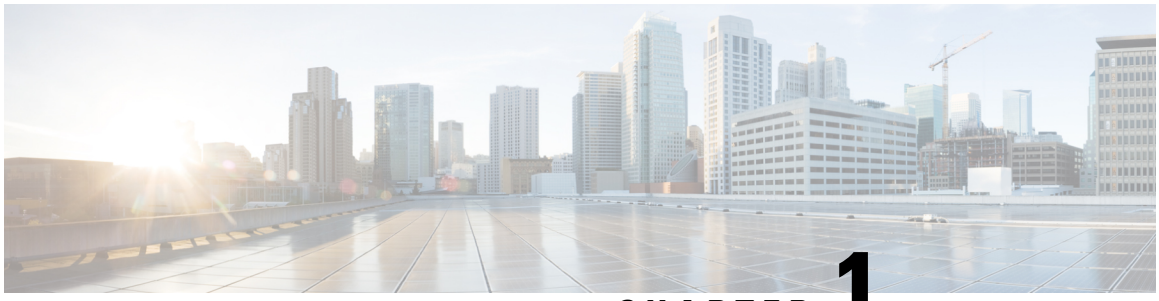
Configuration Guide for Cisco NCS 1014, IOS XR Release 24.1.x

First Published: 2024-03-14

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

CONTENTS



CHAPTER 1

Cisco NCS 1014 Advanced Multihaul Optical Platform - An Overview

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

Cisco NCS 1014 Chassis and Line Cards

The Cisco NCS 1014 chassis is an advanced multihaul optical platform supporting transponders and line system cards. It is a 2RU chassis that delivers a universal transponder solution which provides excellent performance for metro, long-haul and submarine applications.

Cisco NCS 1014 chassis has the following modules:

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

In Release 7.11.1, the Cisco NCS 1014 chassis supports the following line cards:

- NCS1K14-2.4T-K9—2.4T DWDM Transponder Card
- NCS1K14-CCMD-16-C/L—16-port Colorless Mux/Demux Optical Line Card
- NCS1K4-1.2T-K9—1.2T DWDM Transponder Card

From Release 24.1.1, the Cisco NCS 1014 chassis supports the NCS1K14-2.4T-X-K9—2.4T-X DWDM Transponder/Muxponder Card

1.2T Card

The 1.2T DWDM line card is a transponder that has 12 client ports to deliver 100GE and OTU4 client traffic. This line card has two trunks that operate at any rate between 100G and 600G in 50G increments. It uses Advanced Encryption Standard with a 256-bit key length (AES256)-based Layer-1 encryption to encrypt client-side data for 100GE and OTU4. The NCS1K4-1.2T-K9 line card is a single-slot unit that supports C-band traffic.

2.4T Line Card

The 2.4T line card 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. This line card delivers 400GE, 100GE, and OTU4 client traffic over two trunk ports operating at speeds ranging from 400G to 1.2T each.

CCMD-16 Line Card

The CCMD-16 optical line card has:

- Two line ports to transmit and receive using the same LC connectors.
- 16 ports for add/drop with LC connector-based interfaces

There are two variants of the optical line card:

- **NCS1K14-CCMD-16-C**

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

- **NCS1K14-CCMD-16-L**

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

Document Objective

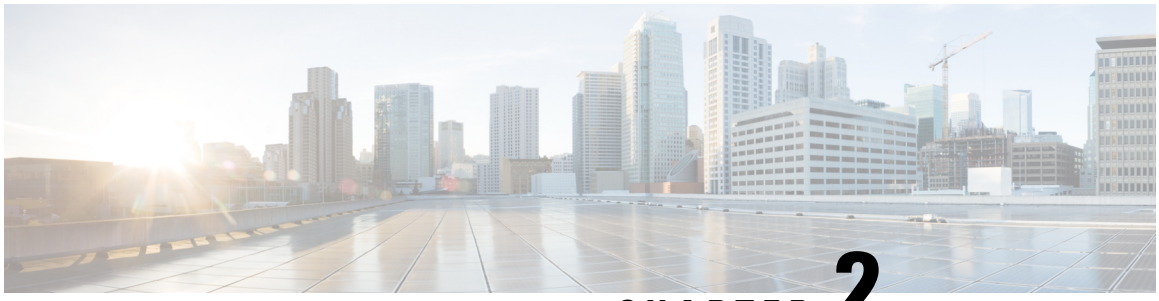
The Cisco NCS 1014 Configuration Guide describes how to configure various card modes for the line cards that are supported in the Cisco NCS 1014 chassis.

Document Organization

This document is organized into the following chapters:

Chapter	Description
Configuring the Card Mode	Describes different card mode configurations and supported data rates for the 1.2T, 2.4T and 2.4T-X line cards.

Chapter	Description
Performance Monitoring	Describes the configuration and retrieval of PM counters for the optics, Ethernet, coherent DSP, OCH, and OMS controller types in flex-bin, 30-second, 15-minute, or 24-hour intervals.



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 Card

The following section describes the supported configurations and procedures to configure the card modes on the line card.

Card Modes

The line cards support module and slice configurations.

The line cards have two trunk ports (0 and 1) and 12 client ports (2 through 13) each. You can configure the line card in two modes:

- Muxponder—In this mode, both trunk ports are configured with the same trunk rate. The client-to-trunk mapping is in a sequence.
- Muxponder slice—In this mode, each trunk port is configured independent of the other with different trunk rates. The client-to-trunk mapping is fixed. For Trunk 0, the client ports are 2 through 7. For Trunk 1, the client ports are 8 through 13.

Sub 50G Configuration

You can configure the sub 50G or coupled mode on the line card only in the muxponder mode. The following table displays the port configuration for the supported data rates.

Trunk Data Rate (per trunk)	Total Configured Data rate	Card Support	Trunk Ports	Client Ports for Trunk 0 (100G)	Shared Client Port (50G per trunk)	Client Ports for Trunk 1 (100G)
50G	100G	1.2T	0, 1	-	2	-
150G	300G	1.2T	0, 1	2	3	4
350G	700G	1.2T	0, 1	2, 3, 4	5	6, 7, 8
450G	900G	1.2T	0, 1	2, 3, 4, 5	6	7, 8, 9, 10
550G	1.1T	1.2T	0, 1	2, 3, 4, 5, 6	7	8, 9, 10, 11, 12

1.2T line card supports an alternate port configuration for Sub 50G (split client port mapping) that you configure using CLI. The following table displays the port configuration for the supported data rates.

Trunk Data Rate (per trunk)	Total Configured Data rate	Card Support	Trunk Ports	Client Ports for Trunk 0 (100G)	Shared Client Port (50G per trunk)	Client Ports for Trunk 1 (100G)
50G	100G	1.2T	0, 1	-	7	-
150G	300G	1.2T	0, 1	2	7	8
250G	500G	1.2T	0, 1	2, 3	7	8, 9
350G	700G	1.2T	0, 1	2, 3, 4	7	8, 9, 10
450G	900G	1.2T	0, 1	2, 3, 4, 5	7	8, 9, 10, 11
550G	1.1T	1.2T	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

The following restrictions apply to the coupled mode configuration:

- Both trunk ports must be configured with the same bits-per-symbol or baud rate and must be sent over same fiber and direction.
- The chromatic dispersion must be configured to the same value for both trunk ports.
- When trunk internal loopback is configured, it must be done for both trunk ports. Configuring internal loopback on only one trunk results in traffic loss.

- 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

You can configure the trunk port to client port mapping for sub 50G data ratesin the default mode or in the split client port mapping mode.

To configure the split client port mapping, use the following commands.

configure

hw-module location *location* **mxponder**

split-client-port-mapping

commit

The following 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
```

To remove the split client port-mapping configuration and configure default client port mapping, use the following commands.

configure

hw-module location *location* **mxponder**

no split-client-port-mapping

commit

The following is a sample in which split client port-mapping configuration is removed.

```
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
```

Verifying the Port Mapping Configuration

The following is a sample ouput of the split client port-mapping.

```
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 following is a sample ouput of the default client port mapping.

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

Supported Data Rates

The following data rates are supported on the line card.

The following table displays the client and trunk ports that are enabled for the muxponder configuration.

Trunk Data Rate	Card Support	Client Data Rate (100GE, OTU4)	Trunk Ports	Client Ports
100	1.2T	100GE, OTU4	0	2
200	1.2T	100GE, OTU4	0, 1	2, 3, 4, 5

Trunk Data Rate	Card Support	Client Data Rate (100GE, OTU4)	Trunk Ports	Client Ports
300	1.2T	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7
400	1.2T	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7, 8, 9
500	1.2T	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11
600	1.2T	100GE, OTU4	0, 1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

The following table displays the client and trunk ports that are enabled for the muxponder slice 0 configuration.

Trunk Data Rate	Card Support	Client Data Rate	Trunk Ports	Client Ports
100	1.2T	100, OTU4	0	2
200	1.2T	100, OTU4	0	2, 3
300	1.2T	100, OTU4	0	2, 3, 4
400	1.2T	100, OTU4	0	2, 3, 4, 5
500	1.2T	100, OTU4	0	2, 3, 4, 5, 6
600	1.2T	100, OTU4	0	2, 3, 4, 5, 6, 7

The following table displays the client and trunk ports that are enabled for the muxponder slice 1 configuration.

Trunk Data Rate	Card Support	Client Data Rate	Trunk Ports	Client Ports
100	1.2T	100, OTU4	1	8
200	1.2T	100, OTU4	1	8, 9
300	1.2T	100, OTU4	1	8, 9, 10
400	1.2T	100, OTU4	1	8, 9, 10, 11
500	1.2T	100, OTU4	1	8, 9, 10, 11, 12
600	1.2T	100, OTU4	1	8, 9, 10, 11, 12, 13

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

The following table displays the trunk parameter ranges for the 1.2T 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

Configuring the Card Mode

You can configure the line card in the module (muxponder) or slice configuration (muxponder slice).

To configure the card in the muxponder mode, use the following commands.

configure

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** }

commit

To configure the card in the muxponder slice mode, use the following commands. **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** }

commit

Examples

The following 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
```

The following 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
```

The following 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
```

The following 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
```

The following 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
```

Verifying the Card Configuration

```
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
```

The following is a sample ouput 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
```

The following is a sample ouput 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
```

The following is a sample ouput 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
```

The following is a sample ouput 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
```

The following 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
```

Use the following command to clear alarm statistics on the optics or coherent DSP controller.

clear counters controller *controllertype R/S/I/P*

The following 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 line card and the other way round. In this mode, only the trunk optics controller and coherentDSP controllers are created.

Configuring the Card in Regen Mode

The supported trunk rates for 1.2T card is100G to 600G in multiples of 100G.

To configure regen mode on 1.2T, use the following commands:

configure

hw-module location *location*

regen

trunk-rate *trunk-rate*

commit

exit

Example

The following is a sample to configure the regen mode on 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
RP/0/RP0/CPU0:ios(config-regen)#commit
RP/0/RP0/CPU0:ios(config-regen)#exit
```

Verifying the Regen Mode

The following is a sample to verify the regen mode.

show hw-module location *location* **regen**

```
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.

Configuring the BPS

You can configure the Bits per Symbol (BPS) to 3.4375 to support 300G trunk configurations on 75 GHz networks using the following commands:

configure

controller optics *R/S/I/P* **bits-per-symbol value**

commit

The following 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
```

Supported Baud Rates

Table 1: 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

Viewing BPS and Baud Rate Ranges

To view the the BPS for a specific range use the following command:

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

```
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
```

To view the baud for a specific range use the following command:

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

```
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
```

Configuring the Trunk Rate for BPSK

You can configure trunk rates of 50G, 100G, and 150G to support Binary Phase-Shift Keying (BPSK) modulation. The BPSK modulation enables information to be carried over radio signals more efficiently.

You can configure trunk rates for BPSK using CLI, NetConf YANG, and OC models.

The following table list the 50G, 100G, and 150G 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

To configure the trunk rate for BPSK modulation, enter the following commands:

configure

hw-module location *location* **mxponder**

trunk-rate {50G | 100G | 150G}

commit

The following 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
```

Viewing the BPSK Trunk Rate Ranges

To view the trunk rate configured for the BPSK modulation, use the following **show** commands:

```
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
```

2.4T and 2.4TX Line Cards

The following section describes the supported configurations and procedures to configure the card modes on the 2.4T and 2.4TX line cards.

Card Modes

The line cards support muxponder slice and muxponder mode configurations.

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

- Muxponder slice—In this mode, each trunk port is configured 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—In this mode, the card allows you to configure both trunk ports with the same trunk rate. The client-to-trunk mapping is fixed.

Note The muxponder mode is supported on the 2.4TX card only.

Supported Data Rates

The following data rates are supported on the 2.4T and 2.4TX line cards.

- 400G
- 500G
- 600G
- 800G

- 1000G
- 1200G

Muxponder Slice Data Rates

The following table displays the client and trunk ports that are enabled for the muxponder slice 0 (Trunk 0) configuration.

Table 2: Slice 0 Port Configurations

Trunk Data Rate	Supported Cards	Client Data Rate	Client Pluggable	Trunk Ports	Client Ports
Configure the 400G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE	QDD-400G[1]	0	1
		4x100GE	QDD-4x100G[2]		
Configure the 500G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE + 1x100GE	QDD-400G[1] + QDD-4x100G[2]	0	1, 2
		5x100GE	2x QDD-4x100G[2]		
Configure the 600G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE + 2x100GE	QDD-400G[1] + QDD-4x100G[2]	0	1, 2
		6x100GE	2x QDD-4x100G[2]		
Configure the 800G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	2x400GE	2x QDD-400G[1]	0	1, 2
		400GE + 4x100GE	QDD-400G[1] + QDD-4x100G[2]		
		8x100GE	2x QDD-4x100G[2]		
Configure the 1000G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	2x400GE + 2x100GE	2x QDD-400G[1] + 2x QDD-4x100G[2]	0	1, 2, 3
		10x100GE	3x QDD-4x100G[2]		

Trunk Data Rate	Supported Cards	Client Data Rate	Client Pluggable	Trunk Ports	Client Ports
Configure the 1200G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	3x400GE	3x QDD-400G[1]	0	1, 2, 3
		2x400GE + 4x100GE	2x QDD-400G[1] + QDD-4x100G[2]		
		400GE + 8x100GE	QDD-400G[1] + 2x QDD-4x100G[2]		
		12x100GE	3x QDD-4x100G[2]		

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

The following table displays the client and trunk ports that are enabled for the muxponder slice 1 (Trunk 7) configuration.

Table 3: Slice 1 Port Configurations

Trunk Data Rate	Card Support	Client Data Rate	Client Pluggable	Trunk Ports	Client Ports
Configure the 400G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE	QDD-400G[3]	7	4
		4x100GE	QDD-4x100G[4]		
Configure the 500G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE + 1x100GE	QDD-400G[3] + QDD-4x100G[4]	7	4, 5
		5x100GE	2x QDD-4x100G[4]		
Configure the 600G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE + 2x100GE	QDD-400G[3] + QDD-4x100G[4]	7	4, 5
		6x100GE	2x QDD-4x100G[4]		
Configure the 800G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	2x400GE	2x QDD-400G[3]	7	4, 5
		400GE + 4x100GE	QDD-400G[3] + QDD-4x100G[4]		
		8x100GE	2x QDD-4x100G[4]		
Configure the 1000G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	2x400GE + 2x100GE	2x QDD-400G[3] + QDD-4x100G[4]	7	4, 5, 6
		10x100GE	3x QDD-4x100G[4]		

Trunk Data Rate	Card Support	Client Data Rate	Client Pluggable	Trunk Ports	Client Ports
Configure the 1200G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	3x400GE	3x QDD-400G[3]	7	4, 5,6
		2x400GE + 4x100GE	2x QDD-400G[3] + QDD-4x100G[4]		
		400GE + 8x100GE	QDD-400G[3] + 2x QDD-4x100G[4]		
		12x100GE	3x QDD-4x100G[4]		

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

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

The following table displays the baud rate ranges for each trunk rate in the 2.4T card.

Table 4: 2.4T Card Baud Rate Ranges

Trunk Payload	Minimum Baud Rate (GBd)	Maximum Baud Rate (GBd)
400G	43.34518	130.4647
500G	49.61196	147.7235
600G	59.53435	148.0555
800G	79.37913	148.0555
1000G	99.22392	148.0555
1200G	119.0687	148.0555

The following table displays the trunk parameter ranges for the 2.4TX card.

Table 5: 2.4TX Card Trunk Parameter Ranges

Trunk Payload	Minimum Baud Rate (GBd)	Maximum Baud Rate (GBd)	Default Baud Rate (GBd)	Minimum BPS	Maximum BPS
400G	43.34518	130.4647	127.931418	2.1	4.1
500G	49.61196	147.7235	137.8340588	2.5	5
600G	59.53435	148.0555	137.738007	2.8	5.1
800G	79.37913	148.0555	137.978388	3.5	5.1
1000G	99.22392	148.0555	137.834059	4.3	5.3

Trunk Payload	Minimum Baud Rate (GBd)	Maximum Baud Rate (GBd)	Default Baud Rate (GBd)	Minimum BPS	Maximum BPS
1200G	119.0687	148.0555	137.738007	5.3	5.7

Muxponder Mode Data Rates

The following table displays the data rates that are supported in the muxponder mode.

Table 6: Muxponder Mode Port Configurations

Trunk Rate	Total Configured Rate	Supported Cards	Trunk 0 Client Ports	Shared Client Port	Trunk 7 Client Ports
600G	1200G	2.4TX	1	2	4
1000G	2000G	2.4TX	1, 2	3	4, 5

Muxponder Slice Configuration on the 2.4T and 2.4TX Cards

You can configure the 2.4T and 2.4TX Cards in the slice configuration (muxponder slice) mode.

In the muxponder slice mode, 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 400G for Slice 0 and 600G for Slice 1, then Trunk 0 delivers 400G and Trunk 7 delivers 600G.

The following figure shows the line card slices and the respective trunk-to-client port mapping.

Figure 1: 2.4T Line Card Slices and Ports

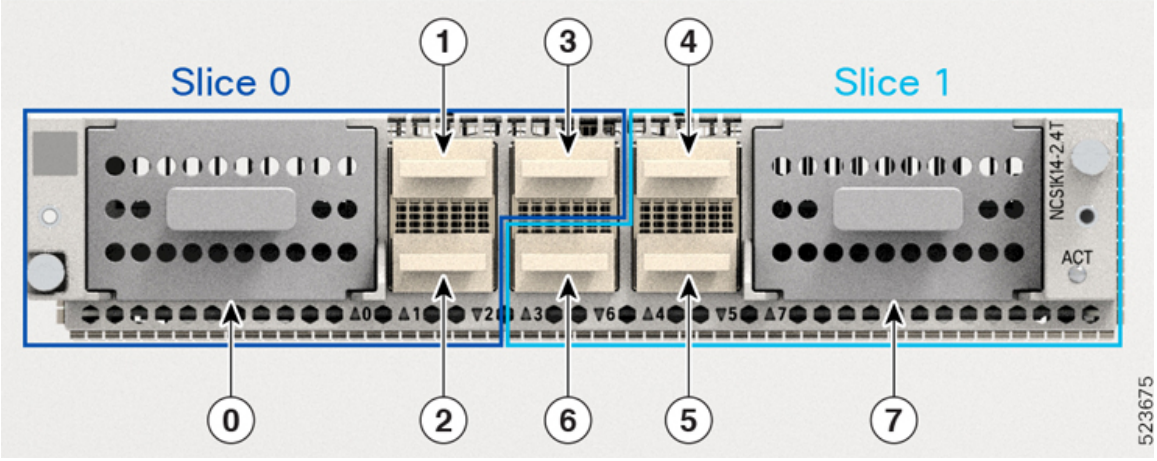
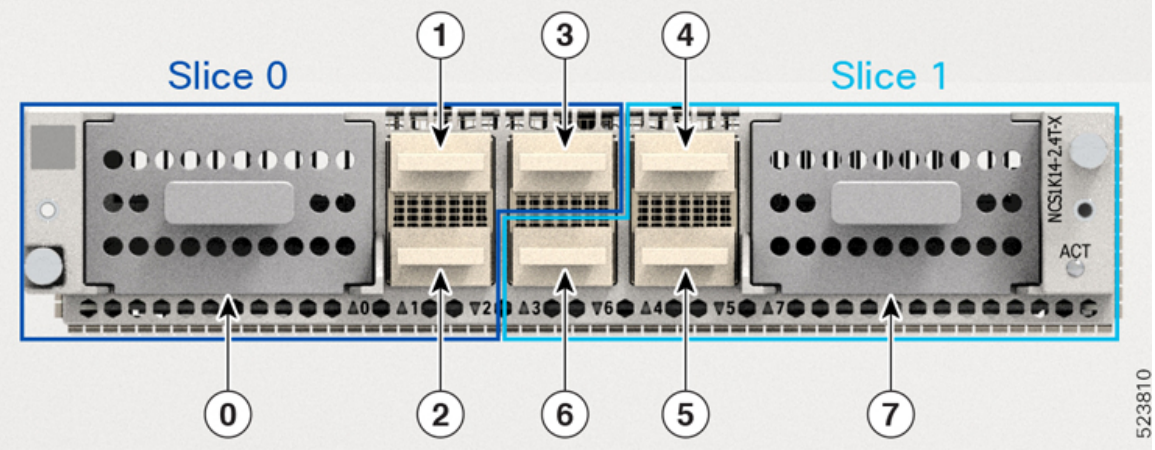


Figure 2: 2.4TX Line Card Slices and Ports



Slice 0		Slice 1	
Trunk Port	Client Ports	Trunk Port	Client Ports
0	1, 2, 3	7	4, 5, 6

To configure the card in the muxponder slice mode, use the following commands.

configuration

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate { 400G | 500G | 600G | 800G | 1000G | 1200G }

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE**

Note QDD-4x100G-LR-S and QDD-400G-DR4-S client pluggable modules support both breakout and nonbreakout channels. For the 100GE breakout configuration, you must use the **lane** keyword. For the 400GE nonbreakout configuration, the **lane** keyword is not required.

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate { 400G | 500G | 600G | 800G | 1000G | 1200G }

client-port-rate *client-port-number* **client-type 400GE**

commit

Note Per trunk port, maximum time that it takes for the laser to be on after completion of datapath configuration is four minutes.

Configure Muxponde Slice Mode on the 2.4T and 2.4TX Cards

The following sections describe the configurations for each trunk rate in the muxponde slice mode on the on the 2.4T and 2.4TX cards.

Configure the 400G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards with 400G trunk rate in the muxponder slice mode, use the following commands:

configuration

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate 400G

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE** OR

client-port-rate *client-port-number* **client-type 400GE**

commit

Examples:

Configuration 1: Using QDD-400G Pluggable

The following sample configuration provisions the 400G trunk rate on the *mxponder-slice 0* mode. This configuration has 400GE client on port 1.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 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 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 400G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 400G Slice Configuration

The following is a sample to verify the 400G trunk rate configured with 400GE client rate in the *mxponder-slice 0* mode.

Output 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:       400GE
Trunk  Bitrate:       400G
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
```

The following is a sample to verify the 400G trunk rate configured with 400GE client rate in the *mxponder-slice 1* mode.

Output Example

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:00:52.022 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       400GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                             100
```

Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 400G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on port 1.

Example Commands

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

The following sample configuration provisions the 400G trunk rate in the *mxponder-slice 1* mode. This configuration has 4x100GE client on port 4.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 400G Slice Configuration

The following is a sample to verify the 400G trunk rate configured with 100GE client rate in the *mxponder-slice 0* mode.

Example Output

```
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:       400G
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
```

The following is a sample to verify the 400G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:06:57.575 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

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

Configure the 500G Trunk in the Muxponder Slice Mode

To configure the 2.4TX card with 500G trunk rate in the muxponder slice mode, use the following commands:

configuration

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate 500G

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE**

client-port-rate *client-port-number* **client-type 400GE**

commit

Configuration 1: Using QDD-400G and QDD-4x100G Pluggable Modules

The following sample configuration provisions the 500G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on port 1 and 1x100GE client on port 2.

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

The following sample configuration provisions the 500G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4 and 1x100GE client on port 5.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 500G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
```

Verify the 500G Slice Configuration

The following is a sample to verify the 500G trunk rate configured with mixed client rate in the *mxponder-slice 0* mode.

Example Output

```
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:       500G
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
```

The following is a sample to verify the 500G trunk rate configured with mixed client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:00:52.022 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       MIXED
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                          100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/7/5/1                        100
```

Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 2.4T card2.4T and 2.4TX cards with the 500G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 1 and 2.

Example Commands

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

The following sample configuration provisions the 2.4T and 2.4TX cards with the 500G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 4 and 5.

Example Commands

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 500G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 500G Slice Configuration

The following is a sample to verify the 500G trunk rate configured with 100GE client in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 16:35:23.121 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       500G
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
```

The following is a sample to verify the 600G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:35:23.121 UTC

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          CoherentDSP0/1/0/7
                               Traffic Split Percentage

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

Configure the 600G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards with 600G trunk rate in the muxponder slice mode, use the following commands:

configuration

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate 600G

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE**

client-port-rate *client-port-number* **client-type 400GE**

commit

Configuration 1: Using QDD-400G and QDD-4x100G Pluggable Modules

The following sample configuration provisions the 2.4T and 2.4TX cards with the 600G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on port 1 and 2x100GE client on port 2.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
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 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards with the 600G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4 and 2x100GE client on port 5.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 600G Slice Configuration

The following is a sample to verify the 600G trunk rate configured with mixed client rate in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 16:20:04.745 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
```

The following is a sample to verify the 600G trunk rate configured with mixed client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:20:04.745 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:        MIXED
Trunk  Bitrate:       600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                          100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/7/5/1                        100
HundredGigECtrlr0/1/0/5/2       ODU-FLEX0/1/0/7/5/2                        100
```

Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 2.4T and 2.4TX cards with the 600G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 1 and 2.

Example Commands

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards with the 600G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 4 and 5.

Example Commands

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 600G Slice Configuration

The following is a sample to verify the 600G trunk rate configured with 100GE client in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 16:35:23.121 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
```

The following is a sample to verify the 600G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:35:23.121 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

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

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

Configure the 800G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards with 800G trunk rate in the muxponder slice mode, use the following commands:

configuration

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate 800G

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE**

client-port-rate *client-port-number* **client-type 400GE**

commit

Configuration 1: Using QDD-400G and QDD-4x100G Pluggable Modules

The following sample configuration provisions the 2.4T and 2.4TX cards with the 800G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on port 1 and 4x100GE client on port 2.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
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 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards with the 800G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4 and 4x100GE client on port 5.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 800G Slice Configuration

The following is a sample to verify the 800G trunk rate configured with mixed client rates in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 15:17:14.082 UTC

Location:             0/1/NXR0
```

```
Slice ID:             0
Client Bitrate:       MIXED
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

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

The following is a sample to verify the 800G trunk rate configured with mixed client rates in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:20:51.482 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       MIXED
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                    Mapper/Trunk Port          CoherentDSP0/1/0/7
                               Traffic Split Percentage

HundredGigECtrlr0/1/0/5/1      ODU-FLEX0/1/0/7/5/1                        100
HundredGigECtrlr0/1/0/5/2      ODU-FLEX0/1/0/7/5/2                        100
HundredGigECtrlr0/1/0/5/3      ODU-FLEX0/1/0/7/5/3                        100
HundredGigECtrlr0/1/0/5/4      ODU-FLEX0/1/0/7/5/4                        100
FourHundredGigECtrlr0/1/0/4    ODU-FLEX0/1/0/7/4                          100
```

Configuration 2: Using QDD-400G Pluggable Module

The following sample configuration provisions the 2.4T and 2.4TX cards with the 800G trunk rate on the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1and 2.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards with the 800G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4 and 5.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 800G Slice Configuration

The following is a sample to verify the 800G trunk rate configured with 400GE client rate in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 13:51:31.599 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       400GE
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

FourHundredGigECtrlr0/1/0/1     ODU-FLEX0/1/0/0/1                             100
FourHundredGigECtrlr0/1/0/2     ODU-FLEX0/1/0/0/2                             100
```

The following is a sample to verify the 800G trunk rate configured with 400GE client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 14:50:32.407 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       400GE
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                             100
FourHundredGigECtrlr0/1/0/5     ODU-FLEX0/1/0/7/5                             100
```

Configuration 3: Using 4x100GE Pluggable

The following sample configuration provisions the 2.4T and 2.4TX cards with the 800G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 1 and 2.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 3 client-type 100GE
```

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

The following sample configuration provisions the 2.4T and 2.4TX cards with the 800G trunk rate in the *mxponder-slice 1* mode. This configuration has 4x100GE client on ports 4 and 5.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
Thu Nov 16 15:01:28.989 UTC
```

Verify the 800G Slice Configuration

The following is a sample to verify the 800G trunk rate configured with 100GE client rate in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 14:59:39.538 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-FLEX0/1/0/0/1/1                         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
HundredGigECtrlr0/1/0/2/3       ODU-FLEX0/1/0/0/2/3                         100
HundredGigECtrlr0/1/0/2/4       ODU-FLEX0/1/0/0/2/4                         100
```

The following is a sample to verify the 800G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:11:45.618 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       800G
```

```
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

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

Configure the 1000G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards with the 1000G trunk rate in the muxponder slice mode, use the following commands:

configuration

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate 1000G

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE**

client-port-rate *client-port-number* **client-type 400GE**

commit

Configuration 1: Using QDD-400G and QDD-4x100GE Pluggable Modules

The following sample configuration provisions the 2.4T and 2.4TX cards with the 1000G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1, 2 and 2x100GE client on port 3.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards with the 1000G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4, 5 and 2x100GE client on port 6.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 1000G Slice Configuration

The following is a sample to verify the 1000G configured with mixed client rate in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 15:17:14.082 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       MIXED
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

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

The following is a sample to verify the 1000G trunk rate configured with mixed client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:20:51.482 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       MIXED
Trunk  Bitrate:       1000G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/6/1       ODU-FLEX0/1/0/7/6/1                         100
HundredGigECtrlr0/1/0/6/2       ODU-FLEX0/1/0/7/6/2                         100
FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                           100
FourHundredGigECtrlr0/1/0/5     ODU-FLEX0/1/0/7/5                           100
```

Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 2.4T and 2.4TX cards with the 1000G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 1, 2, and 3.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
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
```

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

The following sample configuration provisions the 2.4T and 2.4TX cards with the 1000G trunk rate in the *mxponder-slice 1* mode. This configuration has 4x100GE client on ports 4, 5, and 6.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
Thu Nov 16 15:01:28.989 UTC
```

Verify the 1000G Slice Configuration

The following is a sample to verify the 1000G trunk rate configured with 100GE client rate in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 14:59:39.538 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-FLEX0/1/0/0/1/1                          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
HundredGigECtrlr0/1/0/2/3       ODU-FLEX0/1/0/0/2/3                          100
HundredGigECtrlr0/1/0/2/4       ODU-FLEX0/1/0/0/2/4                          100
HundredGigECtrlr0/1/0/3/1       ODU-FLEX0/1/0/0/3/1                          100
HundredGigECtrlr0/1/0/3/2       ODU-FLEX0/1/0/0/3/2                          100
```

The following is a sample to verify the 1000G trunk rate configure with 100GE client rate configured in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:11:45.618 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       1000G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

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

Configure the 1200G Trunk in the Muxponder Slice Mode

To configure the 2.4TX card with 1200G trunk rate in the muxponder slice mode, use the following commands:

configuration

hw-module location *location* **mxponder-slice** *mxponder-slice-number*

trunk-rate 1200G

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE**

client-port-rate *client-port-number* **client-type 400GE**

commit

Configuration 1: Using QDD-400G and QDD-4x100G Pluggable Modules

The following sample configuration provisions the 2.4TX card with the 1200G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1, 2 and 4x100GE client on port 3.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1200G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 1 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 2 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 3 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 4 client-type 1x100GE
```

The following sample configuration provisions the 2.4TX card with the 1200G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4, 5, and 4x100GE client on port 6.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1100G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 1 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 2 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 3 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 4 client-type 1x100GE
```

Verify the 1200G Slice Configuration

The following is a sample to verify the 1200G trunk rate with mixed client rate configured in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 15:17:14.082 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       MIXED
Trunk  Bitrate:       1200G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

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

The following is a sample to verify the 1200G trunk rate configured with mixed client rate in *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:20:51.482 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       MIXED
Trunk  Bitrate:       1200G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/6/1       ODU-FLEX0/1/0/7/6/1                         100
HundredGigECtrlr0/1/0/6/2       ODU-FLEX0/1/0/7/6/2                         100
HundredGigECtrlr0/1/0/6/3       ODU-FLEX0/1/0/7/6/3                         100
HundredGigECtrlr0/1/0/6/4       ODU-FLEX0/1/0/7/6/4                         100
FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                           100
FourHundredGigECtrlr0/1/0/5     ODU-FLEX0/1/0/7/5                           100
```

Configuration 2: Using QDD-400G Pluggable Module

The following sample configuration provisions the 2.4TX card with the 1200G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1, 2, and 3.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1200G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 client-type 400GE
```

The following sample configuration provisions the 2.4TX card with the 1200G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4, 5, and 6.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1200G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 client-type 400GE
```

Verify the 1200G Configuration

The following is a sample to verify the 1200G trunk rate configured with 100GE client rate in the *mxponder-slice 0* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 14:59:39.538 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       1200G
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
HundredGigECtrlr0/1/0/2/3      ODU-FLEX0/1/0/0/2/3                        100
HundredGigECtrlr0/1/0/2/4      ODU-FLEX0/1/0/0/2/4                        100
HundredGigECtrlr0/1/0/3/1      ODU-FLEX0/1/0/0/3/1                        100
HundredGigECtrlr0/1/0/3/2      ODU-FLEX0/1/0/0/3/2                        100
HundredGigECtrlr0/1/0/3/3      ODU-FLEX0/1/0/0/3/3                        100
HundredGigECtrlr0/1/0/3/4      ODU-FLEX0/1/0/0/3/4                        100
```

The following is a sample to verify the 1200G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:11:45.618 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       1200G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

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

Muxponder Mode Configuration on the 2.4TX Card

Table 7: Feature History

Feature Name	Release Information	Feature Description
Bandwidth Splitting for Efficient Trunk Port Utilization	Cisco IOS XR Release 24.1.1	The NCS1K14-2.4T-X-K9 card can split traffic on a client port between the two trunk ports. This feature is useful when you configure the line card for a trunk capacity of 600GE (supplied by two 400GE client ports) or 1000GE (supplied by three 400GE client ports), which leaves 200GE surplus data on one client port. In the Muxponder mode, the other trunk port consumes this 200GE surplus data from a specific shared client port, effectively splitting the 400GE client port capacity equally across two trunks. The shared client ports are Port 2 for 600G and Port 3 for 1000G trunk payloads.

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.

For example, if you set the data rate as 600G per trunk, client ports 1 and 4 carry 400GE each and the shared port 2 equally splits the 400GE traffic between the Trunk ports. Hence the Trunk Ports 0 and 7 always carry the same trunk rates in the muxponder mode.

Note In muxponder mode,

- In a point-to-point topology, port 0 in the near end node must be connected to port 0 in the far end node. The same is applicable to port 7 in the near end and port 7 in the far end nodes.
- In a point-to-point topology, the length of the optic fibers that are connected to both the trunk ports (0 and 7) must be the same. The fiber length difference must be less than 500 m. If the fiber length difference is more than 500 m, then the traffic on the split port does not come up.

Example Scenario

For 600G trunk rate, you must configure Port 1 as 400GE, Port 2 as 400GE, and Ports 4 as 400GE. 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 carry 600G data rate each.

Commands

To configure the card in the muxponder mode, use the following commands.

configuration

hw-module location *location* **mxponder**

trunk-rate { 600G | 1000G }

client-port-rate *client-port-number* **client-type 400GE**

client-port-rate *client-port-number* **lane** *lane-number* **client-type 100GE**

commit

Client Rate and Pluggable Combinations

The following table shows the client rate and client pluggable combinations that are supported for the muxponder mode trunk rates.

Table 8: Trunk Rate and Client Pluggable Combinations

Trunk Rate	Client Rate Combination	Client Pluggable Combination	Client Ports	Shared Client Port
600G	Configure the 600G Trunk in the Muxponder Mode	3x QDD-400G	1, 4	2
1000G	Configure the 1000G Trunk in the Muxponder Mode	5x QDD-400G [5]	1, 2, 4, 5	3

[5] QDD-400G refers to QDD-400G-FR4-S, QDD-400G-AOCxM, and QDD-400G-DR4-S pluggables.

Configure the Muxponder Mode on the 2.4TX Card

The following sections describe the configurations for each trunk rate in the muxponder mode.

Configure the 600G Trunk in the Muxponder Mode

Commands

To configure the 2.4TX card in the 600G muxponder mode, use the following commands:

configuration

hw-module location *location* **mxponder**

trunk-rate 600G

client-port-rate *client-port-number* **client-type 400GE**

commit

Configuration

The following is a sample in which the 2.4TX card is configured in the *mxponder* mode for 600G trunk rate. This configuration has 400GE client on ports 1, 2, and 4.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 600G Muxponder Mode

The following is a sample to verify the configured 600G muxponder mode. In the following output, the 400GE client traffic in *port 2 is split* between the two trunk coherentDSP controllers.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0 mxponder

Location:             0/2/NXR0
Client Bitrate:       400GE
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
FourHundredGigECtrlr0/2/0/2     ODU-FLEX0/2/0/0/2                              50
              50
FourHundredGigECtrlr0/2/0/4     ODU-FLEX0/2/0/7/4                               0
             100
```

Configure the 1000G Trunk in the Muxponder Mode

Commands

To configure the 2.4TX card in the 1000G muxponder mode, use the following commands:

configuration

hw-module location *location* **mxponder**

trunk-rate 1000G

client-port-rate *client-port-number* **client-type 400GE**

commit

Configuration

The following is a sample in which the 2.4TX card is configured in the *mxponder* mode for 1000G trunk rate. This configuration has 400GE client on ports 1, 2, 3, 4, and 5.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
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
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

Verify the 1000G Muxponder Mode

The following is a sample to verify the configured 1000G muxponder mode. In the following output, the 400GE client traffic in *port 3 is split* between the two trunk coherentDSP controllers.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0 mxponder

Location:             0/2/NXR0
Client Bitrate:       400GE
Trunk  Bitrate:       1000G
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
FourHundredGigECtrlr0/2/0/2     ODU-FLEX0/2/0/0/2                          100
              0
FourHundredGigECtrlr0/2/0/3     ODU-FLEX0/2/0/0/3                           50
             50
FourHundredGigECtrlr0/2/0/4     ODU-FLEX0/2/0/7/4                            0
            100
FourHundredGigECtrlr0/2/0/5     ODU-FLEX0/2/0/7/5                            0
            100
```

QXP Card

Table 9: 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 10: 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

Slice	Trunk Port	Client Port
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.

Supported Data Rates for QXP Card

The following table displays 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-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

Configure 400G Transponder Mode

Use the following commands to configure and provision 400G TXP.

hw-module location *location*

mxponder-slice *slice-number*

trunk-rate 400G

trunk-mode [ZR | OR]

client-port-rate *port-number***client-type 400GE**

The following is a sample configuration of configuring 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 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 client-type 400GE
```

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

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/0 mxponder-slice 0
Sat Jun 25 21:32:58.799 UTC

Location:             0/0
```

```
Slice ID:             0
Client Bitrate:       400GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                    Mapper/Trunk Port          CoherentDSP0/0/0/0
                               Traffic Split Percentage

FourHundredGigECtrlr0/0/0/1                 -                           100
```

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

Configure 400G Muxponder Mode

Use the following commands to configure and provision 400G MXP.

hw-module location *location*

mxponder-slice *slice-number*

trunk-rate 400G

client-port-rate *port-number* **lane** *lane-number***client-type 100GE**

The following is a sample configuration of configuring 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
```

The following 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 the following commands to configure and provision 2x100G MXP.

hw-module location *location*

mxponder-slice *slice-number*

trunk-rate 200G

client-port-rate *port-number* **lane** *lane-number***client-type 100GE**

The following is a sample configuration of configuring a 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
```

The following 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
```

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

Table 11: 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

The following 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.

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 operates only as an Ethernet transponder.

Use the following commands to configure OTN data path on the Bright ZR+ pluggable optical modules. The **trunk-mode OR** refers to OpenROADM.

hw-module location *location*

mxponder-slice 1 *slice-number*

trunk-mode OR

trunk-rate *rate*

Use the following commands to configure Ethernet data path on the Bright ZR+ pluggable optical modules.

hw-module location *location*

mxponder-slice 1 *slice-number*

trunk-mode ZR

trunk-rate *rate*

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

The following is a sample configuration of configuring 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
```

The following is a sample configuration of configuring 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
```

The following is a sample configuration of setting 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
```

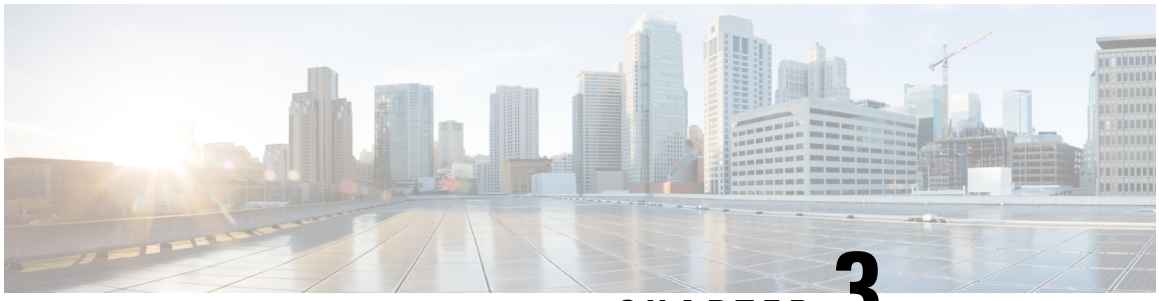
The following is a sample output of the **show controllers optics** command, with the transmit power set to 0 dBm.

```
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
```

Table 12: Operating Modes Supported for Bright ZR+ Pluggable Modules on QXP Card

Operating mode	Modulation	FEC
4x100GE MXP	16-QAM	CFEC

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



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.

AINS

The Automatic-In-Service (AINS) feature allows the controller to automatically move to the automatic-in-service state after the maintenance window is completed. A soak time period is associated with the AINS state. The controller automatically moves to the In-Service state after the soak time period is

completed. During the AINS maintenance window, alarms are not propagated to the EMS/NMS monitoring system.

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

AINS States

The following 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 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.

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.

Soak Time Period

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

All alarms are suppressed during the AINS state. When the optical and ethernet alarms are raised on the port during the soak time period, the AINS state moves to Pending. These alarms are not displayed in the output of the **show alarms brief card location 0/RP0/CPU0 active** command but in the output of the **show alarms brief card location 0/RP0/CPU0 conditions** command. When all the alarms clear, the soak time period starts, and the AINS state moves to Running. When the soak time period expires, the port moves to IS state.

Configuring AINS

To configure AINS on a controller, use the following command:

automatic-in-service controller *controller rate controller* **hours** *hours* **minutes** *minutes*

The following is a sample to configure AINS on a controller.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios#automatic-in-service controller coherentDSP 0/0/0/12 hours 0 minutes 15
RP/0/RP0/CPU0:ios(config)#commit
```

To configure AINS on a muxponder slice, use the following command:

configure

hw-module location *location* **mxponder-slice** *slice-number* **client-port-ains-soak hours** *hours* **minutes** *minutes*

commit

The following is a sample in which slice 0 client ports are configured with AINS with soak time period specified to be 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
```

To configure AINS globally, use the following command:

ains-soak hours *hours* **minutes** *minutes*

The following 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
```

Disabling AINS

To disable AINS on all muxponder client ports, set the hours and minutes to 0. Use the following commands:

configure

hw-module location *location* **mxponder-slice** *slice-number* **client-port-ains-soak hours** *hours* **minutes** *minutes*

commit

The following 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
```

To disable AINS on a muxponder slice, set the hours and minutes to 0. Use the following command:

configure

hw-module location *location* **mxponder-slice** *slice-number* **client-port-ains-soak hours** *hours* **minutes** *minutes*

commit

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

```
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
```

Displaying the AINS Configuration

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.

This example displays 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
```

This example displays 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
```

This example displays 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 used for controlling errors during data transmission. This feature can be enabled on 1.2T and 2.4T cards and works by adding data redundancy to the transmitted message using an algorithm. This redundancy allows the receiver to detect and correct a limited number of errors occurring anywhere in the message, instead of having to ask the transmitter to resend the message.

FEC is automatically enabled on the pluggables installed in the Cisco NCS 1014. When you upgrade the software of an NCS 1014 with pluggables in the FEC disabled mode, traffic is affected.

The following sample shows the running FEC configuration on a 2.4T card:

```
RP/0/RP0/CPU0:ios#sh controllers fourHundredGigEctrlr 0/1/0/1
Fri Nov 17 10:01:22.840 UTC
Operational data for interface FourHundredGigECtrlr0/1/0/1:

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: Line
    BER monitoring:
        Not supported
    Forward error correction: Standard (Reed-Solomon)
    Holdoff Time: 0ms
```

Configuring FEC on the Ethernet Controller

To configure FEC on the Ethernet controller, use the following command:

configure

controller HundredGigECtrlr *R/S/I/P* **fec { none | standard }**

commit

The following 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
```

The following 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
```

FEC States for CoherentDSP Controller

The following table lists the FEC states for the coherentDSP controllers.

Table 13: 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.

Configuring FEC on CoherentDSP Controllers

To configure FEC on the CoherentDSP controller, use the following command:

configure

controller coherentDSP *R/S/I/P*

fec {**EnhancedSD15**}

commit

The following 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
```

Verifying FEC on CoherentDSP Controllers

The following 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 2.4TX 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. Hence squelching happens faster when compared to previous releases. Squelch happens for client alarms also like Ingress LF, LOA, and CSF (not for egress client alarms) in addition to trunk fault cases.

To configure laser squelching on 1.2T card, use the following commands:

configure

controller HundredGigECtrlr *Rack/Slot/Instance/Port*

laser-squelch

commit

To configure laser squelching on a 2.4T and 2.4TX cards, use the following commands:

configure

controller HundredGigECtrlr *Rack/Slot/Instance/Port/Lane* | **controller fourHundredGigECtrlr** *Rack/Slot/Instance/Port*

laser-squelch

commit

Note In case of muxponder configuration on the 2.4TX 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.

The following 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
```

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

```
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
```

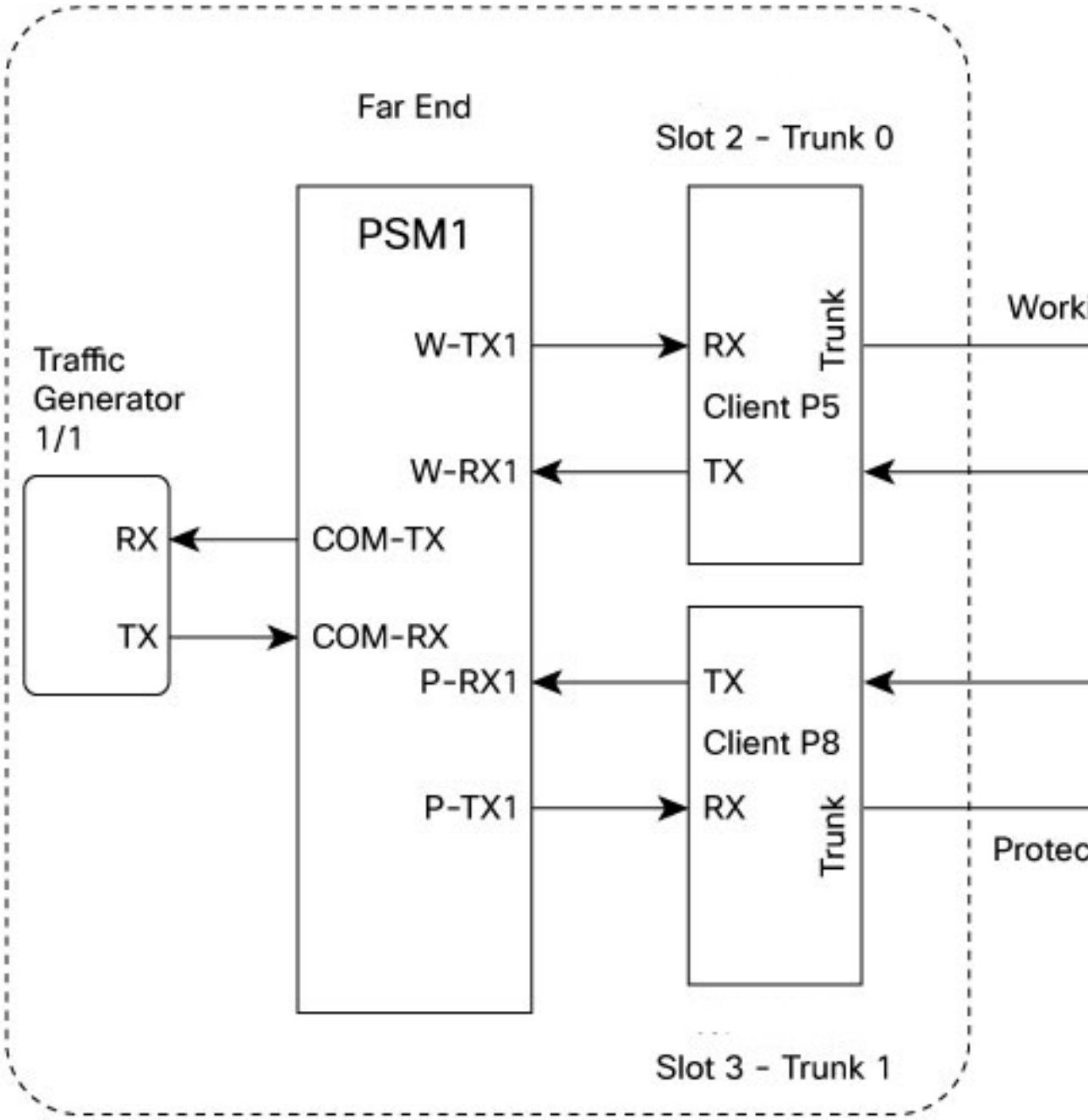
Protection Switching Use Cases

Fast-Squelching provides increased protection switching speed when there is a trunk fault or a client fault. Fast-Squelching is supported on 1.2T cards.

Note Protection Switching is not supported on 2.4T card.

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

Figure 3: 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.

Note

- 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 we can’t use 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.

Idle Insertion

When a fault occurs on the trunk port, you can hold the propagation of local faults using the idle insertion feature. This feature is can be enabled on the ethernet controllers of 1.2T, 2.4T, and 2.4TX 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 Idle Insertion

You can enable the idle insertion feature on a 1.2T card using the following commands:

configure

controller HundredGigECtrlr *Rack/Slot/Instance/Port*

holdoff-time trunk-fault *time-value*

You can enable the idle insertion feature on 2.4T card and 2.4TX card using the following commands:

configure

controller HundredGigECtrlr *Rack/Slot/Instance/Port/Lane* | **controller fourHundredGigECtrlr** *Rack/Slot/Instance/Port*

holdoff-time trunk-fault *time-value*

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

Note In case of muxponder configuration on the 2.4TX card, holdoff timer is not supported on the split ports.

Example 1

The following is a sample for enabling the hold off -timer in 100GE controller:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller HundredGigECtrlr 0/0/0/4/1
RP/0/RP0/CPU0:ios (config-eth-ctrlr)#holdoff-time trunk-fault 3000
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

View Hold Off Timer

To view the hold-off time that is configured on 100GE controller, use the following command:

show controllers hundredGigECtrlr *Rack/Slot/Instance/Port*

Example 2

```
RP/0/RP0/CPU0:ios#show controllers HundredGigECtrlr 0/1/0/1
Fri Feb 22 18:58:06.888 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: Disabled

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: 3000ms
```

Idle Insertion for Ethernet Controllers

Idle insertion for Ethernet controllers feature allows you to perform end-to-end link verification between 100GE or 400GE 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.

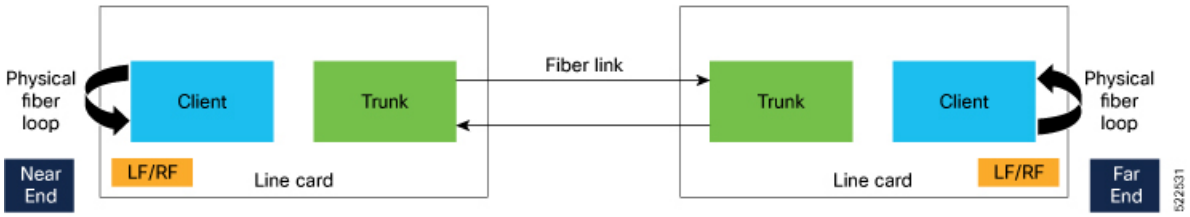
Idle frames can be inserted in both the ingress and egress directions on Ethernet controllers and the LOCAL-FAULT and REMOTE-FAULT alarms are cleared. The performance monitoring counters on the pcs layer are monitored to check for any errors on the link.

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

Recommended Topology for Link Verification

The following diagram describes the recommended topology for link verification:

Figure 4: Topology for Link Verification



The following 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.

Configuring Idle Insertion for Ethernet Controllers

Before You Begin:

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

You can configure this feature by using the following commands:

configure

controller hundredGigECtrlr *Rack/Slot/Instance/Port/*

insert-idle ingress

insert-idle egress

commit

end

The following 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
```

You can disable this feature by using the following commands:

configure

controller hundredGigECtrlr *Rack/Slot/Instance/Port*

no insert-idle ingress

no insert-idle egress

commit

end

Limitation

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:

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

Verifying Idle Insertion Configuration for Ethernet Controllers

To verify the *idle ingress* and *idle egress* that is configured on the Ethernet controllers of a 1.2T card, use the following command:

RP/0/RP0/CPU0:ios# **show controllers hundredGigECtrlr** *Rack/Slot/Instance/Port*

Example

```
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
```

Enable Idle Insertion on QXP Card

You can enable idle insertion on 100GE or 400GE controllers for the QXP card.

Configure Idle Insertion on 100GE Controllers

To configure idle insertion on the 100GE controllers for the QXP card, use the following commands:

configure

controller HundredGigECtrlr *Rack/Slot/Instance/Port*

holdoff-time trunk-fault *time-value*

commit

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

The following is a sample where idle insertion 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
```

The following is a sample to view the idle insertion 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
```

Configure Idle Insertion on 400GE Controllers

To configure idle insertion on the 400GE controllers for the QXP card, use the following commands:

configure

controller fourHundredGigECtrlr *Rack/Slot/Instance/Port*

holdoff-time trunk-fault *time-value*

commit

The following is a sample where idle insertion 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
```

The following is a sample to view the idle insertion 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
```

FlexO GID and IID

The 2.4T and 2.4TX cards use flexible OTN (flexO) interfaces on trunk ports. These flexO interfaces provide a flexible and interoperable mechanism to transport OTU signals by grouping standard lower rate interfaces. Each flexO interface group is identified by a flexO group identification (GID) number, which ranges 1–1,048,576. Each member of a flexO group is identified by a flexO instance identification (IID) number. The IID cannot be changed.

Configuring FlexO GID

To configure flexO GID and IID on the coherentDSP controller, enter the following commands:

configure

controller coherentDSP *R/S/I/P*

flexo

gid <*gid-no*>

commit

The following sample shows how to configure flexO GID on the CoherentDSP controller:

```
P/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
```

Verifying FlexO GID

The following 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
```

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.

The following table describes the global LLDP attributes that the user can configure:

Table 14:

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)

The following 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
```

Verification

You can verify the LLDP configuration using the **show running-config lldp** command.

The output of **show running-config lldp** command is as follows:

```
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
!
```

You can verify the LLDP data using the **show lldp interface** and **show lldp neighbors** commands.

The output of **show lldp interface** command is as follows:

```
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
```

The output of **show lldp neighbors** command is as follows:

```
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.

Enabling LLDP per Management Interface

The following 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
```

Disabling LLDP Transmit and Receive Operations

The following 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
```

The following 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
```

Debugging LLDP Issues

The following 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**

MAC Address Snooping on Client Ports

MAC address snooping allows you to learn the MAC address of the neighbor on 1.2T cards, that is connected to the client ports. You can enable ARP snooping on all client ports and learn the MAC address of neighbors through CLI.

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.

Configuring MAC Address Snooping on Client Ports

You can configure MAC address or ARP snoop on slice in Muxponder slice mode using the following commands.

configure

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

commit

Example

The following 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
```

The following 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
```

The following 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
```

Viewing Neighbor MAC Address

You can view the neighbor's physical address after enabling MAC address or ARP snoop using the following command. MAC address snoop output is enabled after ARP packets are received on the respective 100G client.

show controllers hundredGigEctrlr *R/S/I/P*

The following 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.

Configuring Transmit Shutdown on Trunk Optics Controller

To perform transmit shutdown, enter the following 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
```

Verifying Transmit Shutdown on Trunk Optics Controller

To verify the transmit shutdown details on the trunk optics controller, use the following command:

show controllers optics *R/S/I/P*

Example

Following is an example to view the transmit shutdown details:

```
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, 24TX 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.

Configuring Loopback on the 1.2T Card

To configure the loopback on a 1.2T card, use the following commands:

controller *controllertype Rack/Slot/Instance/Port*

sec-admin-state maintenance

loopback [**internal**]

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

Configuring Loopback on 2.4T Card and 2.4TX Card

To configure the loopback on 2.4T card and 2.4TX card, use the following commands:

controller {**HundredGigECtrlr** *Rack/Slot/Instance/Port/Lane* | **fourHundredGigECtrlr***Rack/Slot/Instance/Port*}

sec-admin-state maintenance

loopback [**line** | **internal**]

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 1

The following 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
```

Example 2

The following 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
```

Example 3

The following 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
```

Configuring Loopback on the QXP Card

Example 1

The following 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
```

The following 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 2

The following 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
```

The following 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 3

The following 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
```

The following 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 4

The following 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
```

The following 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 5

The following 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
```

The following 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
```

Viewing Loopback Configuration Alarm

The following example shows how to view the loopback configuration alarms.

```
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
```

Restore Factory Settings

Note Perform this operation only on the console port.

You can restore the factory settings on the NCS 1014. The entire system configuration, including usernames, passwords, and IP addresses, is removed. You can perform this operation only through the console port and not on the management interface. To restore NCS 1014 to factory settings, use the **commit replace** command. 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.

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 192.0.0.4
sending stop hb
Cause: User initiated graceful reload
VM IP addr sent for reload 192.0.0.4

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. !!!!!!!!!!!!!!!!!!!
```

Headless Mode

During process restarts, CPU reload, or removal of CPU, the NCS 1014 operates in headless mode for up to 72 hours. During this time, traffic is not impacted, although the control plane is not up and running. Fault

propagation continues to operate for failures on client and trunk ports. However, you cannot provision anything nor view operational data with a non-functional CPU. Performance monitoring data based on 15 minutes and 24 hour intervals is not supported with a non-functional CPU.

Trail Trace Identifier

The Trail trace identifier (TTI) feature helps you to identify the signal from the source to the destination within the network. You can configure the TTI sent or expected in ASCII and HEX. When the expected TTI string does not match the received TTI trace string, the controller goes down and the TIM alarm is raised. To configure TTI on the coherent DSP controllers of a 1.2T, 2.4T, or 2.4TX card, use the following commands:

configure

controller coherentDSP *R/S/I/P* **tti** {**sent | expected**} **ascii | hex***tti-string*

commit

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

The following sample displays 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
```

The following is 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
```

The following example 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
```

Configure TTI on QXP Card

You can 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**.

The following 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

To configure TTI, use the following commands:

configure

controller *controller-type R/S/I/P* **tti** {**sent | expected**} {**ascii | sapi ascii | dapi ascii | operator-specific ascii** } *tti-string*

commit

The following 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
```

The following 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
```

The following 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
```

The following 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
```

The following is 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
```

The following 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
```

Chromatic Dispersion

You can 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.

The following 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
600G	BPS=5.2578125	1.2T	-2000 to 2,000 ps/nm

The following table lists the default CD search range for 2.4T, 2.4TX, cards.

Line Rate				Sy
	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 -10000
600G	74000 to -10000	74000 to -10000	74000 to -10000	60000 to -10000
500G	90000 to -10000	90000 to -10000	90000 to -10000	72000 to -10000
400G	—	90000 to -10000	90000 to -10000	72000 to -10000

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.

The following 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
!
```

Chromatic Dispersion Threshold

You can 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.

The following is a sample of configuring the minimum and maximum chromatic dispersion threshold:

To configure the maximum nd minimum acceptable CD, use the following command:

configure

controller optics *R/S/I/P*

cd-high-threshold *cd-high*

cd-low-threshold *cd-low*

commit

The following 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
```

The following 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
```

Transmit Power

To configure transmit power on the trunk (CIM8 and PICO) optics within the *-190* to *+50* range (in units of 0.1dBm), use the following commands:

```
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
```

The following 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
```

Laser Bias Current High Threshold

You can configure the threshold of the laser bias current flowing on the physical pluggable port on the trunk optics controller. The range is 0 to 100%

To configure the laser bias current threshold, use the following command:

configure

controller optics *R/S/I/P*

lbc-high-threshold *lbc-value*

commit

The following sample 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
```

The following sample 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
```

Differential Group Delay Threshold

You can 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).

To configure the maximum acceptable DGD, use the following command:

configure

controller optics *R/S/I/P*

dgd-high-threshold *dgd-value*

commit

The following 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
```

The following 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
```

Optical Signal to Noise Ratio

You can 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).

To configure the minimum acceptable OSNR, use the following command:

configure

controller optics *R/S/I/P*

osnr-low-threshold *osnr-value*

commit

The following 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
```

The following 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
```

Receive Power Threshold

You can 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).

To configure the high and low receive power threshold, use the following command:

configure

controller optics *R/S/I/P*

rx-high-threshold *rx-high*

rx-low-threshold *rx-low*

commit

The following 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
```

The following 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
```

Transmit Power Threshold

You can configure the high and low threshold of the total optical signal power of the transmitted signal on the 1.2T, 2.4TX, and 2.4T cards.

The range is –400 to 300 (in the units of 0.1 dBm).

To configure the high and low transmit power threshold, use the following command:

configure

controller optics *R/S/I/P*

tx-high-threshold *tx-high*

tx-low-threshold *tx-low*

commit

The following sample configures the high transmit 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)#tx-high-threshold 40
RP/0/RP0/CPU0:ios(config-Optics)#tx-low-threshold 20
RP/0/RP0/CPU0:ios(config-Optics)#commit
```

The following sample shows the high transmit 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
```

Frequency

You can configure the frequency on trunk ports of the1.2T, 2.4TX, and 2.4T line cards.

The following table lists the frequency range with grid spacing supported on the line card:

Line Card	Frequency Range (THz)	Default Frequency (THz)	Grid Spacing
1.2T	191.25 to 196.1	193.1	50GHz and 100MHz
1.2TL [6]	186.1 to 190.85	188.5	100MHz
2.4T and 2.4TX	191.25 to 196.1	193.1	50GHz and 100MHz

[6] Only non-ITU channels are supported

To configure the wavelength, use the following commands:

configure

controller optics *Rack/Slot/Instance/Port*

dwdm-carrier {**100MHz-grid frequency** *frequency*} | {**50GHz-grid** [**frequency** *frequency*}

commit

Pseudo Random Binary Sequence

The Pseudo Random Binary Sequence (PRBS) feature feature enables data integrity checks between NCS1014 trunk and client links without generating client traffic.

To ensure that the traffic is error-free during link bring up without relying on the peer port, you must enable the PRBS feature on both the transmitting and receiving ports of your NCS 1014 trunk. The transmitting trunk port creates a bit pattern and sends it to the peer NCS 1014 device. The device then confirms if the sent bit pattern is received. The 1.2T card supports PRBS on the ODU4 controller.

You can configure PRBS on the NCS 1014 trunk and client ports of a 1.2T, 2.4T, and 2.4TX cards.

- **Source mode —** The NCS 1014 at trunk port generates PRBS signal on the line continuously as per the configured PRBS pattern.
- **Sink mode —** The NCS 1014 at trunk port gets locked to the ingress signal according to the configured pattern, analyzes and reports the errors.
- **Source-Sink mode —** The NCS 1014 at trunk port acts as both the PRBS transmitter and receiver, that is, it generates PRBS signal as per the configured pattern, and also gets locked to the ingress signal with the same pattern, and reports the errors.

NCS 1014 trunk port supports the following 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.

Note Interoperability for ethernet PRBS PN23 pattern is not supported in the 2.4T and 2.4TX cards.

Note In case of muxponder configuration, PRBS is not supported on the split ports of the 2.4TX card.

Configuring Pseudo Random Binary Sequence

You can configure PRBS on a coherentDSP or Ethernet controller of a 1.2T, 2.4T, or 2.4TXcard. 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*.

Configure PRBS on CoherentDSP Controller

To configure PRBS on the trunk port of the coherentDSP controller of a 1.2T, 2.4T, or 2.4TX card, use the following configuration commands in the configuration mode:

controller coherentDSP *R/S/I/P*

secondary-admin-state maintenance

prbs mode {**source | sink | source-sink**} **pattern** {**pn31 | pn23 | pn15 | pn7**}

Example to Configure PRBS on CoherentDSP Controller:

Use the following sample configuration to configure PRBS on trunk ports of a coherentDSP controller:

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

Configure PRBS on ODU Controller

You can configure PRBS in the following 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

To enable PRBS on ODU4 controller's trunk ports of a 1.2T card, use these commands in configuration mode:

controller odu4 *R/S/I/P*

secondary-admin-state maintenance

opu prbs mode {**source | sink | source-sink**} **pattern** {**pn31 | pn23 | pn15 | pn7**}

Example to Configure PRBS on ODU Controller:

Following is an example to configure PRBS on client ports of an ODU controller of a 1.2T card:

```
RP/0/RP0/CPU0:ios(config)#controller odu4 0/0/0/4
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern pn23
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Configure PRBS on Ethernet Controller

To configure PRBS on the client ports of an Ethernet controller of the 2.4T and 2.4TX cards, use the following commands in the configuration mode:

controller {**fourHundredGigECtrlr**} *R/S/I/P*

secondary-admin-state maintenance

opu prbs mode {**source | sink | source-sink**} **pattern** {**pn31 | pn23** }

Example to Configure PRBS on Ethernet Controller:

Following is an example to configure PRBS on client ports of an Ethernet controller of a 2.4T card:

```
RP/0/RP0/CPU0:ios(config)#controller FourHundredGigECtrlr 0/3/0/1
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern pn31
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Verifying PRBS

You can monitor the status of Pseudo Random Binary Sequence (PRBS) using the following command:

show controllers coherentDSP | ODU4 | fourHundredGigEctrlr *R/S/I/P* **prbs-details**

Example to view PRBS details on Ethernet controller

Use the following sample configuration to display PRBS details configured on an Ethernet controller:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4
Wed Nov 15 18:39:29.478 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: PN23
      Direction: Line
      Framing: Framed
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled

Phy:
    Media type: Not known
    Statistics:
        FEC:
            Corrected Codeword Count: 2019127152             Valid: True       Start time:
 17:35:46 Wed Nov 15 2023
            Uncorrected Codeword Count: 6                    Valid: True       Start time:
 17:35:46 Wed Nov 15 2023
        PCS:
            Total BIP errors: 0                              Valid: True       Start time:
 17:35:46 Wed Nov 15 2023
            Total frame errors: 0                            Valid: False      Start time:
 17:35:46 Wed Nov 15 2023
            Total Bad SH: 0                                  Valid: False      Start time:
 17:35:46 Wed Nov 15 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
```

Example to view PRBS details on CoherentDSP controller

Use the following sample configuration to display PRBS details configured on a coherentDSP controller:

```
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
```

Example to view PRBS details on ODU controller

Use the following sample configuration to display PRBS details configured on an ODU controller where the PRBS status is displayed as **Not Applicable**, when the mode is **Source**.

```
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

------------------------------------------------
```

Viewing PRBS Performance Monitoring Parameters

To view the PRBS performance monitoring parameters on a coherentDSP, ODU, or Ethernet controller, use the following command:

show controllers coherentDSP | ODU4 *R/S/I/P* **pm** {**current | history** }{**15-min|24-hour**} **prbs**

Following is an example of how to view the cumulative count of PRBS bit errors in the 15-min sampling interval on a CoherentDSP controller:

```
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
```

Following is an example of how to view PRBS performance monitoring parameters for a 15-minute sampling interval on an ODU controller:

```
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
```

The following tables describes the fields of PRBS PM parameters.

Following is an example of how to view PRBS performance monitoring parameters for a 15-minute sampling interval on an Ethernet controller:

```
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
```

The following tables describes the fields of PRBS PM parameters.

Table 15: 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.

Clearing Bit Errors and Lock Time for PRBS

Lock time is the time that is elapsed since the last PRBS lock is detected.

The following sample shows that bit errors are observed during the PRBS test:

```
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

-------------------------------------------------
```

To clear the lock time and bit errors before the PRBS test, use the **clear** command:

```
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
```

The following sample displays the bit errors and lock time are removed.

```
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
```

CCMD-16 Controllers

The NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L cards have two types of controllers. The controllers are OMS and OCH 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.

To view the OCH controllers, run the following command:

Command

```
RP/0/RP0/CPU0:ios#show controllers och ?
```

The following 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
```

To view the OMS controllers, run the following command:

Command

```
RP/0/RP0/CPU0:ios#show controllers oms ?
```

The following output shows the active OMS controller in *Slot 0*.

Output Example

```
0/0/0/0              Oms Interface Instance
```

OCH Controller

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 16: 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
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.

To view the parameters of an OCH controller, use the following command:

Command Example

```
RP/0/RP0/CPU0:ios#show controllers och 0/1/0/1
```

The following output shows the parameters of the OCH controller.

Output Example

```
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
```

OMS Controller

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 17: 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.

To view the parameters of an OMS controller, use the following command:

Command Example

```
RP/0/RP0/CPU0:ios#show controllers oms 0/1/0/0
```

The following output shows the parameters of an OMS controller.

Output Example

```
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 18: 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 19: 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>`	This is used for the gain-range selection for the EDFA located on the COM-Tx port. The default value is 'Normal'. **Note** If you disable the extended gain range mode without configuring the gain value for the normal gain-range mode, then, during the RP reload the following happens: • The egress-ampli-gain configuration is lost. • The EDFA switches to the preconfigured or default value of the normal mode gain. This gain change results in traffic impact. Hence, we recommend that you explicitly configure the gain range mode as normal or extended, and the corresponding gain values for each mode.

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 20: 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.

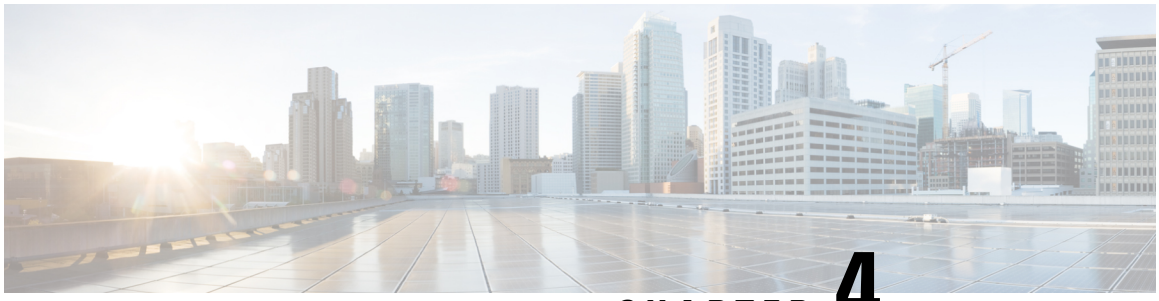
Configure Optical Safety Parameters

Use the following parameters for configuring optical safety.

Table 21: 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.



CHAPTER 4

Performance Monitoring

Performance monitoring (PM) parameters are used by service providers to gather, store, set thresholds for, and report performance data for early detection of network issues. You can configure and retrieve PM counters for the various controllers in 30 second, 15-minute, or 24-hour flex-bin intervals. These parameters simplify troubleshooting operations and enhance data that can be 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 the following commands:

```
process shutdown pm_collector
 run
 cd /misc/config
 rm chkpt_pm_collector_*
 exit
```

Use the following 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
```


Performance Monitoring

Performance monitoring (PM) parameters are used by service providers to gather, store, set thresholds for, and report performance data for early detection of network issues. You can configure and retrieve PM counters for the various controllers in 30 second, 15-minute, or 24-hour flex-bin intervals. These parameters simplify troubleshooting operations and enhance data that can be 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 the following commands:

```
process shutdown pm_collector
 run
 cd /misc/config
 rm chkpt_pm_collector_*
 exit
```

Use the following 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
```

Configuring PM Parameters

You can configure and view the performance monitoring parameters for the Optics, Ethernet, odu-flex, and coherent DSP controllers.

To configure PM parameters, use the following commands.

configure

show controller *controllertype R/S/I/P* { **pm** { **current | history**} { **30-sec 15-min** | | **24-hour** } { **optics** | **ether** | **fec** | **otn | prbs**} **linenumber** }

commit

Examples

The following is a sample in which the performance monitoring parameters of Optics controller are configured for 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
```

The following is a sample in which the performance monitoring parameters of the Ethernet controller are configured for 15-minute intervals.

```
RP/0/RP0/CPU0:chassisA164(config)#controller fourHundredGigECtrlr 0/1/0/4 pm 15-min ether
threshold rx-pkt 1
```

The following is a sample in which performance monitoring parameters of Coherent DSP controller are configured for 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
```

Viewing PM Parameters

To view the performance monitoring parameters for Optics, Ethernet, and Coherent DSP controllers, use the following command:

show controllers *controllertype R/S/I/P* **{ pm { current | history } { 30 sec | 15-min | 24-hour } { optics | ether | fec | otn | prbs}** *linenumber* }

Example 1: The following command 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
```

Example 2: The following command 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 3: The following command 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 4: The following command displays the current performance monitoring for FEC for the Coherent DSP controller for FEC 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
```

Example 5: The following command 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 6: The following command 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 7: The following command 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
```

Example 8: The following command 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 9: The following command 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 10: The following command 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
```

Example 11: The following command 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 12: The following command 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 13: 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 14: The following command 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 15: The following command 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 16: 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 17: Displays the current performance monitoring parameters of the client with 15-minute intervals PCS.

```
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
```

Instantaneous Q-Margin

Scenarios on Instantaneous Q-margin

In the following 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.

- Shutdown or no shutdown on optics
- Trunk rate change
- Fiber cut

To overcome such situations, avoid the initial PM bucket readings while monitoring the instantaneous Q-margin values for these scenarios.

The following sample illustrates that the initial PM bucket readings for specified scenarios are invalid and at a later point the PM buckets readings are valid although the instantaneous Q-margin value is 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
```

Clearing PM Parameters

To clear the performance monitoring parameters for Ethernet and Coherent DSP controllers, use this command:

clear controller *controllertype R/S/I/P* **pm**

Example 1: Clears the PM parameters on the Coherent DSP 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
    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 2: To clear the PM parameters on the Ethernet controller, use the following command:.

```
RP/0/RP0/CPU0:ios#clear controller HundredGigECtrlr 0/0/0/2/1 pm
```

Viewing Ethernet Statistics

To view the PM statistics for the Ethernet controllers, use the following command:

```
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.

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, the following list describes the time that is required to fill all historical buckets of each bucket type, later while fetching PM historical data, no error appears.

- 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 is required to fill 7 historical bucket.

PM counters are updated continuously in current bucket for all bucket types (flex, 30-second, 15-minute, and 24-hour). After the timer expires for the respective bucket type, the current PM data is moved to the historical PM bucket. This process of moving PM data to the historical bucket is called Rollover. After rollover, you can access the current PM data as historical PM data.

In case of deletion or removal of the controller, the PM data is persistent for 3 hours. Unless the controller is brought up within 3 hours, the PM data is cleared because the controller is considered to be not in use.

Limitations

If NCS 1014 reload happens during the rollover time, one of the following scenarios occurs:

- Complete PM bucket is missing and the next PM bucket is marked 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.

Performance Monitoring for NCS1K14-2.4T-X-K9 Card

Performance monitoring (PM) parameters are used by service providers to gather, store, set thresholds for, and report performance data for early detection of network issues. You can configure and retrieve PM counters for 30-second, 15-minute, or 24-hour intervals. These parameters simplify troubleshooting operations and enhance data that can be collected directly from the equipment.

Limitations

On the 2.4TX card in the muxponder mode, PM parameters do not show the Runt and invalid Start Frame Delimiter (SFD) values for the split ports 2 and 3 for 600G and 1000G trunk rates respectively.

Performance Monitoring for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L Cards

Table 22: 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.

Performance monitoring (PM) parameters are used by service providers to gather, store, set thresholds for, and report performance data for early detection of network issues. You can configure and retrieve PM counters for the OCH and OMS controllers in 30-second, 15-minute, 24-hour intervals or in 10-second flexible bin interval. These parameters simplify troubleshooting operations and enhance data that can be collected directly from the equipment.

PM Parameters Supported on OMS Controller

The PM parameters that are supported on OMS controller are given below.

Table 23: 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

The PM parameters that are supported on OCH controller are given below.

Controller	Supported PM Parameters	Description
OCH	OPT (dBm)	Transmitted Power
	OPR (dBm)	Received Power

Configuring PM Parameters for NCS1K14-CCMD-16-C and NCS1K14-CCMD-16-L Cards

You can configure and view the performance monitoring parameters for the OMS and OCH controllers.

To configure minimum and maximum threshold for individual parameters, use the following commands.

configure

controller *controllertype R/S/I/P* **pm** {**30-sec** | **15-min** | **24-hour**} **optics threshold** { **parameter-name**} {**max**|**min**} {**value**}

commit

To enable reporting of threshold crossing alarms for individual parameters, use the following commands.

configure

controller *controllertype R/S/I/P* **pm** {**30-sec** | **15-min** | **24-hour**} **optics report** { **parameter-name**} {**min-tca**|**max-tca**}

commit

Examples

The following is a sample with the performance monitoring parameters of 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
```

The following is a sample with the performance monitoring parameters of 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
```

Viewing PM Parameters

To view the performance monitoring parameters for OMS and OCH controllers, use this command:

show controllers *controllertype R/S/I/P* **pm { current | history } { 30 sec | 15-min | 24-hour | flex-bin } optics** { *linenumber* }

Examples for viewing PM parameters for OMS controller are given below:

Example 1

```
RP/0/RP0/CPU0:Tethys_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
```

Example 2

```
RP/0/RP0/CPU0:Tethys_P2A_DT_03#show controllers oms 0/1/0/0 pm current 15-min optics 1

Optics in the current interval [15:00:00 - 15:03:18 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     -8.23     -8.20    -50.00            NA              NO   30.00
          NA              NO
OPR[dBm]      : -1.80     -1.60     -1.30    -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]     : -3.40     -3.34     -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
```

Example 3

```
RP/0/RP0/CPU0:Tethys_P2A_DT_03#show controllers oms 0/1/0/0 pm current flex-bin optics 1

Optics in the current interval [15:03:40 - 15:03:44 Mon Nov 20 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)
OPT[dBm] : -8.30 -8.22 -8.20 0.00 NA NO 0.00 NA NO
OPR[dBm] : -1.50 -1.50 -1.50 0.00 NA NO 0.00 NA NO
OPBR[dBm] : -11.61 -11.61 -11.61 0.00 NA NO 0.00 NA NO
OPBRR[dB] : -3.40 -3.38 -3.30 0.00 NA NO 0.00 NA NO
EAGN[dB] : 2.00 2.00 2.00 0.00 NA NO 0.00 NA NO
EATL[dB] : 0.00 0.00 0.00 0.00 NA NO 0.00 NA NO
IAGN[dB] : 5.00 5.00 5.00 0.00 NA NO 0.00 NA NO
IATL[dB] : 0.00 0.00 0.00 0.00 NA NO 0.00 NA NO

Last clearing of "show controllers OPTICS" counters never
```

Example 4

```
RP/0/RP0/CPU0:Tethys_P2A_DT_03#show controllers oms 0/1/0/0 pm current 24-hour optics 1

Optics in the current interval [00:00:00 - 15:04:07 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     -8.27     -8.20    -50.00            NA              NO   30.00
          NA              NO
OPR[dBm]     : -3.00     -1.62     -0.20    -50.00            NA              NO   30.00
          NA              NO
OPBR[dBm]    : -11.61    -11.61    -11.51   -50.00             NA              NO   -10.00
           NA              NO
OPBRR[dB]    : -3.40     -3.31     -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      0.10     -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
```

Examples for viewing PM parameters for OCH controller are given below:

Example 1

```
RP/0/RP0/CPU0:Tethys_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
```

Example 2

```
RP/0/RP0/CPU0:Tethys_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
```

Example 3

```
RP/0/RP0/CPU0:Tethys_P2A_DT_03#show controllers och 0/1/0/2 pm current flex-bin optics 1

Optics in the current interval [15:05:20 - 15:05:28 Mon Nov 20 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)
OPT[dBm] : -1.40 -1.36 -1.30 0.00 NA NO 0.00 NA NO
OPR[dBm] : -5.80 -5.73 -5.70 0.00 NA NO 0.00 NA NO

Last clearing of "show controllers OPTICS" counters never
```

Example 4

```
RP/0/RP0/CPU0:Tethys_P2A_DT_03#show controllers och 0/1/0/2 pm current 24-hour optics 1

Optics in the current interval [00:00:00 - 15:06:11 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] : -3.00 -1.58 -0.10 -50.00 NA NO 30.00 NA NO
OPR[dBm] : -5.80 -5.76 -5.70 -50.00 NA NO 30.00 NA NO

Last clearing of "show controllers OPTICS" counters never
```

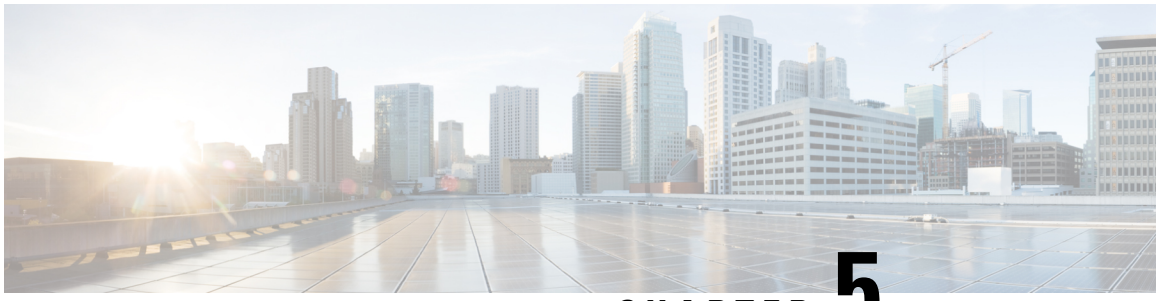
Viewing PM History Parameters

To view the performance monitoring parameters for OMS and OCH controllers, use this command:

show controllers *controllertype R/S/I/P* **pm history { 30 sec | 15-min | 24-hour } optics** { *linenumber* }

Example

```
RP/0/RP0/CPU0:Tethys_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 (PRBS) feature allows users to perform data integrity checks on their encapsulated packet data payloads using a pseudo-random bit stream pattern. PRBS generates a bit pattern and sends it to the peer router that uses this feature to detect if the sent bit pattern is intact or not.

- Pseudo Random Binary Sequence, on page 151

Pseudo Random Binary Sequence

Pseudo Random Binary Sequence (PRBS) feature allows users to perform data integrity checks on their encapsulated packet data payloads using a pseudo-random bit stream pattern. PRBS generates a bit pattern and sends it to the peer router that uses this feature to detect if the sent bit pattern is intact or not.

The Pseudo Random Binary Sequence (PRBS) feature allows you to perform data integrity checks between the NCS1014 trunk links and client links without enabling the actual client traffic.

PRBS supports:

- Trunk PRBS(coherentDSPCtrl)
- Client PRBS(HundredGigECtrlr and FourHundredGigECtrlr)

You need to enable PRBS feature on both the transmitting and receiving NCS 1014 trunk ports. The transmitting trunk port generates a bit pattern and sends it to the peer NCS 1014 device. The device detects if the sent bit pattern is received.

You can configure PRBS on the NCS 1014 trunk port and client port for the NCS1K4-2.4T-K9 card.

- **Source mode —** The NCS 1014 at trunk port generates PRBS signal on the line continuously as per the configured PRBS pattern.
- **Sink mode —** The NCS 1014 at trunk port gets locked to the ingress signal according to the configured pattern, analyzes and reports the errors.
- **Source-Sink mode —** The NCS 1014 at trunk port acts as both the PRBS transmitter and receiver, that is, it generates PRBS signal as per the configured pattern, and also gets locked to the ingress signal with the same pattern, and reports the errors.

Trunk PRBS

NCS 1014 trunk port supports the following 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
- **PRBS7** — Sequence length is from 2^7 -1 bits.

Note NCS1K4-2.4T-K9 Interoperability for ethernet PRBS PN23 pattern is not supported.

Configuring Trunk PRBS on NCS1K4-2.4T-K9

Use the following sample configuration to configure PRBS trunk mode on the NCS1K4-2.4T-K9:

```
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
```

Use the following sample configuration to display trunk controllers details:

```
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
```

Use the following sample configuration to display cumulative count of PRBS bit errors in the 15-min sampling interval:

```
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

NCS 1014 client port supports the following PRBS patterns:

- **PRBS31** — Sequence length is from 2^31 -1 bits
- **PRBS23** — Sequence length is from 2^23 -1 bits

Configuring Client PRBS on NCS1K4-2.4T-K9

Use the following sample configuration to configure PRBS client mode on the NCS1K4-2.4T-K9:

```
RP/0/RP0/CPU0:ios(config)#controller fourHundredGigECtrlr 0/0/0/4
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern pn23
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Use the following sample configuration to display four hundred gigabit client controllers details:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4
Wed Nov 15 18:39:29.478 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: PN23
      Direction: Line
      Framing: Framed
    Laser Squelch: Disabled
    Insert Idle Ingress: Disabled
    Insert Idle Egress: Disabled

Phy:
    Media type: Not known
    Statistics:
        FEC:
           Corrected Codeword Count: 2019127152              Valid: True       Start time:
 17:35:46 Wed Nov 15 2023
           Uncorrected Codeword Count: 6                     Valid: True       Start time:
 17:35:46 Wed Nov 15 2023
        PCS:
           Total BIP errors: 0                               Valid: True       Start time:
 17:35:46 Wed Nov 15 2023
           Total frame errors: 0                             Valid: False      Start time:
 17:35:46 Wed Nov 15 2023
           Total Bad SH: 0                                   Valid: False      Start time:
 17:35:46 Wed Nov 15 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
```

Use the following sample configuration to display four hundred gigabit client controller PRBS bit errors in the 15-min sampling interval:

```
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
```