs-details** command to display PRBS details configured on an ODU controller where the PRBS status is displayed as **Not Applicable**, when the mode is **Source**.

Example:

```
RP/0/RP0/CPU0:ios#show controllers ODU4 0/3/0/8 prbs-details
Mon Jan 11 05:29:12.436 UTC

 -------------------------PRBS details----------------------------------
PRBS Test                      : Enable
```

```
PRBS Mode                       : Source
PRBS Pattern                    : PN7
PRBS Status                     : Not Applicable

------------------------------------------------
```

Verify the PRBS performance monitor parameters

Use this task to verify the PRBS performance monitoring parameters on a coherentDSP, ODU, or Ethernet controller.

Procedure

Step 1 Run the **show controllers coherentDSP** *R/S/I/P* **pm** {**current | history** }{**15-min|24-hour**} **prbs** command.

Example:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 pm current 15-min prbs
Wed Nov 15 18:19:10.308 UTC

 PRBS in the current interval [18:15:00 - 18:19:10 Wed Nov 15 2023]

 PRBS current bucket type : Valid

 EBC          : 0           Threshold : 0         TCA(enable)  : NO
 FOUND-COUNT  : 0           Threshold : 0         TCA(enable)  : NO
 LOST-COUNT   : 0           Threshold : 0         TCA(enable)  : NO

 FOUND-AT-TS  : NULL
 LOST-AT-TS   : NULL

 CONFIG-PTRN  : PRBS_PATTERN_PN15
 STATUS       : LOCKED

Last clearing of "show controllers OTU" counters never
```

Step 2 Run the **show controllers ODU4** *R/S/I/P* **pm** {**current | history** }{**15-min|24-hour**} **prbs** command to view PRBS performance monitoring parameters for a 15-minute sampling interval on an ODU controller.

Example:

```
RP/0/RP0:ios#show controllers ODU4 0/3/0/1 pm current 15-min prbs
Mon Jan 11 00:58:48.327 UTC

PRBS in the current interval [00:45:00 - 00:58:48 Mon Jan 11 2021]
PRBS current bucket type : Valid
EBC                     : 40437528165
FOUND-COUNT             : 1 FOUND-AT-TS : 00:51:22 Mon Jan 11 2021
LOST-COUNT              : 1 LOST-AT-TS  : 00:52:52 Mon Jan 11 2021
CONFIG-PTRN             : PRBS_PATTERN_PN7
Last clearing of "show controllers ODU" counters never
```

Step 3 Run the **show controllers ODU4** *R/S/I/P* **pm** {**current | history** }{**15-min|24-hour**} **prbs** command to view PRBS performance monitoring parameters for a 15-minute sampling interval on an Ethernet controller.

Example:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4 pm current 15-min prbs
Wed Nov 15 18:48:19.114 UTC

 PRBS in the current interval [18:45:00 - 18:48:19 Wed Nov 15 2023]

 PRBS current bucket type : Valid

 EBC          : 0           Threshold : 0         TCA(enable)  : NO
 FOUND-COUNT  : 0           Threshold : 0         TCA(enable)  : NO
 LOST-COUNT   : 0           Threshold : 0         TCA(enable)  : NO

 FOUND-AT-TS  : NULL
 LOST-AT-TS   : NULL

 CONFIG-PTRN  : PRBS_PATTERN_PN23
 STATUS       : LOCKED

Last clearing of "show controllers ETHERNET" counters never
```

This table describes the fields of PRBS PM parameters.

Table 42: PRBS PM parameters

PM parameter	Description
EBC	Cumulative count of PRBS bit errors in the sampling interval (15-minute or 24-hour). PRBS bit errors are accumulated only if PRBS signal is locked.
FOUND-COUNT	Number of state transitions from signal unlocked state to signal locked state in the sampling interval. If state change is not observed in the interval, the count is 0.
LOST-COUNT	Number of state transitions from signal locked state to signal unlocked state in the sampling interval. If state change is not observed in the interval, the count is 0.
FOUND-AT-TS	Latest timestamp when the PRBS state moves from unlocked state to locked state in the sampling interval. If state change is not observed in the interval, the value is null.
CONFIG-PTRN	Configured PRBS pattern on the port.
STATUS	Displays the PRBS status.

Clear PRBS bit errors and lock time

Lock time refers to the duration elapsed since the last PRBS lock was detected.

Use this task to clear the bit errors and lock time.

Procedure

Step 1 Run the **show controllers ODU4** *R/S/I/P* **prbs-details** command to show that bit errors are observed during the PRBS test.

Example:

```
RP/0/RP0/CPU0:ios#show controllers odu4 0/2/0/5 prbs-details
Fri Nov 13 03:21:44.191 UTC

-------------------------PRBS details----------------------------------
PRBS Test : Enable
PRBS Mode : Source-Sink
PRBS Pattern : INVERTED PN31
PRBS Status : Locked
PRBS Direction : Line
PRBS Lock Time(in seconds) : 28
PRBS Bit Errors : 23776

-------------------------------------------------
```

Step 2 Run the **clear controller ODU4** *R/S/I/P* **prbs details** command to clear the lock time and bit errors before the PRBS test.

Example:

```
RP/0/RP0/CPU0:ios#clear controller odu4 0/2/0/5 prbs-details
Fri Nov 13 03:21:50.726 UTC
PRBS bit errors cleared
```

This sample shows the removal of bit errors and lock time.

```
RP/0/RP0/CPU0:ios#show controllers odu4 0/2/0/5 prbs-details
Fri Nov 14 03:21:44.191 UTC
-------------------------PRBS details----------------------------------
PRBS Test : Enable
PRBS Mode : Source-Sink
PRBS Pattern : INVERTED PN31
PRBS Status : Locked
PRBS Direction : Line
PRBS Lock Time(in seconds) : 2
PRBS Bit Errors : 0
```

Trunk PRBS

Table 43: Feature History

Feature Name	Release Information	Description
Cumulative PRBS on CoherentDSP controllers	Cisco IOS XR Release 24.3.1	The cumulative PRBS (Pseudo-Random Binary Sequence) on CoherentDSP controllers enhances troubleshooting capabilities between the trunk ports. Show coherentDSP R/S/I/P prbs-details command output now includes the newly supported fields.

The trunk PRBS feature performs data integrity checks between trunk links without generating actual client traffic.

NCS 1014 trunk port supports these PRBS patterns:

- PRBS31: Sequence length is from 2^31 -1 bits.
- PRBS23: Sequence length is from 2^23 -1 bits.
- PRBS15: Sequence length is from 2^15 -1 bits.
- PRBS11: Sequence length is from 2^11 -1 bits.
- PRBS7: Sequence length is from 2^7 -1 bits.

Tip It is recommended to use these patterns for higher datarates like 100G and 400G:

- high sequence length PRBS patterns
- PRBS inverted pattern

Note The ethernet PRBS PN23 pattern interoperability is not supported on the 2.4T and 2.4TX cards.

Configure the trunk PRBS

Use this task to configure PRBS trunk mode on the NCS1K4-2.4T-K9 card .

Procedure

Step 1 Run the **controller coherentDSP** *R/S/I/P* command to enter the coherentDSP controller configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)#controller CoherentDSP0/0/0/7
RP/0/RP0/CPU0:ios(config-CoDSP)#secondary-admin-state maintenance
RP/0/RP0/CPU0:ios(config-CoDSP)#prbs mode source-sink pattern pn15
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
Wed Nov 15 18:11:55.450 UTC
```

Step 2 Run the **show controllers coherentDSP** *R/S/I/P* **prbs-details** command to display PRBS details configured on a coherentDSP controller.

Example:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 prbs-details
Wed Nov 15 18:13:35.210 UTC

----------------------PRBS details------------------
PRBS Test               : Enable
PRBS Mode               : Source-Sink
PRBS Pattern            : PN15
PRBS Status             : Locked
PRBS Lock Time(in seconds)      : 37
PRBS Bit Errors                 : 0
PRBS Found Count                : 1
PRBS Lost Count                 : 0
PRBS Configured Time            : 11 Feb 00:20:43 (719 seconds elapsed)
PRBS First Lock Established Time: 11 Feb 00:32:05 (37 seconds elapsed)
Result Summary                  : PASS
```

The **Result Summary** will display **PASS** if the PRBS bit errors are 0 and the `PRBS elapsed lock time` is equal to the elapsed first lock established time.

Step 3 Run the **clear controllers coherentDSP** *R/S/I/P* **prbs-details** command to clear the counters.

Example:

```
clear controller coherentDSP 0/0/0/7 prbs-detailsentDSP
```

Step 4 Run the **show controllers coherentDSP** *R/S/I/P* **prbs-details** command to display cumulative count of PRBS bit errors in the 15-min sampling interval.

Example:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 pm current 15-min prbs
Wed Nov 15 18:19:10.308 UTC

 PRBS in the current interval [18:15:00 - 18:19:10 Wed Nov 15 2023]

 PRBS current bucket type : Valid

 EBC          : 0           Threshold : 0         TCA(enable)  : NO
 FOUND-COUNT  : 0           Threshold : 0         TCA(enable)  : NO
 LOST-COUNT   : 0           Threshold : 0         TCA(enable)  : NO

 FOUND-AT-TS  : NULL
 LOST-AT-TS   : NULL

 CONFIG-PTRN  : PRBS_PATTERN_PN15
 STATUS       : LOCKED

Last clearing of "show controllers OTU" counters never
```

Client PRBS

Table 44: Feature History

Feature Name	Release Information	Description
New PRBS Counters on Ethernet Controllers	Cisco IOS XR Release 24.4.1	The new Pseudo-Random Binary Sequence (PRBS) counters on Ethernet controllers collect statistics in a cumulative manner. The output of the `show controllers controller-type R/S/I/P` command now includes these counters: • Configured Time • First Lock Established Time. • Lock Time (in seconds) • Bit Errors • Lock Found Count • Lock the Lost Count • Result

The client PRBS feature performs data integrity checks without generating client traffic on client links.

The NCS 1014 client port supports these PRBS patterns:

- PRBS31: Sequence length is from 2^31 -1 bits.
- PRBS23: Sequence length is from 2^23 -1 bits.

Note In case of muxponder configuration, PRBS is not supported on the split ports of the 2.4TX card.

Configure the client PRBS

Use this task to configure PRBS client mode on the NCS1K4-2.4T-K9 card .

Procedure

Step 1 Run the **controller** {**fourHundredGigECtrlr**} *R/S/I/P* command to enter the four hundred gigabit client controller configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config)#controller fourHundredGigECtrlr 0/2/0/4
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern <pn31|pn23> direction system
<line|system>
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Step 2 Run the **show controllers fourHundredGigEctrlr** *R/S/I/P* **prbs-details** command to display four hundred gigabit client controllers details.

Example:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4
Fri Sep 25 09:48:03.959 UTC
Operational data for interface FourHundredGigECtrlr0/0/0/4:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Enabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
    PRBS:
      Status: Locked
      Mode: Source-sink
      Pattern: PN31
      Direction: Line
      Framing: Framed
      Configured Time: 25 Sep 09:44:56 (187 seconds elapsed)
      First Lock Established Time: 25 Sep 09:47:52 (11 seconds elapsed)
      Lock Time(in seconds): 11
      Bit errors: 0
      Lock Found count: 0
      Lock Lost count: 0
      Result: PASS
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled
    State transition count: 1
    Last link flapped: 00:32:32
```

The **Result** displays **PASS**, only if the **First Lock Established Time** and **Lock Time (in seconds)** are same, and **Bit errors** is 0.

Step 3 Run the **clear controllers fourHundredGigEctrlr** *R/S/I/P* **prbs-details** command to clear four hundred gigabit client controller PRBS statistics.

Example:

```
RP/0/RP0/CPU0:ios#clear controller fourHundredGigEctrlr 0/0/0/4 prbs-stats
Fri Sep 25 09:47:52.678 UTC
Operational data for interface FourHundredGigECtrlr0/0/0/4:
State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On
    Maintenance: Enabled
    AINS Soak: None
      Total Duration: 0 hour(s) 0 minute(s)
      Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
```

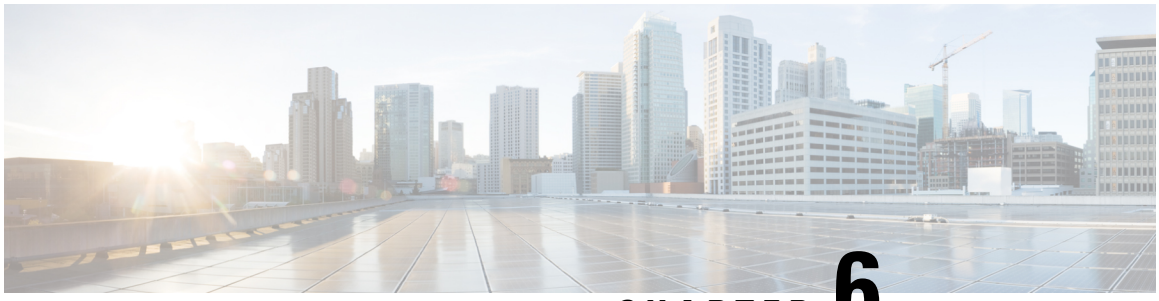
```
    PRBS:
      Status: Locked
      Mode: Source-sink
      Pattern: PN31
      Direction: Line
      Framing: Framed
      Configured Time: 25 Sep 09:44:56 (179 seconds elapsed)
      First Lock Established Time: 25 Sep 09:47:52 (3 seconds elapsed)
      Lock Time(in seconds): 3
      Bit errors: 0
      Lock Found count: 0
      Lock Lost count: 0
      Result: PASS
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled
    State transition count: 1
    Last link flapped: 00:32:24
```

Step 4 Run the **show controllers fourHundredGigEctrlr** *R/S/I/P* **prbs-details** command to display four hundred gigabit client controller PRBS bit errors in the 15-min sampling interval.

Example:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4 pm current 15-min prbs
Wed Nov 15 18:48:19.114 UTC
 PRBS in the current interval [18:45:00 - 18:48:19 Wed Nov 15 2023]
 PRBS current bucket type : Valid
 EBC          : 0           Threshold : 0         TCA(enable)  : NO
 FOUND-COUNT  : 0           Threshold : 0         TCA(enable)  : NO
 LOST-COUNT   : 0           Threshold : 0         TCA(enable)  : NO
 FOUND-AT-TS  : NULL
 LOST-AT-TS   : NULL
 CONFIG-PTRN  : PRBS_PATTERN_PN23
 STATUS       : LOCKED

Last clearing of "show controllers ETHERNET" counters never
```



CHAPTER 6

Implementing Host Services and Applications

Cisco IOS XR software Host Services and Applications features on the router are used primarily for checking network connectivity and the route a packet follows to reach a destination, mapping a hostname to an IP address or an IP address to a hostname, and transferring files between routers and UNIX workstations.

Prerequisites for implementing Host Services and Applications

Ensure to install the relevant optional RPM package before using the host services or applications.

- HTTP Client application , on page 169
- Transmission Control Protocol , on page 171

HTTP Client application

HTTP Client allows files to be transferred from an HTTP server to another device over a network using the HTTP protocol.

You can configure the HTTP Client and its associated parameters by using the **http client** command.

This table lists the commands used to configure HTTP Client settings:

Table 45: Commands used to configure HTTP Client settings

Features	Description
connection	Configure HTTP Client connection by using either retry or timeout options.
response	How long HTTP Client waits for a response from the server for a request message before giving up.
secure-verify-host	Verify host in peer's certificate. To disable verifying this, you can use the command **http client secure-verify-host disable**
secure-verify-peer	Verify authenticity of the peer's certificate.
source-interface	Specifies the source interface for all outgoing HTTP connections. You can enter an IPv4 address, an IPv6 address, or both addresses.

Features	Description
ssl version	SSL version (configuration) to be used for HTTPS requests.
tcp-window-scale scale	Sets the TCP window-scale factor for high-latency links.
version version	HTTP version to be used in HTTP requests. • 1.0—HTTP1.0 will be used for all HTTP requests. • 1.1—HTTP1.1 will be used for all HTTP requests. • default libcurl—will use HTTP version automatically.
vrf name	Name of the VRF.

Configure HTTP Client application

The HTTP Client application is available by default. You can configure HTTP client settings, or you can view and modify existing settings.

Use this task to configure the settings.

Procedure

Step 1 Enter the XR config mode.

Step 2 Run the **http client** command in XR config mode.

Example:

```
Router #configure
Router(config)#http client ?
connection          Configure HTTP Client connection
response            How long HTTP Client waits for a response from the server
                    for a request message before giving up
secure-verify-host  Verify that if server certificate is for the server it is known as
secure-verify-peer  Verify authenticity of the peer's certificate
source-interface    Specify interface for source address
ssl                 SSL configuration to be used for HTTPS requests
tcp-window-scale    Set tcp window-scale factor for High Latency links
version             HTTP Version to be used in HTTP requests
vrf                 Name of vrf
```

Example:

This example shows how to set the TCP window-scale to 8.

```
Router(config)#http client tcp-window-scale 8
```

This example shows how to set the HTTP version to 1.0.

```
Router(config)#http client version 1.0
```

Note
The HTTP Client uses libcurl version 7.30.

Transmission Control Protocol

Transmission Control Protocol (TCP) is a connection-oriented protocol that defines the format of data and acknowledgments exchanged between two computer systems to facilitate data transfer.

- TCP outlines procedures to ensure that the data being transferred arrives correctly at the intended destination.
- TCP enables multiple applications on a system to communicate simultaneously. It manages the demultiplexing of incoming traffic among application programs.

TCP dump file converter

The TCP dump file converter is a tool that converts IOS-XR dump files from binary format into user-friendly formats such as PCAP or text.

Key features

The key features of the file converter include:

- The converter is especially useful when Non-Stop Routing (NSR) is disabled or a session flap occurs on your system. In these cases, the TCP process on the NCS system automatically stores the latest 200 packet traces in binary format in a temporary folder.
- TCP dump packet traces also include data about the configured routing protocols and overall network traffic on your system. This data provides insights to help you identify and resolve network infrastructure issues, enabling proactive troubleshooting.

Binary files

You can view packet traces binary files in the user-readable format using these methods:

- Use the **show tcp dump-file** *binary-filename* command to view each binary file in text format manually. For more information, refer to View binary files in text format manually, on page 172

Note This process consumes much time, as you have to view each file manually one after another.

- Convert all stored packet traces in binary files into PCAP, text, or both using the **tcp dump-file convert** command.

For more information, refer to Convert binary files to readable format using TCP dump file converter , on page 173. This active approach greatly improves the efficiency and ease of packet analysis during network troubleshooting.

Limitations and restrictions for TCP dump file converter

The TCP dump file converter has these limitations and restrictions:

- The system only stores the most recent 200 message exchanges that occurred right before the session termination, when NSR is disabled, or during a session flap.
- You can view only one binary file in text format using the **show tcp dump-file** *binary- filename* command.
- TCP dump files are generated by default for BGP, MSDP, MPLS LDP, and SSH.

View binary files in text format manually

Use this task to view each packet trace binary file in text format without using the TCP dump file converter.

Procedure

Step 1 Run the **show tcp dump-file list all** command to view the list of packet traces in binary files stored in the tcpdump folder.

Example:

```
RP/0/RP0/CPU0:ios# show tcp dump-file list all
                        total 1176
                        -rw-r--r-- 1 root root 5927 Nov 22 12:42 31_0_0_126.179.20966.cl.1700656933
                        -rw-r--r-- 1 root root 5892 Nov 22 12:42 31_0_0_127.179.35234.cl.1700656933
                        -rw-r--r-- 1 root root 6148 Nov 22 12:42 31_0_0_149.179.54939.cl.1700656933
                        -rw-r--r-- 1 root root 5894 Nov 22 12:42 31_0_0_155.179.18134.cl.1700656933
                        -rw-r--r-- 1 root root 6063 Nov 22 12:42 31_0_0_156.179.25445.cl.1700656933
                        -rw-r--r-- 1 root root 5860 Nov 22 12:42 31_0_0_161.179.30859.cl.1700656933
                        -rw-r--r-- 1 root root 5832 Nov 22 12:42 31_0_0_173.179.36935.cl.1700656933
                        -rw-r--r-- 1 root root 5906 Nov 22 12:42 31_0_0_190.179.25642.cl.1700656933
```

Step 2 Run the **show tcp dump-file** *binary- filename* command to view each packet traces binary file in text format.

Example:

```
RP/0/RP0/CPU0:ios# show tcp dump-file 10_106_0_73.179.34849.cl.1707424077 location 0/RP0/CPU0
                        Filename: 10_106_0_73.179.34849.cl.1707424077

                        ============================================================
                        Connection state is CLOSED, I/O status: 0, socket status: 103
                        PCB 0x00007f86bc05e3b8, SO 0x7f86bc05e648, TCPCB 0x7f86bc0c3718, vrfid
0x60000000,
                        Pak Prio: Medium, TOS: 192, TTL: 1, Hash index: 1593
                        Local host: 10.106.0.72, Local port: 179 (Local App PID: 11354)
                        Foreign host: 10.106.0.73, Foreign port: 34849
                        (Local App PID/instance/SPL_APP_ID: 11354/1/0)

                        Current send queue size in bytes: 0 (max 0)
                        Current receive queue size in bytes: 0 (max 0)  mis-ordered: 0 bytes
                        Current receive queue size in packets: 0 (max 0)
```

```
Timer          Starts    Wakeups         Next(msec)
Retrans        103448          8                0
SendWnd             0          0                0
TimeWait            1          0                0
AckHold        106815     106545                0
KeepAlive           1          0                0
PmtuAger            0          0                0
GiveUp              0          0                0
Throttle            0          0                0
FirstSyn            0          0                0

iss: 161240548   snduna: 163206936   sndnxt: 163206936
sndmax: 163206936   sndwnd: 63104       sndcwnd: 18120
irs: 3691232436  rcvnxt: 3693473072  rcvwnd: 26099   rcvadv: 3693499171
```

This sample shows only a portion of the full output; the complete output provides additional details.

Convert binary files to readable format using TCP dump file converter

Use this task to convert the dump packet traces in binary files into PCAP and text formats.

Procedure

Step 1 Run the **tcp dump-file convert all-formats all** command to convert the dump packet traces in binary files into PCAP and text formats.

Example:

```
RP/0/RP0/CPU0:ios# tcp dump-file convert all-formats all
                        ascii file is saved at :
/harddisk:/decoded_dumpfiles/text_tcpdump_peer_all_node0_RP0_CPU0_2024_3_19_10_8_53.462070.txt
                        pcap file is saved at :
/harddisk:/decoded_dumpfiles/pcap_tcpdump_peer_all_node0_RP0_CPU0_2024_3_19_10_8_40.154838.pcap
                        [OK]
```

By default, the system stores the converted files in the "decoded_dumpfiles" folder on the "hard disk".

Use the **location node-id** and **file** *file path* keywords to save the converted TCP dump file to your desired location.

For example, **tcp dump-file convert all-formats all location** *0/RP0/CPU0* **file** */harddisk:/demo2* .

```
RP/0/RP0/CPU0:ios# tcp dump-file convert all-formats all location 0/RP0/CPU0 file /harddisk:/demo2
                        ascii file is saved at : /harddisk:/demo2.txt
                        pcap file is saved at : /harddisk:/demo2.pcap
                        [OK]
```

Step 2 Run the **run cat text** *file path* command to view the converted text file in the CLI.

Example:

```
RP/0/RP0/CPU0:ios# run cat
/harddisk:/decoded_dumpfiles/text_tcpdump_peer_all_node0_RP0_CPU0_2024_3_19_10_8_53.462070.txt
                        Filename: 2024_3_19_10_8_53.462070

                        ============================================================
                        Connection state is CLOSED, I/O status: 0, socket status: 103
                        PCB 0x0000000000f47a80, SO 0xf476d0, TCPCB 0xf6a370, vrfid 0x60000000,
                        Pak Prio: Medium, TOS: 192, TTL: 255, Hash index: 563
                        Local host: 14:11:11::1, Local port: 47743 (Local App PID: 19579)
```

```
Foreign host: 14:11:11::2, Foreign port: 179
(Local App PID/instance/SPL_APP_ID: 19579/1/0)

Current send queue size in bytes: 0 (max 0)
Current receive queue size in bytes: 0 (max 0)  mis-ordered: 0 bytes
Current receive queue size in packets: 0 (max 0)

Timer          Starts    Wakeups         Next(msec)
Retrans            70          2                 0
SendWnd             0          0                 0
TimeWait            2          0                 0
AckHold            66         61                 0
KeepAlive           1          0                 0
PmtuAger            0          0                 0
GiveUp              0          0                 0
Throttle            0          0                 0
FirstSyn            1          1                 0

iss: 3113104891  snduna: 3113106213  sndnxt: 3113106213
sndmax: 3113106213  sndwnd: 31523       sndcwnd: 2832
irs: 4250126727  rcvnxt: 4250128049  rcvwnd: 31448   rcvadv: 4250159497
```

This sample shows only a portion of the full output; the complete output provides additional details.

Step 3 Run the **scp** command to copy the converted packet traces from the system to your local computer and view the converted PCAP file.