Configuring the Card Mode

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

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

1.2T line cards

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

A 1.2T line card

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

Port details:

The line cards are equipped with trunk and client ports:

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

Configuration modes

You can configure the line cards in these two modes:

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

- **Muxponder slice mode**: the client-to-trunk mapping is fixed.

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

Table 1: Client-to-trunk mapping for muxponder slice mode

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

Supported data rates on the 1.2T line card

The 1.2T line card supports these data rates.

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

- Muxponder
- Muxponder slice 0
- Muxponder slice 1

Table 2: Data rates for muxponder and muxponder slice 0 and slice 1 mode configuration

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

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

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

Table 3: Trunk parameter range for the 1.2T line card

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

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

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

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

Table 4: Supported data rates for muxponder mode

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

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

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

Table 5: Supported data rates for split client port mapping mode

Trunk data rate (per trunk)	Total configured data rate	Trunk ports	Client ports for trunk 0 (100G)	Shared client port (50G per trunk)	Client ports for trunk 1 (100G)
50G	100G	0, 1	-	7	-
150G	300G	0, 1	2	7	8
250G	500G	0, 1	2, 3	7	8, 9
350G	700G	0, 1	2, 3, 4	7	8, 9, 10
450G	900G	0, 1	2, 3, 4, 5	7	8, 9, 10, 11
550G	1.1T	0, 1	2, 3, 4, 5, 6	7	8, 9, 10, 11, 12

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

Coupled mode restrictions

These restrictions apply to the coupled mode configuration:

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

Configure split client port mapping

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

Procedure

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

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

Example:

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

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

  Example:

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

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

Example:

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

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

Example:

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

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

The split client port mapping is configured.

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

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

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

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

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

Configure the card mode

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

Procedure

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

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

Example:

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

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

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

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

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

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

Example:

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

```
RP/0/RP0/CPU0:ios#config
Tue Oct 15 01:24:56.355 UTC
```

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder trunk-rate 550G
RP/0/RP0/CPU0:ios(config)#commit
```

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

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

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

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

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

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

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

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

Verify the 1.2T line card configuration

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

Procedure

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

Example:

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

Location:             0/2/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
```

```
Client Port            Mapper/Trunk Port   CoherentDSP0/2/0/0  CoherentDSP0/2/0/1
                     Traffic Split Percentage

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Wed Mar 11 13:59:11.073 UTC

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

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

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

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

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


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

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

```
HundredGigECtrlr0/1/0/11        ODU40/1/0/1/4                         100
HundredGigECtrlr0/1/0/12        ODU40/1/0/1/5                         100
```

Clear alarm statistics

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

Procedure

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

Example:

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

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

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

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

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

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

FEC mode                                        : Soft-Decision 15

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

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

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

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

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

FEC mode                                          : Soft-Decision 15

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

Regeneration mode

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

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

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

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

Configure the card in Regen mode

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

Procedure

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

Example:

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

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

Step 2 Commit and exit the configuration.

Example:

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

Verify the Regen mode

Use this task to verify the regeneration mode configuration.

Procedure

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

Example:

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

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

Configure the BPS

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

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

Supported baud rates are shown in the table.

Table 6: Supported baud rates

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

Use this task to configure BPS value to 3.4375 on the 1.2T and 2-QDD-C line cards to support 300G trunk configurations on 75 GHz networks.

Procedure

Step 1 Run the **controller optics** *R/S/I/P* **bits-per-symbol value** command to configure the BPS to 3.4375 on the 1.2T and 2-QDD-C line cards. This configuration supports 300G trunk configurations on 75 GHz networks.

Example:

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

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

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

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

Example:

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

Example:

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

```
300G            SD27            4.8359375       43.0744397
300G            SD27            4.8437500       43.0049648
```

Trunk rates and BPSK modulation

Trunk rates on the 1.2T and 2-QDD-C line cards can be configured to 50G, 100G, and 150G to support Binary Phase-Shift Keying (BPSK) modulation.

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

Configuration methods

You can configure trunk rates for BPSK modulation using these methods:

- Command-Line Interface (CLI)
- NETCONF YANG
- OC Models

Supported trunk rates and BPSK modulation

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

Table 7: Trunk rates with the supported BPSK modulation

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

Configure trunk rate for BPSK modulation

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

Procedure

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

Example:

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

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

```
RP/0/RP0/CPU0:(config-hwmod-mxp)#trunk-rate 50G
RP/0/RP0/CPU0:(config-hwmod-mxp)#commit
```

View BPSK trunk rate ranges

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

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

Procedure

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

Example:

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

Tue Feb 25 11:13:41.934 UTC

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

                                Traffic Split Percentage


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


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

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


         LOS/LOL/Fault Status:

         Alarm Statistics:

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

         Performance Monitoring: Enable

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

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

         Configured Tx Power = 2.00 dBm
         Configured CD High Threshold = 180000 ps/nm
         Configured CD lower Threshold = -180000 ps/nm
         Configured OSNR lower Threshold = 0.00 dB
         Configured DGD Higher Threshold = 180.00 ps
         Baud Rate =  34.7175521851 GBd
         Bits per Symbol = 1.0000000000  bits/symbol
         Modulation Type: BPSK
         Chromatic Dispersion -9 ps/nm
         Configured CD-MIN -180000 ps/nm  CD-MAX 180000 ps/nm
         Polarization Mode Dispersion = 0.0 ps
         Second Order Polarization Mode Dispersion = 125.00 ps^2
         Optical Signal to Noise Ratio = 34.60 dB
         SNR = 20.30 dB
         Polarization Dependent Loss = 0.20 dB
         Polarization Change Rate = 0.00 rad/s
         Differential Group Delay = 2.00 ps
         Filter Roll Off Factor : 0.100
         Rx VOA Fixed Ratio : 15.00 dB
         Enhanced Colorless Mode : 0
         Enhanced SOP Tolerance Mode : 0
         NLEQ Compensation Mode : 0
         Cross Polarization Gain Mode : 0
         Cross Polarization Weight Mode : 0
         Carrier Phase Recovery Window : 0
         Carrier Phase Recovery Extended Window : 0
AINS Soak                : None
AINS Timer               : 0h, 0m
AINS remaining time      : 0 seconds
```

QXP card

Table 8: 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 9: 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.
- QDD-400G-ZR-S pluggable module supports FEC mode CFEC only.
- QDD-400G-ZR-S pluggable module operates only as an Ethernet transponder.

Supported data rates for QXP card

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

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

DAC supported modes for NCS1K4-QXP-K9 card

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

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

Table 10: 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

Trunk rate	Modulation format	Default value	Modified DAC supported
400G	16-QAM	1x1	1x1.50

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

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

Note

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

Configure 400G transponder mode

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

Procedure

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

Example:

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

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

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

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

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

Note

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

Configure 400G muxponder mode

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

Procedure

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

Example:

The is a sample to configure a 400G MXP.

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

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

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/3 mxponder-slice 1
Sat Jun 25 23:03:20.823 UTC
Location: 0/3
Slice ID: 1
Client Bitrate: 100GE
Trunk Bitrate: 400G
Status: Provisioned
LLDP Drop Enabled: FALSE
ARP Snoop Enabled: FALSE
Client Port Mapper/Trunk Port CoherentDSP0/3/0/2
Traffic Split Percentage
HundredGigECtrlr0/3/0/3/1 - 100
HundredGigECtrlr0/3/0/3/2 - 100
HundredGigECtrlr0/3/0/3/3 - 100
HundredGigECtrlr0/3/0/3/4 - 100
```

Configure 2x100G muxponder mode

Use this task to configure and provision 2x100G MXP.

Procedure

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

Example:

This is a sample to configure 2x100G MXP.

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

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

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

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

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

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

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

- DP04QSDD-HK9 operates as Ethernet or OTN transponder.
- DP04QSDD-HE0 operates only as an Ethernet transponder.

Table 11: Operating modes supported for Bright ZR+ pluggable modules on QXP card

Operating mode	Modulation	FEC
4x100GE MXP	16-QAM	CFEC
4x100GE MXP	16-QAM	OFEC

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

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

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

Procedure

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

Example:

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

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

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

Example:

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

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

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

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

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

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

         Performance Monitoring: Enable

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

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

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

Note

DP04QSDD-HK9 operates as Ethernet or OTN transponder. DP04QSDD-HE0 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.

Configure the GCC0 interface on a QXP card

Use this task to enable and configure the GCC0 interface on a Coherent DSP controller in a QXP card. Assign an IPv4 address to the interface to facilitate configuration. The GCC0 interface operates at a data rate of 7.7 Mbps on the QXP card.

Table 12: Feature History

Feature Name	Release Information	Feature Description
GCC0 interface support on NCS1K4-QXP-K9 card	Cisco IOS XR Release 25.4.1	This feature introduces GCC0 interface support in Trunk OpenROADM mode for the DP04QSDD-HK9 pluggable on the NCS1K4-QXP-K9 card. The Coherent DSP controller supports the GCC0 interface, enabling you to remotely manage, monitor, and operate the chassis and line cards, especially in environments without direct Data Communication Network (DCN) access.

Follow these steps to configure the GCC0 interface on a QXP card.

Procedure

Step 1 Enter configuration mode for the Coherent DSP controller and enable the GCC0 interface.

Example:

```
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)#gcc0
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

Step 2 Enter the **ipv4 address** *ipv4-address net-mask* command to assign the IPv4 address and subset mask to the GCC0 interface.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface gcc0 0/0/0/0
RP/0/RP0/CPU0:ios(config-CoDSP)#ipv4 address 192.0.2.1 255.255.255.0
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

Step 3 (Optional) Enter the **interface Loopback** *R/S/I/P* **ipv4 address** *ipv4-address* command to configure the interface loopback.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface Loopback0
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 20.1.1.1 255.255.255.255
```

Step 4 (Optional) Enter the **ipv4 unnumbered loopback 0** command to configure the GCC0 interface using the loopback IP address.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface GCC0 0/1/0/0
```

```
RP/0/RP0/CPU0:ios(config-if)#ipv4 unnumbered loopback 0
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#exit
```

The GCC0 interface is enabled with the specified IPv4 address and configured on the Coherent DSP controller of the QXP card.

Verify the GCC0 interface status and IPv4 configuration

Use this task to confirm the operational status and assigned IPv4 address of GCC0 interfaces.

Procedure

Run the **show ipv4 interface brief** command to display a summary of IPv4 interfaces.

Example:

```
RP/0/RP0/CPU0:ios#show ipv4 interface brief
Tue Sep 16 00:40:52.056 UTC
Interface                 IP-Address      Status          Protocol        Vrf-Name
GCC00/0/0/0               198.51.100.51     Up              Up              default
MgmtEth0/RP0/CPU0/0       192.0.2.32    Up              Up              default
MgmtEth0/RP0/CPU0/1       unassigned      Shutdown        Down            default
MgmtEth0/RP0/CPU0/2       unassigned      Shutdown        Down            default
```

The output displays the IPv4 address, status, and protocol for GCC0 interfaces, confirming their configuration.

Configure the MTU to prevent IP fragmentation on GCC0 for SCP

Use this task to prevent IP fragmentation on GCC0 interfaces during SCP protocol operations by limiting the maximum transmission unit (MTU).

IP fragmentation is not supported on GCC0 interfaces for the SCP protocol. To avoid fragmentation, configure the interface to restrict the maximum packet size to less than 1454 bytes, which is the fragmentation limit.

Procedure

Step 1 Enter the configuration mode and enable the GCC0 interface.

Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface gcc0 0/0/0/0
```

Step 2 Enter the command **ipv4 mtu** *size* to set the IPv4 MTU size for the GCC0 interface.

The MTU size must be 1200 bytes.

Example:

```
RP/0/RP0/CPU0:ios(config-if)#ipv4 mtu 1200
RP/0/RP0/CPU0:ios(config-if)#commit
```

The IPv4 MTU is now configured on the GCC0 interface. This setting limits the maximum packet size and prevents SCP-related fragmentation.

ONS-QDD-OLS pluggable

Table 13: Feature History

Feature Name	Release Information	Description
Pluggable support	Cisco IOS XR Release 25.2.1	The NCS1K4-QXP-K9 line card now supports the new ONS-QDD-OLS optical amplifier pluggable. It is supported independently on all 16 ports of the QXP card and offers various channel breakout options to combine or separate each channel from a coherent DWDM optical source using these breakout cables: • ONS-BRK-CS-8LC • ONS-BRK-CS-16LC • ONS-CAB-CS-LC-5 This pluggable increases fiber bandwidth and lowers power dissipation. CLI: These keywords are added to the **hw-module location** command: • **ols-port <port number>** • **mode edfa**

ONS QDD optical line systems

The ONS-QDD-OLS is a pluggable optical amplifier that interconnects two routers or switches for transporting a limited number of coherent optical channels over a single span point-to-point link.

ONS-QDD-OLS features and support

These are the key features of the ONS-QDD-OLS pluggable optical amplifier:

- OLS Optics is supported independently on all 16 ports of NCS1K4-QXP-K9 line card. The EDFA `ols-port` mode is supported on ports 0 through 15 of the ONS-QDD-OLS pluggable.
- New XR CLI commands are introduced for OLS configuration:
  - `OLS-PORT` is used to select a specific port, extending the `hwmode` configuration.
  - `OLS-MODE` is used under the `hw-module` configuration specifically for EDFA settings.
- When a port is configured as an `OLS-PORT`, the corresponding TXP/MXP slice becomes unavailable for provisioning.
  - COM is represented as `OTS R/S/I/P/0`.
  - LINE is represented as `OTS R/S/I/P/1`.
- On the OTS controller, only egress parameters configuration is supported; ingress parameters are not supported.

The OLS configurations also utilize these additional breakout cable- assembly and patch-cord to establish connections between the EDFA module and the QDD-ZR/ZRP optical channels:

- ONS-BRK-CS-8LC: A dual-fanout 1x8 cable-assembly with embedded passive splitter and coupler.
- ONS-BRK-CS-16LC: A dual-fanout 1x16 cable-assembly with embedded passive splitter and coupler.
- ONS-CAB-CS-LC-5: A 5-meter dual adapter patch-cord with CS-connectors on one end and LC-connectors on the other.

Supported wavelength or frequency configuration

For each channel supported through ONS-BRK-CS-8LC or ONS-BRK-CS-16LC passive/mux cable, the wavelength or the frequency must be configured according to this table:

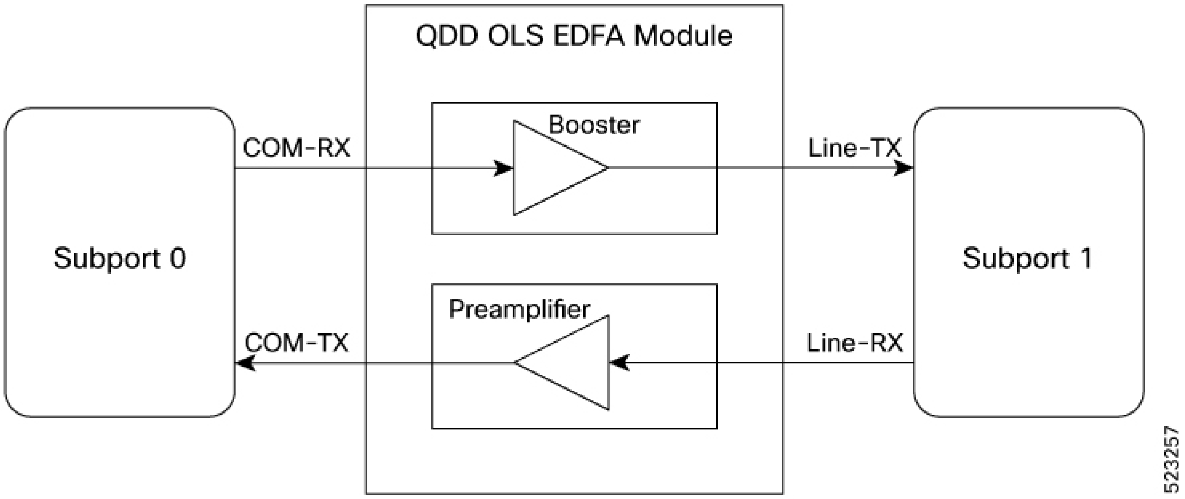
Table 14: ONS-QDD-OLS operating signal wavelength range

Channel spacing	Total bandwidth	Wavelength		Frequency	
		Start	End	Start	End
8 channels - 200 GHz spaced 16 channels - 100 GHz spaced	19.2 nm 2.4 THz	1539.1 nm	1558.4 nm	192.375 THz	194.775 THz

Functional description of QDD-OLS

The QDD OLS pluggable contains the COM side and the Line side as shown in this figure:

Figure 1: Functional description of QDD OLS



Each physical port of the QDD OLS pluggable is represented as two ots controllers (subport 0 and subport 1). COM port is subport 0 and Line port is subport 1.

The Gain of the Booster is associated to subport 1 while the gain of the Preamplifier is associated to subport 0.

Table 15: OTS and optical ports

Controller	Optical ports
ots R/S/I/P/0	COM-RX (booster input)
	COM-TX (preamplifier output)
ots R/S/I/P/1	LINE-RX (preamplifier input)
	LINE-TX (booster output)

Configure the ols-port in EDFA mode

Use this task to configure the ONS-QDD-OLS pluggable ols-port in EDFA mode.

Procedure

Step 1 Run the **hw-module location** command to configure the pluggable on specific ols-port.

Example:

This is a sample to configure the pluggable on slot 2 and port 14.

```
RP/0/RP0/CPU0:ios#conf
Fri Feb 28 22:36:59.927 IST
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 ols-port 14
```

Step 2 Configure the ols-port in the EDFA mode.

Example:

```
RP/0/RP0/CPU0:ios(config-ols)#mode edfa
```

Step 3 Run the **commit** and **end** commands to commit the changes and exit the configuration mode.

Example:

```
RP/0/RP0/CPU0:ios(config-ols)#commit
Fri Feb 28 22:37:26.891 IST
RP/0/RP0/CPU0:ios(config-ols)#end
RP/0/RP0/CPU0:ios#
```

Step 4 Verify the configuration using the **show hw-module location**_location_**ols-port** command in EDFA mode.

Example:

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0
         ols-port 14
         mode edfa
```

OTS parameters and operational data sample configurations

This table lists configuration examples for ONS-QDD-OLS pluggable OTS parameters:

Table 16: OTS parameters

Parameters	Configuration example
Gain setting in COM port	RP/0/RP0/CPU0:ios#configur Fri Feb 28 23:06:25.489 IST RP/0/RP0/CPU0:ios(config)#controller ots 0/2/0/14/0 RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-gain 200 RP/0/RP0/CPU0:ios(config-Ots)#commit Fri Feb 28 23:06:48.834 IST RP/0/RP0/CPU0:ios(config-Ots)#end RP/0/RP0/CPU0:ios# RP/0/RP0/CPU0:ios#
Operational mode	RP/0/RP0/CPU0:ios#configur Mon Feb 3 19:20:02.757 UTC RP/0/RP0/CPU0:ios(config)#controller ots 0/0/0/1/0 RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-mode ? power-control Set amplifier to power control mode RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-mode power-control RP/0/RP0/CPU0:ios(config-Ots)#commit Mon Feb 3 19:20:13.832 UTC

Parameters	Configuration example

Gain setting in Line port	`RP/0/RP0/CPU0:ios#configur`
	`Fri Feb 28 23:08:08.172 IST`
	`RP/0/RP0/CPU0:ios(config)#`
	`RP/0/RP0/CPU0:ios(config)#controller ots 0/2/0/14/1`
	`RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-gain 210`
	`RP/0/RP0/CPU0:ios(config-Ots)#commit`
	`Fri Feb 28 23:08:20.677 IST`
	`RP/0/RP0/CPU0:ios(config-Ots)#`

Power	`RP/0/RP0/CPU0:ios#configur`
	`Mon Feb  3 19:22:36.395 UTC`
	`RP/0/RP0/CPU0:ios(config)#controller ots 0/0/0/1/0`
	`RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-power 110`
	`RP/0/RP0/CPU0:ios(config-Ots)#commit`
	`Mon Feb  3 19:22:45.173 UTC`

Egress ampli OSRI mode	`RP/0/RP0/CPU0:ios(config)#controller ots 0/2/0/14/0`
	`RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-osri`
	`RP/0/RP0/CPU0:ios(config-Ots)#commit`
	`Fri Feb 28 23:13:07.065 IST`
	`RP/0/RP0/CPU0:ios(config-Ots)#`

Delete configuration for egress ampli OSRI mode	`RP/0/RP0/CPU0:ios(config)#controller ots 0/2/0/14/0`
	`RP/0/RP0/CPU0:ios(config-Ots)#no egress-ampli-osri`
	`RP/0/RP0/CPU0:ios(config-Ots)#commit`
	`Fri Feb 28 23:14:05.117 IST`
	`RP/0/RP0/CPU0:ios(config-Ots)#`

ALS on line	`RP/0/RP0/CPU0:ios#configur`
	`Mon Feb  3 19:11:03.983 UTC`
	`RP/0/RP0/CPU0:ios(config)#controller ots 0/1/0/1/1`
	`RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-safety-control-mode ?`
	`  auto      Select Safety Control Mode: Automatic`
	`  disabled  Disable Safety Control Mode`
	`RP/0/RP0/CPU0:ios(config-Ots)#egress-ampli-safety-control-mode disabled`
	`RP/0/RP0/CPU0:ios(config-Ots)#commit`
	`Mon Feb  3 19:11:30.980 UTC`

Parameters	Configuration example

TX low threshold	`RP/0/RP0/CPU0:ios#configur`
	`Mon Feb  3 18:38:42.101 UTC`
	`RP/0/RP0/CPU0:ios(config)#controller ots 0/0/0/1/0`
	`RP/0/RP0/CPU0:ios(config-Ots)#tx-low-threshold 160`
	`RP/0/RP0/CPU0:ios(config-Ots)#commit`
	`Mon Feb  3 18:39:09.280 UTC`

RX low threshold	`RP/0/RP0/CPU0:ios#configur`
	`Mon Feb  3 18:42:06.049 UTC`
	`RP/0/RP0/CPU0:ios(config)#controller ots 0/0/0/1/1`
	`RP/0/RP0/CPU0:ios(config-Ots)#rx-low-threshold -40`
	`RP/0/RP0/CPU0:ios(config-Ots)#commit`
	`Mon Feb  3 18:42:27.695 UTC`

Operational data on COM port, line port, and optics

This table lists configurations examples and unsupported parameters on the ONS-QDD-OLS pluggable:

Table 17: Operational data for COM port, line port, and optics

Operational data	Configuration example	Unsupported parameters

COM port (OTS 0)	RP/0/RP0/CPU0:ios#show controllers ots 0/2/0/14/0
	Fri Feb 28 22:44:42.823 IST
	 Controller State: Up
	 Transport Admin State: In Service
	 LED State: Green
	 Last link flapped: 00:38:04
	         Alarm Status:
	         ------------- 	
	         Detected Alarms: None
	         Alarm Statistics:
	         ----------------- 	
	         RX-LOS-P = 0
	         RX-LOC = 0
	         TX-POWER-FAIL-LOW = 0
	         INGRESS-AUTO-LASER-SHUT = 0
	
	         INGRESS-AUTO-POW-RED = 0
	         INGRESS-AMPLI-GAIN-LOW = 0
	
	         INGRESS-AMPLI-GAIN-HIGH = 0
	
	         EGRESS-AUTO-LASER-SHUT = 0
	
	         EGRESS-AUTO-POW-RED = 0
	         EGRESS-AMPLI-GAIN-LOW = 0
	         EGRESS-AMPLI-GAIN-HIGH = 0
	
	         HIGH-TX-BR-PWR = 0
	         HIGH-RX-BR-PWR = 0
	         SPAN-TOO-SHORT-TX = 0
	         SPAN-TOO-SHORT-RX = 0
	         INGRESS-AMPLI-LASER-OFF = 0
	
	         EGRESS-AMPLI-LASER-OFF = 0
	
	         Parameter Statistics:
	         --------------------- 	
	         Total Rx Power = -9.18 dBm
	         Total Tx Power = 14.36 dBm
	         Egress Ampli Mode = Gain
	         Egress Ampli Gain = 19.0 dB
	         Egress Ampli OSRI = OFF
	         Egress Ampli Force APR = OFF
	
	         Configured Parameters:
	         ------------- 	
	         Egress Ampli Mode = Gain
	         Egress Ampli Gain = 19.0 dB
	         Egress Ampli Power = 8.0 dBm
	         Egress Ampli OSRI = OFF
	         Rx Low Threshold = -30.0 dBm
	         Tx Low Threshold = -5.0 dBm
	
	RP/0/RP0/CPU0:ios#
	RP/0/RP0/CPU0:ios#
		• INGRESS Parameters(alarms statistics)
		• HIGH-TX/RX-BR-POWER
		• SPAN-TOO-SHORT-TX/RX
		• Egress Ampli Force APR

<table>
<tr><th>Operational data</th><th>Configuration example</th><th>Unsupported parameters</th></tr>
<tr><td>Line port (OTS 1)</td><td>

```
RP/0/RP0/CPU0:ios#sh controllers ots
0/2/0/14/1
Fri Feb 28 22:54:15.156 IST
 Controller State: Up
 Transport Admin State: In Service
 LED State: Green
 Last link flapped: 00:47:36
         Alarm Status:
         -------------
         Detected Alarms: None
         Alarm Statistics:
         -----------------
         RX-LOS-P = 0
         RX-LOC = 0
         TX-POWER-FAIL-LOW = 0
         INGRESS-AUTO-LASER-SHUT = 0

         INGRESS-AUTO-POW-RED = 0
         INGRESS-AMPLI-GAIN-LOW = 0

         INGRESS-AMPLI-GAIN-HIGH = 0

         EGRESS-AUTO-LASER-SHUT = 0

         EGRESS-AUTO-POW-RED = 0
         EGRESS-AMPLI-GAIN-LOW = 0
         EGRESS-AMPLI-GAIN-HIGH = 0

         HIGH-TX-BR-PWR = 0
         HIGH-RX-BR-PWR = 0
         SPAN-TOO-SHORT-TX = 0
         SPAN-TOO-SHORT-RX = 0
         INGRESS-AMPLI-LASER-OFF = 0

         EGRESS-AMPLI-LASER-OFF = 0
  Parameter Statistics:
         ---------------------
         Total Rx Power = -5.67 dBm
         Total Tx Power = 10.80 dBm
         Egress Ampli Mode = Gain
         Egress Ampli Gain = 21.0 dB
         Egress Ampli Safety Control mode =
disabled
         Egress Ampli OSRI = OFF
         Egress Ampli Force APR = OFF

         Configured Parameters:
         -------------
         Egress Ampli Mode = Gain
         Egress Ampli Gain = 21.0 dB
         Egress Ampli Power = 8.0 dBm
         Egress Ampli Safety Control mode =
auto
         Egress Ampli OSRI = OFF
         Rx Low Threshold = -30.0 dBm
         Tx Low Threshold = -5.0 dBm
```

</td><td>

- INGRESS Parameters(alarms statistics)
- HIGH-TX/RX-BR-POWER
- SPAN-TOO-SHORT-TX/RX
- Egress Ampli Force APR

</td></tr>
</table>

Operational data	Configuration example	Unsupported parameters

Optics	RP/0/RP0/CPU0:Node68#sh controllers ots
	Ots Ots-Och
	RP/0/RP0/CPU0:Node68#sh controllers optics 0/3/0/2
	 Controller State: Administratively Down
	 Transport Admin State: Out Of Service
	 Laser State: Off
	 LED State: Off Optics Status
	         Optics Type: QSFP-DD DUAL EDFA
	Transceiver Vendor Details
	
	         Form Factor : QSFP-DD
	         Name :
	CISCO-ACCELINK
	        Part Number : 10-100458-01
	
	         Rev Number : 27
	         Serial Number : ACW2739Z00M
	
	         PID : ONS-QDD-OLS
	
	         VID : V01
	         Firmware Version :
	Major.Minor.Build
	         Active : 2.07.
	         Inactive : 2.05.
	         Date Code(yy/mm/dd) : 23/10/04
	         Fiber Connector Type: CS
	         Otn Application Code: Not Set
	         Sonet Application Code: Not Set
	         Ethernet Compliance Code: Not set
		—

DP04QSDD-E26-A1 pluggable

A DP04QSDD-E26-A1 pluggable is a transceiver module that:

- Operates as a variant of the ZR pluggable family.
- Is supported on the QXP line card.
- Can be hosted by a DWDM line interface to support TXP or MXP datapaths.

Limitations for DP04QSDD-E26-A1 pluggable

- Supports only CFEC mode for FEC.
- Datapath support is limited to 400GE TXP and 4x100GE MXP.
- Supports only the ZRP trunk framing format.

2-QDD-C line card

Table 18: Feature History

Feature	Release Information	Description
NCS1K4-2-QDD-C-K9 C-Band Line Card	Cisco IOS XR Release 25.2.1	NCS 1014 now supports the NCS1K4-2-QDD-C-K9 C-Band line card. This card features eight client ports (QSFP28 and QSFP-DD) and two software-configurable DWDM dual sub-channel module trunk ports. Each trunk port supports line rates of 200, 300, and 400 Gbps with precise control over modulation format, baud rate, and forward error correction. Additionally, the line card supports both module and slice configurations, enhancing network flexibility and performance.

This section describes the supported configurations and procedures to configure the card modes on the 2-QDD-C line card.

Limitations for 2-QDD-C

- Flex Ethernet is not supported.
- A single 400GE cannot be split and use as 4x 100GE due to hardware limitations.

2-QDD-C card modes

A 2-QDD-C card mode is an operational configuration for 2-QDD-C line cards that support module and slice configurations.

The line cards have two trunk ports (0 and 1) and 8 client ports (2 through 9) each. You can configure the line card in two modes:

- Muxponder mode: Both trunk ports are configured with the same trunk rate. The client-to-trunk mapping is in a sequence in vertical order.
- Muxponder slice mode: Each trunk port is configured independently of the other with different trunk rates. The client-to-trunk mapping is fixed in vertical order. For Trunk 0, the client ports are 2 through 5. For Trunk 1, the client ports are 6 through 9.

Sub 50G configuration

You can configure sub 50G muxponder mode in these combination of trunk and client rates:

- 100GE Muxponder mode:
    - 1x100GE and 2x50G
    - 3x100GE and 2x150G

  - 5x100GE and 2x250G
  - 7x100GE and 2x350G

- OTU4 Muxponder mode:
  - 1xOTU4 and 2x50G
  - 3xOTU4 and 2x150G
  - 5xOTU4 and 2x250G
  - 7xOTU4 and 2x350G

This table displays the port configuration for the supported data rates.

Trunk data rate (per trunk)	Total configured data rate	Trunk ports	Client ports for trunk 0 (100G)	Shared client port (50G per trunk)	Client ports for trunk 1 (100G)
50G	100G	0, 1	-	2	-
150G	300G	0, 1	2	3	4
250G	500G	0, 1	2, 3	4	5, 6
350G	700G	0, 1	2, 3, 4	5	6, 7, 8

From Release 7.5.2, 2-QDD-C cards support 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	Trunk ports	Client ports for trunk 0 (100G)	Shared client port (50G per trunk)	Client ports for trunk 1 (100G)
50G	100G	0, 1	-	5	-
150G	300G	0, 1	2	5	6
250G	500G	0, 1	2, 3	5	6, 7
350G	700G	0, 1	2, 3, 4	5	6, 7, 8

For information on how to configure split client port mapping, see Configure Split Client Port Mapping .

Coupled mode restrictions

These restrictions apply to the coupled mode configuration:

- Both trunk ports must be configured with the same bits-per-symbol or baud rate 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.

Supported data rates for 2-QDD-C card

These tables display the supported data rates for the 2-QDD-C card.

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

Trunk data rate	Card support	Client data rate	Client optics	Trunk ports	Client ports
200	2-QDD-C	100GE, OTU4	QSFP-28	0, 1	2, 3, 4, 5
300	2-QDD-C	100GE, OTU4	QSFP-28	0, 1	2, 3, 4, 5, 6, 7
400	2-QDD-C	100GE, OTU4	QSFP-28	0, 1	2, 3, 4, 5, 6, 7, 8, 9
200	2-QDD-C	400GE	QSFP-DD	0, 1	4
400	2-QDD-C	400GE	QSFP-DD	0, 1	4,8

This 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
200	2-QDD-C	100GE, OTU4	0	2, 3
300	2-QDD-C	100GE, OTU4	0	2, 3, 4
400	2-QDD-C	100GE, OTU4	0	2, 3, 4, 5
400	2-QDD-C	400GE	0	4

This 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
200	2-QDD-C	100GE, OTU4	1	6, 7
300	2-QDD-C	100GE, OTU4	1	6, 7, 8
400	2-QDD-C	100GE, OTU4	1	6, 7, 8, 9
400	2-QDD-C	400GE	1	8

This table displays the trunk parameter ranges for the 2-QDD-C card.

Trunk payload	FEC	Min BPS	Max BPS	Min GBd	Max GBd
150G	27%	1.453125	4.335938	24.02079	71.67494

Trunk payload	FEC	Min BPS	Max BPS	Min GBd	Max GBd
200G	27%	2	4.40625	31.51	69.43
250G	27%	2.414063	6	28.93129	71.9069
300G	27%	2.8984375	6	34.7175497	71.8681352
350G	27%	3.382813	6	40.5038	71.84047
400G	27%	3.8671875	6	46.2900663	71.8197392
150G	15%	1.320313	3.9375	24.02079	71.67494
200G	15%	1.7578125	5.25	24.02079115	71.74209625
250G	15%	2.195313	6	26.27274	71.80592
300G	15%	3.8203125	6	31.52728839	49.51525048
350G	15%	3.070313	6	36.78184	71.87901
400G	15%	3.8671875	6	42.03638452	71.9018782

Note The recommended value for 6 BPS for corresponding line rates are listed below:

Trunk payload	FEC	BPS	GBd
300G	27%	6	34.7175
350G	27%	6	40.5038
400G	15%	6	42.0364

Configure the card mode for 2-QDD-C card

Use this task to configure a 2-QDD-C line card in either muxponder mode or muxponder slice mode. This configuration defines the client rate and the trunk rate for the line card.

Procedure

Step 1 Run these commands to configure the 2-QDD-C card in muxponder mode. Set the client rate and the trunk rate as specified.

- **hw-module location** *location* **mxponder client-rate {100GE | OTU4 }**
- **hw-module location** *location* **mxponder trunk-rate** {**100G | 150G | 200G | 250G | 300G | 350G | 400G** }
- **hw-module location** *location* **mxponder client-rate { 400GE}**

- **hw-module location** *location* **mxponder trunk-rate { 200G | 400G** }

Example:

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

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

This is a sample in which the card is configured in the muxponder mode with a 400GE trunk rate.

```
RP/0/RP0/CPU0:west#configure
Thu Oct  7 11:43:01.914 IST
RP/0/RP0/CPU0:west(config)#hw-module location 0/2 mxponder trunk-rate 4
400G  450G
RP/0/RP0/CPU0:west(config)#hw-module location 0/2 mxponder trunk-rate 400G
RP/0/RP0/CPU0:west(config)#hw-module location 0/2 mxponder client-rate 400GE
RP/0/RP0/CPU0:west(config)#commit
```

Step 2 Run these commands to configure the 2-QDD-C card in muxponder slice mode. Set the slice number, client rate, and trunk rate as required.

- **hw-module location** *location* **mxponder-slice** *mxponder-slice-number* **client-rate { 100GE | 400GE}**

- **hw-module location** *location* **mxponder-slice** *mxponder-slice-number* **trunk-rate** { **100G | 200G | 300G | 400G**}

Example:

This is a sample in which the card is configured in the muxponder slice 0 mode with a 400G trunk rate.

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

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

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

The 2-QDD-C card is configured in the specified muxponder or muxponder slice mode with the defined client and trunk rates.

Configure mixed client traffic mode

Use this task to configure the client traffic mode on each trunk in a line card independently. This provides flexibility for the same card to carry both OTN and Ethernet client traffic at the same time across two slices.

100G, 200G, and 300G trunk rates are supported on both slices (slice 0 and slice 1) with different client modes, including 100GE and OTU4.

From Release 7.10.1, you can configure both Ethernet and OTU interfaces on different client ports for each trunk in the 2-QDD-C line card. This enhancement allows the same line card to carry both OTN and Ethernet client traffic at the same time in a single slice for each trunk rate.

An additional 400G trunk rate is supported on both slices (slice 0 and slice 1) with different client modes, such as 100GE and OTU4.

Procedure

Step 1 Run these commands to configure the card for mixed client traffic mode with different slices (slice 0 and slice 1).

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

Example:

This is a sample in which the card is configured with mixed client rates in the muxponder slice 0 and 1 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 mxponder-slice 0 client-rate OTU4 trunk-rate 400G
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1 mxponder-slice 1 client-rate 100GE trunk-rate 400G
RP/0/RP0/CPU0:ios(config)#commit
```

This is a sample configuration of the mixed client traffic mode in different slices.

```
hw-module location 0/0
 mxponder-slice 0
  trunk-rate 400G
  client-rate OTU4
 !
 mxponder-slice 1
  trunk-rate 400G
  client-rate 100GE
 !
!
```

This is a sample configuration in which both the slices use the same client mode.

```
hw-module location 0/3
 mxponder
  trunk-rate 350G
  client-rate 100GE
 !
!
```

Step 2 Run the **show hw-module** *location* **mxponder** command to verify the card configuration.

Example:

This is a sample to verify the mixed client traffic mode in different slices.

```
Location:             0/0
Slice ID:             0
Client Bitrate:       OTU4
Trunk  Bitrate:       400G
Status:               Provisioned
Client Port                     Peer/Trunk Port            CoherentDSP0/0/0/0
                                Traffic Split Percentage
```

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


Location:             0/0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       400G
Status:               Provisioned
Client Port                     Peer/Trunk Port            CoherentDSP0/0/0/1
                                Traffic Split Percentage
HundredGigECtrlr0/0/0/6         ODU40/0/0/1/1                         100
HundredGigECtrlr0/0/0/7         ODU40/0/0/1/2                         100
HundredGigECtrlr0/0/0/8         ODU40/0/0/1/3                         100
HundredGigECtrlr0/0/0/9         ODU40/0/0/1/4                         100
```

This is a sample to verify both the slices using the same client mode.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/3 mxponder
Fri Nov 26 12:21:16.174 UTC

Location:             0/3
Client Bitrate:       100GE
Trunk  Bitrate:       350G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/3/0/0   CoherentDSP0/3/0/1

                                Traffic Split Percentage

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

Step 3 Run this command to configure the card for mixed client traffic mode in the same slice.

- **hw-module location** *location* **mxponder-slice** *mxponder-slice-number* **trunk-rate** {**100G** | **200G** | **300G** | **400G**|}

Example:

This is a sample in which the card is configured with mixed client rates in the muxponder slice 0 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 mxponder-slice 0 client-port-rate 2 client-type OTU4
 trunk-rate 400G
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1 mxponder-slice 0 client-port-rate 3 client-type
100GE trunk-rate 400G
RP/0/RP0/CPU0:ios(config)#commit
```

This is a sample configuration of the mixed client port rate in same slice.

```
hw-module location 0/0
 mxponder-slice 0
  trunk-rate 200G
  client-port-rate 2 client-type 100G
  client-port-rate 3 client-type otu4
 !
 mxponder-slice 1
  trunk-rate 400G
  client-port-rate 4 client-type 100G
  client-port-rate 8 client-type otu4
 !
!
```

2.4T, 2.4TX, and 2.4TA card modes

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

Table 19: Feature History

Feature Name	Release Information	Feature Description
Muxponder Slice and Muxponder card modes for NCS1K14-2.4T-A-K9 Line Card	Cisco IOS Release 26.1.1	The new NCS1K14-2.4T-A-K9 line card supports both Muxponder Slice and Muxponder card modes, similar to the 2.4T and 2.4TX transponder cards. It can handle 1.2T of data per trunk and supports client data rates of 100GE and 400GE, with the capability to increase up to 800G per client. By grouping multiple client interfaces, muxponder slice and muxponder card modes enhances bandwidth utilization for efficient transport over high-capacity DWDM links.

Available card modes

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

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

Note The 2.4T card does not support mxponder mode.

2.4T, 2.4TX, and 2.4TA card trunk pluggables and datarates

Coherent Interconnect Module 8

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

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

CIM8-C-K9

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

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

The default frequency is 193.10 THz.

CIM8-CE-K9

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

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

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

CIM8-LE-K9

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

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

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

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

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

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

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

PID	Frequency range supported
CIM8-LE-K9	1861500 to 1909250

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

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

QPSK modulation in CIM8 pluggables for 400G trunk rate

QPSK modulation in CIM8 pluggables refers to a line-side signal configuration.

Until Release 25.2.1, the CIM8 pluggable supported only the PCS-based modulation format for line-side configurations. This format was enabled by default and could not be changed by the user. For a 400G trunk rate, the default setting is 128 GBd with PCS modulation.

From Release 25.3.1, CIM8 pluggables on the NCS1K4-2.4T-K9, NCS1K14-2.4T-X-K9, NCS1K14-2.4TXL-K9 and NCS1K14-2.4T-L-K9 cards support 400G QPSK modulation. When the user configures the baud rate to 118 GBd for a trunk rate of 400G, the system automatically selects QPSK modulation.

Table 20: Feature History

Feature Name	Release Information	Description
QPSK modulation support for 400G trunk rate	Cisco IOS XR Release 25.3.1	The NCS1K4-2.4T-K9, NCS1K14-2.4T-X-K9, NCS1K14-2.4TXL-K9, and NCS1K14-2.4T-L-K9 cards now support QPSK modulation configuration for 400G trunk rate on CIM8 pluggables at a baud rate of 118 GBd. This enhancement delivers improved performance and extended reach for long-distance and subsea applications.

Verify QPSK modulation in CIM8 pluggables

You can confirm that QPSK modulation is correctly enabled and functioning on the CIM8 pluggables.

Procedure

Run the **show controller** *controllertype R/S/I/P* command to verify the QPSK modulation on the CIM8 pluggables for baud rate 118.

Example:

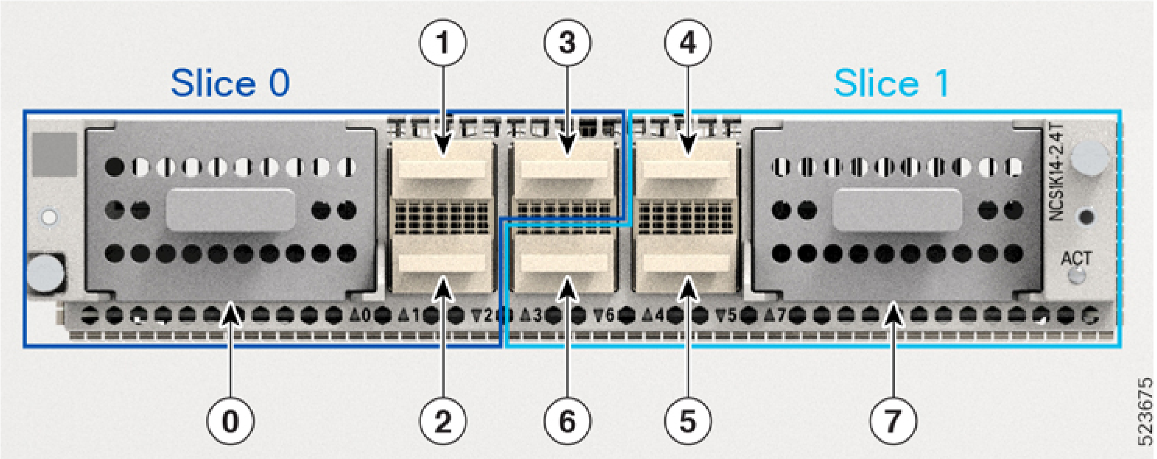
```
RP/0/RP0/CPU0:NE2063#show controllers optics 0/2/0/0
Wed Aug 13 08:59:58.896 UTC
 Controller State: Up
 Transport Admin State: In Service
 Laser State: On
 LED State: Green
 Last link flapped: 00:13:23

 Optics Status
         Optics Type:  CIM8 DWDM
         DWDM carrier Info: C BAND, MSA ITU Channel=1, Frequency=196.10THz,
         Wavelength=1528.773nm

         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 = 3
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = 2.01 dBm
         RX Power = 1.28 dBm
         RX Signal Power = 1.35 dBm
         Frequency Offset = -61 MHz

         Performance Monitoring: Enable

         THRESHOLD VALUES
         ----------------
         Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
         ------------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)         13.0      -25.0           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 = 600000 ps/nm
         Configured CD lower Threshold = -600000 ps/nm
         Configured OSNR lower Threshold = 14.50 dB
         Configured DGD Higher Threshold = 103.00 ps
         Baud Rate =  118.7111970000 GBd
         Bits per Symbol = 2.0000000000  bits/symbol
         Modulation Type: QPSK
         Chromatic Dispersion -1 ps/nm
         Configured CD-MIN -10000 ps/nm  CD-MAX 90000 ps/nm
         Polarization Mode Dispersion = 0.0 ps
         Second Order Polarization Mode Dispersion = 40.00 ps^2
         Optical Signal to Noise Ratio = 40.00 dB
         SNR = 19.30 dB
         Polarization Dependent Loss = 1.40 dB
         Polarization Change Rate = 0.00 rad/s
         Differential Group Delay = 2.00 ps
         Filter Roll Off Factor : 0.100
         Rx VOA Gain Offset : 0.00 dB
         NLEQ Compensation Mode : 0
         Cross Polarization Gain Mode : 10
         Proprietary Submarine Parameters
           Type : 1        Value : 0
           Type : 2        Value : 0
           Type : 3        Value : 0
           Type : 4        Value : 0
           Type : 5        Value : 10485760
           Type : 6        Value : 0
 Transceiver Vendor Details
         Form Factor            : CIM8
         Name                   : CISCO-ACACIA
         Part Number            : 10-100471-02
         Rev Number             : A0
         Serial Number          : ACA29160037
         PID                    : CIM8-C-K9
         VID                    : V02
         Firmware Version       : Major.Minor.Build
         Active                 : 80.140.8
         Inactive               : 80.130.21
         Date Code(yy/mm/dd)    : 25/04/10
         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

RP/0/RP0/CPU0:NE2063#
```

Muxponder slice mode for 2.4T, 2.4TX, and 2.4TA cards

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

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

Figure 2: 2.4T lne card slices and ports



2.4TX and 2.4TA

Figure 3: line card slices and ports

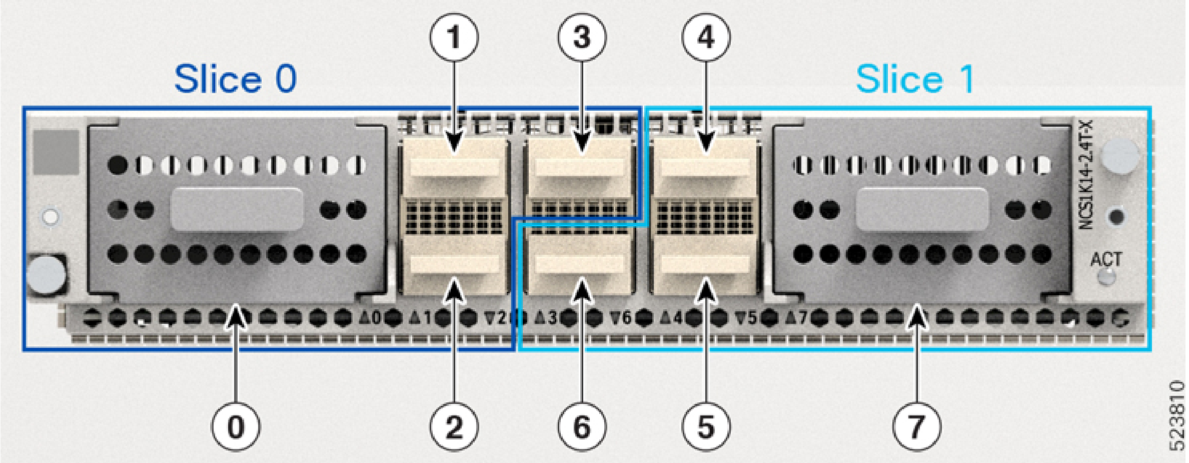


Table 21: Client-to-trunk mapping in slice 0 and slice 1 modes

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

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

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

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

Table 22: Release-wise trunk rates supported by the 2.4T, 2.4TX, and 2.4TA cards

Trunk rate (G)	2.4T	2.4TX	2.4TA
400	7.11.1	24.1.1	26.1.1
500	—	24.1.1	26.1.1
600	7.11.1	24.1.1	26.1.1
700	-	24.2.1	26.1.1
800	7.11.1	24.1.1	26.1.1
900	-	24.2.1	26.1.1
1000	7.11.1	24.1.1	26.1.1
1100	-	24.2.1	26.1.1
1200	-	24.1.1	26.1.1

Recommended trunk parameters in the 2.4T, 2.4TX, and 2.4TA cards

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

Baud rate ranges for each trunk rate in the 2.4T, 2.4TX, 2.4TA, and 2.4TA-L cards

The 2.4T, 2.4TX, 2.4TA, and 2.4TA-L cards carries signals at different trunk rates, with each trunk rate operating within a baud rate range.

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

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

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

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

Table 23: Baud rate and bit rate range for each trunk rate in the 2.4TX and 2.4TA card

Trunk data rate per trunk (G)	Minimum baud r ate (GBd)	Maximum baud rate (GBd)	Default baud rate (GBd)	Minimum bit per second (bps)	Maximum bit per second (bps)
400	62	128	127.931418	2.1	4.1
500	62	138	137.834059	2.5	5
600	72	138	137.738007	2.8	5.1

Trunk data rate per trunk (G)	Minimum baud r ate (GBd)	Maximum baud rate (GBd)	Default baud rate (GBd)	Minimum bit per second (bps)	Maximum bit per second (bps)
700	88	138	138.08166	3.2	5
800	98	138	137.978388	3.5	5.1
900	108	138	137.89817	3.8	5.2
1000	99.22392	138	137.834059	4.3	5.3
1100	118	138	137.78165	4.7	5.3
1200	128	138	137.738007	5.3	5.7

Customize baud rates

Customizing baud rates is to enable users to adjust the default baud rates for the 2.4T, 2.4TX, and 2.4TA cards, particularly when operating in muxponder mode. This customization allows for optimization of network bandwidth based on specific deployment scenarios and ensures that the baud rates align with the available bandwidth in the network.

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

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

Before you begin

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

Procedure

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

Example:

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

Step 2 Enter baud rate.

Example:

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

Step 3 Save the changes.

Example:

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

Client pluggables for configuring muxponder slice modes

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

Pluggable combinations in muxponder slice modes

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

Table 24: Trunk rate and client pluggable combinations for Slices 0 and Slice 1

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

Trunk rate (G) per trunk	Card support	Client rate	Client pluggable	Client ports	
				Slice 0	Slice 1
900	2.4T, 2.4TX , 2.4TA	2x 100 GE 2x 400 GE + 1x 100 GE	QDD-400G + QDD-4x100G	1, 2, 3	4, 5, 6
		400 GE + 5x 100 GE	QDD-400G + QDD-4x100G		
		9x 100 GE	3x QDD-4x100G		
1000	2.4T, 2.4TX, 2.4TA	2x 100 GE 2x 400GE + 2x 100 GE	2x QDD-400G + 2x QDD-4x100G	1, 2, 3	4, 5, 6
		10x 100 GE	3x QDD-4x100G		
1100	2.4TX , 2.4TA	2x 100 GE 2x 400 GE + 3x 100 GE	2x QDD-400G + QDD-4x100G	1, 2, 3	4, 5, 6
		400 GE + 7x 100 GE	2x QDD-400G + QDD-4x100G		
		11x 100 GE	3x QDD-4x100G		
1200	2.4TX	3x 400 GE	3x QDD-400G	1, 2, 3	4, 5,6
		2x 400 GE + 4x 100 GE	2x QDD-400G + QDD-4x100G		
		400 GE + 8x 100 GE	QDD-400G + 2x QDD-4x100G		
		12x 100 GE	3x QDD-4x100G		
		6x 2X100 GE	6x QDD-2X100-CWDM4-S	1, 2, 3, 4, 5, 6	—
			6x QDD-2X100-LR4-S		

[1] QDD-400G refers to QDD-400G-FR4-S, QDD-400G-LR4-S, QDD-400G-AOCxM, and QDD-400G-DR4-S pluggable modules.

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

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

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

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

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

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

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

For more information on the the data rate on each client port, see Client pluggables for configuring muxponder slice modes, on page 51 .

Before you begin

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

Procedure

Step 1 Specify the card location.

Example:

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

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

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

Example:

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

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

Example:

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

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

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

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

Example:

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

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

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

Example:

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

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

Example:

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

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

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

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

Example:

Command

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

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

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

Example:

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

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

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

Example:

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

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

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

For more information on the data rate on each client port, see Client pluggables for configuring muxponder slice modes, on page 51.

Before you begin

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

Procedure

Step 1 Specify the card location.

Example:

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

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

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

Example:

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

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

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

Example:

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

Example:

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

Example:

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

Example:

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

Example:

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

Step 4 Set up the client rate.

For the 2X100G pluggables, run this command.

Example:

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

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

Example:

Command

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

Step 6 Verify the data rate that you set up.

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

Example:

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

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

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

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

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

Example:

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

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

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

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

Example:

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

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

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

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

Example:

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

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

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

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

Example:

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

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

Muxponder mode for 2.4TX and 2.4TA

The muxponder mode enables the 2.4TX and 2.4TA cards 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 and 2.4TA cards activates the split client ports. The shared client ports are client port 2 for 600G and client port 3 for 1000G.

How muxponder mode splits 400GE and 4x100GE client traffic

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

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

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

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

Data rate capabilities for the 2.4TX and 2.4TA card

Table 25: Feature History

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

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

The 2.4TX and 2.4TA card supports different trunk rates.

Table 26: Release-wise trunk rates supported by the 2.4TX and 2.4TA cards

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

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

Client pluggables for configuring 2.4TX and 2.4TA muxponder mode

Table 27: Feature History

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

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

Client pluggable combinations in muxponder mode

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

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

Note The 2.4TA card does not support 2x100GE.

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

Table 28: 2.4TX and 2.4TA muxponder mode port configurations

Trunk rate (G) per trunk	Total configured trunk rate (G)	Client channel rate	Client pluggable	Shared client port	Client ports
500	1000	2x 400GE + 2x 100GE	2x QDD-400G + 1x QDD-4x100G	2	1, 4
		1x 400GE + 6x 100GE	1x QDD-400G + 2x QDD-4x100G		
		10x 100GE	3x QDD-4x100G		
600	1200	3x 400GE	3x QDD-400G	2	1, 4
		2x 400GE + 4x 100GE	2x QDD-400G + 1x QDD-4x100G		
		1x 400GE + 8x 100GE	1x QDD-400G + 2x QDD-4x100G		
900	1800	4x 400GE + 2x 100GE	4x QDD-400G + 1x QDD-4x100G	3	1, 2, 4, 5
		3x 400GE + 6x 100GE	3x QDD-400G + 2x QDD-4x100G		
		2x 400GE + 10x 100GE	2x QDD-400G + 3x QDD-4x100G		
		1x 400GE + 14x 100GE	1x QDD-400G + 4x QDD-4x100G		
		18x 100GE	5x QDD-4x100G		

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

Understanding client rates per client port for each trunk rate

This table shows the sample client rate per client port for each trunk rate. This simplified matrix helps you understand the traffic flow in each client port. It also indicates the number of channels that each client port uses to deliver the client traffic. The type of pluggable module inserted in the shared client port determines the traffic rate through breakout and non-breakout channels.

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

Table 29: Client rate traffic per trunk rate and client pluggable combinations

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

Trunk rate (G) per trunk	Client pluggable	Client rate (GE) per trunk 0 client ports		Client rate (GE) per shared client ports		Client rate (GE) per trunk 1 client ports		
		1	2	2	3	4	5	6
600	3x QDD-400G	400	-	400	-	400	-	-
	2x QDD-400G + 1x QDD-4x100G	400	-	4x 100 [3]	-	400	-	-
	1x QDD-400G + 2x QDD-4x100G + 3x QDD-4x100G	400	-	4x 100 [3]	-	4x 100	-	-
900	4x QDD-400G + 1x QDD-4x100G	400	400	-	2x 100 [3]	400	400	-
	3x QDD-400G + 2x QDD-4x100G	400	400	-	2x 100 [3]	400	4x 100	-
	2x QDD-400G + 3x QDD-4x100G	400	400	-	2x 100 [3]	4x 100	4x 100	-
	1x QDD-400G + 4x QDD-4x100G	400	4x 100	-	2x 100 [3]	4x 100	4x 100	-
	5x QDD-4x100G	4x 100	4x 100	-	2x 100 [3]	4x 100	4x 100	-
1000	5x QDD-400G	400	400	-	400	400	400	-
	4x QDD-400G + 1x QDD-4x100G	400	400	-	4x 100	400	400	-
	3x QDD-400G + 2x QDD-4x100G	400	400	-	4x 100	400	4x 100	-
	2x QDD-400G + 3x QDD-4x100G	400	400	-	4x 100	4x 100	4x 100	-
	1x QDD-400G + 4x QDD-4x100G + 5x QDD-4x100G	400	4x 100	-	4x 100	4x 100	4x 100	-

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

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

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

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

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

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

Note For the 600G trunk rate, the split port supports both 400GE and 4x100GE. For more information on required pluggable modules for other trunk rates, see Client pluggables for configuring 2.4TX and 2.4TA muxponder mode, on page 60 .

Before you begin

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

Procedure

Step 1 Specify the card location.

Example:

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

Step 2 Enter the muxponder card mode.

Example:

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

Step 3 Set up the trunk rate.

Example:

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

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

Example:

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

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

Note

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

Step 5 Save the configuration and exit the muxponder mode.

Example:

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

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

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

Example:

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

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

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

Reset client optics remotely

Use this procedure to remotely reset client optics with a new CLI exec command. This enables cold reboot options to troubleshoot optics or host platform issues.

Table 30: Feature History Table

Feature Name	Release Information	Description
Remote reset of client optics	Cisco IOS XR Release 25.4.1	You can now remotely reset client optics using a new CLI command and Yang model. This feature allows you to perform cold reboots of optics to resolve issues caused by the host or the optics. Remote power cycling enables more flexible and programmable management of optics. This capability does not apply to CIM 8. Supported optics are: • 400GE • 4x100GE • All QSFP optics • ZR optics, including QDD-ZR, QDD-ZR+, 400G-ZR, and 400G Bright ZR+ CLI: The command `reload transceiver Optics R/S/I/P cold` is introduced. Yang model: The new yang model `Cisco-IOS-XR-reset-optics-act.yang` is introduced.

This procedure enables remote power cycling of optics in response to issues arising from the host platform or the optics. It supports flexible and programmable management through CLI. This feature is supported on the 400GE, 4x100GE, ZR optics, including QDD-ZR, QDD-ZR+, 400G-ZR, and 400G Bright ZR+ optics.

Limitations:

- Warm reloading is not supported.
- CIM-8 pluggables are not supported.

Procedure

Enter the command **reload transceiver optics** *R/S/I/P* **[cold]** to reset the optics for a specific port.

Example:

```
RP/0/RP0/CPU0:ios#reload transceiver optics 0/0/0/2 [ cold ]
```

The default is a cold reset, regardless of whether you specify cold or not.

The selected 400GE or 4x100GE optics module is reset remotely, performing a cold reboot as needed for troubleshooting or management.