OpenConfig Support for EDFA2 Card

Table 1: Feature History

Feature Name	Release Information	Feature Description
Enhancement in OC support for EDFA2 card	Cisco IOS XR Release 25.2.1	The EDFA2 card now supports these OpenConfig models for retrieving operational and real-time telemetry data. • `openconfig-channel-monitor.yang` • `openconfig-transport-line-common.yang` In addition to the previously supported operational and telemetry data retrieval, these models now support configuration on the EDFA2 cards. • `openconfig-optical-amplifier.yang` • `openconfig-optical-attenuator.yang` • `openconfig-wavelength-router.yang`

Table 2: Feature History

Feature Name	Release Information	Feature Description
OC support for EDFA2 card	Cisco IOS XR Release 25.1.1	The Open Configuration (OC) support is introduced for the EDFA2 card and the OTDR pluggable. This enables you to retrieve the operational data and real-time telemetry data using these data models: • `openconfig-optical-amplifier.yang` • `openconfig-optical-attenuator.yang` • `openconfig-wavelength-router.yang` • `openconfig/gnoi/OTDR.proto`

EDFA2 card

The EDFA2 line card is an optical amplifier for the NCS1014 Chassis. It functions as a DWDM optical terminal and includes a C-band bidirectional amplifier with channel power control capabilities. This card supports Optical Supervisory Channel (OSC) and Optical Time Domain Reflectometer (OTDR) functionalities.

The card comprises an optical module, pluggable cages for OTDR and OSC, and a DWDM trunk interface. It features integrated management for alarms, performance monitoring, and optical power level control.

Supported OpenConfig Yang models

The NCS1K14-EDFA2 card supports these OpenConfig models:

Note In Release 25.1.1, only MDT is supported, and EDT is not supported. Additionally, only operational data is supported.

Table 3: Supported OC models

Release	Model	Feature
R25.1.1	openconfig-optical-amplifier.yang	State of the optical amplifier.
	openconfig-optical-attenuator.yang	State of the optical attenuator.
	openconfig-wavelength-router.yang	State of the media channels and optical interfaces.
	openconfig/gnoi/OTDR.proto	Initiating forced OTDR scan.
R25.2.1	openconfig-channel-monitor.yang	State of the optical channel monitor.
	openconfig-transport-line-common.yang	State of the optical transport line system.

Supported functions for OpenConfig models

This table highlights the functions supported by the OpenConfig models in releases 25.1.1 and 25.2.1

Table 4: Supported functions for OpenConfig models

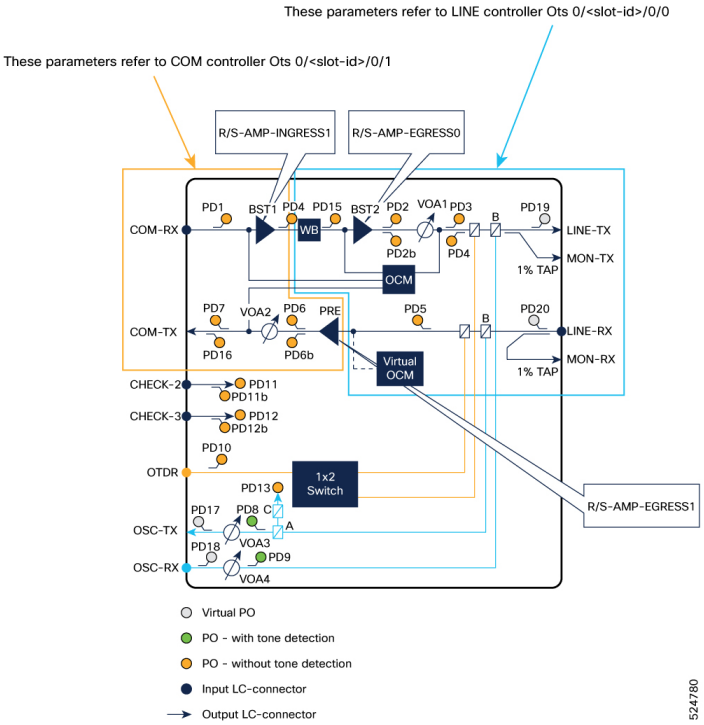
OC model	Operational data support	Configuration support	EDT support
configure-optical-amplifier.yang	R25.1.1	R25.2.1	—
openconfig-optical-attenuator.yang	R25.1.1	R25.2.1	—
openconfig-wavelength-router.yang	R25.1.1	R25.2.1	R25.2.1
openconfig/gnoi/OTDR.proto	R25.1.1	—	—
openconfig-channel-monitor.yang	R25.2.1	—	—
openconfig-transport-line-common.yang	R25.2.1	—	—

Naming conventions

This section describes the naming conventions used in the OpenConfig models supported by the NCS1K14-EDFA2 card.

OpenConfig amplifier model

Figure 1: Optical diagram of NCS1K14-EDFA2 card representing amplifiers



In this optical block diagram, amplifiers are categorized as follows:

- **BST1**: Represented as an ingress amplifier.
- **BST2**: Represented as an egress amplifier.
- **PRE amplifier parameters**: Represented as egress amplifiers.

The naming convention for amplifiers is structured as follows:

`R/S-AMP-<AMP-TYPE>P`

Where:

- **R** stands for Rack.
- **S** stands for Slot.
- **AMP** is the suffix used to denote an amplifier.
- **AMP-TYPE** indicates the type of amplifier.
- **P** stands for Port ID.

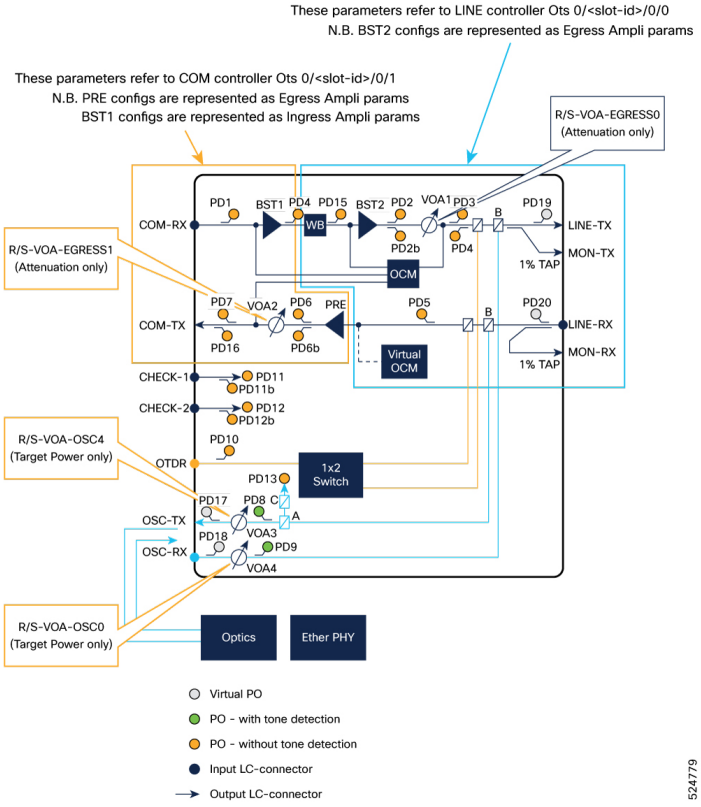
Examples:

- `R/S-AMP-EGRESS0`
- `R/S-AMP-EGRESS1`

- `R/S-AMP-INGRESS1`

OpenConfig attenuator model

Figure 2: Optical diagram of NCS1K14-EDFA2 card representing attenuators



The naming convention for the attenuator model is as follows:

- `R/S-VOA-EGRESS0`
- `R/S-VOA-EGRESS1`
- `R/S-VOA-OSC0`
- `R/S-VOA-OSC4`

Where:

- **R** stands for Rack.
- **S** stands for Slot.
- **VOA** indicates a Variable Optical Attenuator.
- The suffix specifies the type and port number, such as **EGRESS** or **OSC**.

OpenConfig wavelength model

Openconfig supplies a 3-byte value called the ‘index’. This value is interpreted as follows:

- **First Byte**: Represents the channel identifier, which can range from 1 to 97.
- **Second and Third Bytes**: Indicate the rack and slot numbers.

Example:

If Openconfig provides an ‘index’ value of "100":

- **Channel Identifier**: 1
- **Node Location Name**: 0/0/NXR0 on the native side

This means the channel identifier is 1, and because the rack and slot are fixed at 0, the location name is formatted as "0/0/NXR0".

OpenConfig channel monitor model

The naming convention for the channel monitor:

- `R/S-CHMON-RX-0`
- `R/S-CHMON-TX-0`
- `R/S-CHMON-RX-1`
- `R/S-CHMON-TX-1`

OpenConfig transport line common model

As safety ports are already modeled in the inventory, the transport line common model will utilize the same `ots R/S/I/P` representation.

Structure of Yang models supported on NCS1K14-EDFA2

This structure describes the leave structure of YANG models supported on NCS1K14-EDFA2 card:

Note The leaves that are in bold are augmented.

Supported leaves for OpenConfig attenuator model

From R25.1.1, these leaves are supported for get-oper and telemetry:

```
+--rw optical-attenuator
     +--rw attenuators
     |  +--rw attenuator* [name]
     |     +--rw name                                    -> ../config/name
     |     +--rw config
     |     |  +--rw name?                                string
     |     |  +--rw target-output-power?                 decimal64
     |     |  +--rw attenuation?                         decimal64
     |     +--ro state
     |        +--ro name?                                 string
     |        +--ro attenuation-mode?           identityref
     |        +--ro target-output-power?       decimal64
```

```
|        +--ro attenuation?                       decimal64
|        +--ro enabled?                            boolean
|       +--ro component?                      -> /oc-platform:components/component/name

|        +--ro actual-attenuation
|        |  +--ro instant?                          decimal64
|        |  +--ro output-power-total
|        |  +--ro instant?                          decimal64
|        |  +--ro avg?                                decimal64
|        |  +--ro min?                                decimal64
|        |  +--ro max?                               decimal64
|        |  +--ro interval?                          oc-types:stat-interval
|        |  +--ro min-time?                         oc-types:timeticks64
|        |  +--ro max-time?                         oc-types:timeticks64
|        +--ro optical-return-loss
|        |  +--ro instant?                          decimal64
|        |  +--ro avg?                                    decimal64
|        |  +--ro min?                                 decimal64
|        |  +--ro max?                                decimal64
|        |  +--ro interval?                          oc-types:stat-interval
|        |  +--ro min-time?                         oc-types:timeticks64
|        |  +--ro max-time?                        oc-types:timeticks64
```

From R25.2.1, these leaves are supported for NETCONF/GNMI operations, including get, edit-config, and telemetry:

```
+--rw optical-attenuator
     +--rw attenuators
     |  +--rw attenuator* [name]
     |     +--rw name                                   -> ../config/name
     |     +--rw config
     |     |  +--rw name?                              string
     |     |  +--rw attenuation-mode?          identityref
     |     |  +--rw target-output-power?      decimal64
     |     |  +--rw attenuation?                      decimal64
     |     |  +--rw enabled?                            boolean
     |     +--ro state
     |        +--ro name?                               string
     |        +--ro attenuation-mode?           identityref
     |        +--ro target-output-power?       decimal64
     |        +--ro attenuation?                       decimal64
     |        +--ro enabled?                             boolean
     |       +--ro component?                      -> /oc-platform:components/component/name

     |          |        +--ro actual-attenuation
     |        |  +--ro instant?                          decimal64
     |        |        +--ro output-power-total
     |        |  +--ro instant?                          decimal64
     |        |  +--ro avg?                                decimal64
     |        |  +--ro min?                                decimal64
     |        |  +--ro max?                               decimal64
     |        |  +--ro interval?                          oc-types:stat-interval
     |        |  +--ro min-time?                         oc-types:timeticks64
     |        |  +--ro max-time?                         oc-types:timeticks64
     |        +--ro optical-return-loss
     |        |  +--ro instant?                          decimal64
     |        |  +--ro avg?                                decimal64
     |        |  +--ro min?                                decimal64
     |        |  +--ro max?                               decimal64
     |        |  +--ro interval?                          oc-types:stat-interval
     |        |  +--ro min-time?                         oc-types:timeticks64
     |        |  +--ro max-time?                        oc-types:timeticks64
```

```
 +--ro state
     |        +--ro name?                                    string
     |        +--ro target-output-power?       decimal64
     |        +--ro attenuation?                        decimal64
     |        +--ro enabled?                             boolean
     |       +--ro component?                    -> /oc-platform:components/component/name

     |        +--ro actual-attenuation
     |        |  +--ro instant?                             decimal64
     |        +--ro output-power-total
     |        |  +--ro instant?                             decimal64
     |        |  +--ro avg?                                   decimal64
     |        |  +--ro min?                                   decimal64
     |        |  +--ro max?                                  decimal64
     |        |  +--ro interval?                             oc-types:stat-interval
     |        |  +--ro min-time?                            oc-types:timeticks64
     |        |  +--ro max-time?                            oc-types:timeticks64
     |        +--ro optical-return-loss
     |        |  +--ro instant?                               decimal64
     |        |  +--ro avg?                                     decimal64
     |        |  +--ro min?                                     decimal64
     |        |  +--ro max?                                    decimal64
     |        |  +--ro interval?                               oc-types:stat-interval
     |        |  +--ro min-time?                              oc-types:timeticks64
     |        |  +--ro max-time?                             oc-types:timeticks64
```

Supported leaves for OpenConfig amplifier model

From R25.1.1, these leaves are supported for get-oper and telemetry:

```
|  +--rw amplifier* [name]
     |     +--rw name      -> ../config/name
     |     +--rw config
     |     |  +--rw name?                 string (Amplifier name)
     |     |  +--rw type?                 identityref
     |     |  +--rw target-gain?          decimal64
     |     |  +--rw target-gain-tilt?     decimal64
     |     |  +--rw gain-range?           identityref (LOW: Normal, High: Extended)
     |     |  +--rw amp-mode?             identityref
     |     |  +--rw enabled?              boolean
     |     |  +--rw fiber-type-profile?   identityref
     |     +--ro state
     |        +--ro name?                 string
     |        +--ro type?                 identityref
     |        +--ro target-gain?          decimal64
     |        +--ro target-gain-tilt?     decimal64
     |        +--ro gain-range?           identityref
     |        +--ro amp-mode?             identityref
     |        +--ro enabled?              boolean
     |        +--ro fiber-type-profile?   identityref
     |        +--ro component?            -> /oc-platform:components/component/name
     |        +--ro actual-gain
     |        |  +--ro instant?    decimal64
     |        |  +--ro avg?        decimal64
     |        |  +--ro min?        decimal64
     |        |  +--ro max?        decimal64
     |        |  +--ro interval?   oc-types:stat-interval
     |        |  +--ro min-time?   oc-types:timeticks64
     |        |  +--ro max-time?   oc-types:timeticks64
     |        +--ro actual-gain-tilt
     |        |  +--ro instant?    decimal64
     |        |  +--ro avg?        decimal64
     |        |  +--ro min?        decimal64
     |        |  +--ro max?        decimal64
```

```
|        |  +--ro interval?   oc-types:stat-interval
|        |  +--ro min-time?   oc-types:timeticks64
|        |  +--ro max-time?   oc-types:timeticks64
|        +--ro input-power-total
|        |  +--ro instant?    decimal64.
|        |  +--ro avg?        decimal64
|        |  +--ro min?        decimal64
|        |  +--ro max?        decimal64
|        |  +--ro interval?   oc-types:stat-interval
|        |  +--ro min-time?   oc-types:timeticks64
|        |  +--ro max-time?   oc-types:timeticks64
|        +--ro input-power-c-band.
|        |  +--ro instant?    decimal64
|        |  +--ro avg?        decimal64
|        |  +--ro min?        decimal64
|        |  +--ro max?        decimal64
|        |  +--ro interval?   oc-types:stat-interval
|        |  +--ro min-time?   oc-types:timeticks64
|        |  +--ro max-time?   oc-types:timeticks64
|        +--ro output-power-total -→ total tx power
|        |  +--ro instant?    decimal64
|        |  +--ro avg?        decimal64
|        |  +--ro min?        decimal64
|        |  +--ro max?        decimal64
|        |  +--ro interval?   oc-types:stat-interval
|        |  +--ro min-time?   oc-types:timeticks64
|        |  +--ro max-time?   oc-types:timeticks64
|        +--ro output-power-c-band → tx signal power
|        |  +--ro instant?    decimal64
|        |  +--ro avg?        decimal64
|        |  +--ro min?        decimal64
|        |  +--ro max?        decimal64
|        |  +--ro interval?   oc-types:stat-interval
|        |  +--ro min-time?   oc-types:timeticks64
|        |  +--ro max-time?   oc-types:timeticks64
|        +--ro optical-return-loss
|           +--ro instant?    decimal64
|           +--ro avg?        decimal64
|           +--ro min?        decimal64
|           +--ro max?        decimal64
|           +--ro interval?   oc-types:stat-interval
|           +--ro min-time?   oc-types:timeticks64
|           +--ro max-time?   oc-types:timeticks64
+--rw supervisory-channels
   +--rw supervisory-channel* [interface]
      +--rw interface    -> ../config/interface
      +--rw config
      |  +--rw interface?   oc-if:base-interface-ref
      +--ro state
         +--ro interface?           oc-if:base-interface-ref
         +--ro input-power. → Rx power
         |  +--ro instant?    decimal64
         |  +--ro avg?        decimal64
         |  +--ro min?        decimal64
         |  +--ro max?        decimal64
         |  +--ro interval?   oc-types:stat-interval
         |  +--ro min-time?   oc-types:timeticks64
         |  +--ro max-time?   oc-types:timeticks64
         +--ro output-power. → Tx power
         |  +--ro instant?    decimal64
         |  +--ro avg?        decimal64
         |  +--ro min?        decimal64
         |  +--ro max?        decimal64
```

```
            |  +--ro interval?   oc-types:stat-interval
            |  +--ro min-time?   oc-types:timeticks64
            |  +--ro max-time?   oc-types:timeticks64
```

From R25.2.1, these leaves are supported for NETCONF/GNMI operations, including get, edit-config, and telemetry:

```
+-rw optical-amplifier
     +--rw amplifiers
     |  +--rw amplifier* [name]
     |     +--rw name      -> ../config/name
     |     +--rw config
     |     |  +--rw name?                 string
     |     |  +--rw type?                 identityref
     |     |  +--rw target-gain?          decimal64
     |     |  +--rw target-gain-tilt?     decimal64
     |     |  +--rw gain-range?           identityref
     |     |  +--rw amp-mode?             identityref
     |     |  +--rw enabled?              boolean
     |     |  +--rw fiber-type-profile?   identityref
     |     +--ro state
     |        +--ro name?                 string
     |        +--ro type?                 identityref
     |        +--ro target-gain?          decimal64
     |        +--ro target-gain-tilt?     decimal64
     |        +--ro gain-range?           identityref
     |        +--ro amp-mode?             identityref
     |        +--ro enabled?              boolean
     |        +--ro fiber-type-profile?   identityref
     |        +--ro component?            -> /oc-platform:components/component/name
     |        +--ro actual-gain
     |        |  +--ro instant?    decimal64
     |        |  +--ro avg?        decimal64
     |        |  +--ro min?        decimal64
     |        |  +--ro max?        decimal64
     |        |  +--ro interval?   oc-types:stat-interval
     |        |  +--ro min-time?   oc-types:timeticks64
     |        |  +--ro max-time?   oc-types:timeticks64
     |        +--ro actual-gain-tilt
     |        |  +--ro instant?    decimal64
     |        |  +--ro avg?        decimal64
     |        |  +--ro min?        decimal64
     |        |  +--ro max?        decimal64
     |        |  +--ro interval?   oc-types:stat-interval
     |        |  +--ro min-time?   oc-types:timeticks64
     |        |  +--ro max-time?   oc-types:timeticks64
     |        +--ro input-power-total
     |        |  +--ro instant?    decimal64
     |        |  +--ro avg?        decimal64
     |        |  +--ro min?        decimal64
     |        |  +--ro max?        decimal64
     |        |  +--ro interval?   oc-types:stat-interval
     |        |  +--ro min-time?   oc-types:timeticks64
     |        |  +--ro max-time?   oc-types:timeticks64
     |        +--ro input-power-c-band
     |        |  +--ro instant?    decimal64
     |        |  +--ro avg?        decimal64
     |        |  +--ro min?        decimal64
     |        |  +--ro max?        decimal64
     |        |  +--ro interval?   oc-types:stat-interval
     |        |  +--ro min-time?   oc-types:timeticks64
     |        |  +--ro max-time?   oc-types:timeticks64
     |        +--ro output-power-total -→ total tx power
     |        |  +--ro instant?    decimal64
```

```
|        |  +--ro avg?        decimal64
|        |  +--ro min?        decimal64
|        |  +--ro max?        decimal64
|        |  +--ro interval?   oc-types:stat-interval
|        |  +--ro min-time?   oc-types:timeticks64
|        |  +--ro max-time?   oc-types:timeticks64
|        +--ro output-power-c-band → tx signal power
|        |  +--ro instant?    decimal64
|        |  +--ro avg?        decimal64
|        |  +--ro min?        decimal64
|        |  +--ro max?        decimal64
|        |  +--ro interval?   oc-types:stat-interval
|        |  +--ro min-time?   oc-types:timeticks64
|        |  +--ro max-time?   oc-types:timeticks64
|        +--ro optical-return-loss → OPBRR
|           +--ro instant?    decimal64
|           +--ro avg?        decimal64
|           +--ro min?        decimal64
|           +--ro max?        decimal64
|           +--ro interval?   oc-types:stat-interval
|           +--ro min-time?   oc-types:timeticks64
|           +--ro max-time?   oc-types:timeticks64
+--rw supervisory-channels
   +--rw supervisory-channel* [interface]
      +--rw interface    -> ../config/interface
      +--rw config
      |  +--rw interface?   oc-if:base-interface-ref
      +--ro state
         +--ro interface?            oc-if:base-interface-ref
         +--ro input-power. → Rx power
         |  +--ro instant?    decimal64
         |  +--ro avg?        decimal64
         |  +--ro min?        decimal64
         |  +--ro max?        decimal64
         |  +--ro interval?   oc-types:stat-interval
         |  +--ro min-time?   oc-types:timeticks64
         |  +--ro max-time?   oc-types:timeticks64
         +--ro output-power. → Tx power
         |  +--ro instant?    decimal64
         |  +--ro avg?        decimal64
         |  +--ro min?        decimal64
         |  +--ro max?        decimal64
         |  +--ro interval?   oc-types:stat-interval
         |  +--ro min-time?   oc-types:timeticks64
         |  +--ro max-time?   oc-types:timeticks64

|     +--ro state
      |        +--ro name?                  string
      |        +--ro type?                  identityref
      |        +--ro target-gain?           decimal64
      |        +--ro target-gain-tilt?      decimal64
      |        +--ro gain-range?            identityref
      |        +--ro amp-mode?              identityref
      |        +--ro enabled?               boolean
      |        +--ro fiber-type-profile?    identityref -→
      |        +--ro component?             -> /oc-platform:components/component/name
      |        +--ro actual-gain
      |        |  +--ro instant?    decimal64
      |        |  +--ro avg?        decimal64
      |        |  +--ro min?        decimal64
      |        |  +--ro max?        decimal64
      |        |  +--ro interval?   oc-types:stat-interval
      |        |  +--ro min-time?   oc-types:timeticks64
      |        |  +--ro max-time?   oc-types:timeticks64
```

```
     |        +--ro actual-gain-tilt
     |        |  +--ro instant?    decimal64
     |        |  +--ro avg?        decimal64
     |        |  +--ro min?        decimal64
     |        |  +--ro max?        decimal64
     |        |  +--ro interval?   oc-types:stat-interval
     |        |  +--ro min-time?   oc-types:timeticks64
     |        |  +--ro max-time?   oc-types:timeticks64
...........
 |        +--ro optical-return-loss.
     |           +--ro instant?    decimal64
     |           +--ro avg?        decimal64
     |           +--ro min?        decimal64
     |           +--ro max?        decimal64
     |           +--ro interval?   oc-types:stat-interval
     |           +--ro min-time?   oc-types:timeticks64
     |           +--ro max-time?   oc-types:timeticks64

+--rw supervisory-channels
        +--rw supervisory-channel* [interface]
           +--rw interface    -> ../config/interface
           +--rw config
           |  +--rw interface?   oc-if:base-interface-ref
           +--ro state
              +--ro interface?           oc-if:base-interface-ref
              +--ro input-power. → Rx power
              |  +--ro instant?    decimal64
              |  +--ro avg?        decimal64
              |  +--ro min?        decimal64
              |  +--ro max?        decimal64
              |  +--ro interval?   oc-types:stat-interval
              |  +--ro min-time?   oc-types:timeticks64
              |  +--ro max-time?   oc-types:timeticks64
              +--ro output-power. → Tx power
              |  +--ro instant?    decimal64
              |  +--ro avg?        decimal64
              |  +--ro min?        decimal64
              |  +--ro max?        decimal64
              |  +--ro interval?   oc-types:stat-interval
              |  +--ro min-time?   oc-types:timeticks64
              |  +--ro max-time?   oc-types:timeticks64
```

Supported leaves for OpenConfig Wavelength router model

From R25.1.1, these leaves are supported for get-oper and telemetry:

```
 module: openconfig-wavelength-router
  +--rw wavelength-router
     +--rw media-channels
     |  +--rw channel* [index]
     |     +--rw index                    -> ../config/index
     |     +--rw config
     |     |  +--rw index?                            uint32
     |     |  +--rw lower-frequency?                  oc-opt-types:frequency-type
     |     |  +--rw upper-frequency?                  oc-opt-types:frequency-type
     |     |  +--rw admin-status?                     oc-opt-types:admin-state-type
     |     +--ro state
     |     |  +--ro index?                            uint32
     |     |  +--ro lower-frequency?                  oc-opt-types:frequency-type
     |     |  +--ro upper-frequency?                  oc-opt-types:frequency-type
     |     |  +--ro admin-status?                     oc-opt-types:admin-state-type
     |     |  +--ro oper-status?                      enumeration
     |     +--rw source
     |     |  +--rw config
```

```
|     |  |  +--rw port-name?   -> /oc-platform:components/component/name
|     |  +--ro state
|     |     +--ro port-name?   -> /oc-platform:components/component/name
|     +--rw dest
|     |  +--rw config
|     |  |  +--rw port-name?   -> /oc-platform:components/component/name
|     |  +--ro state
|     |     +--ro port-name?   -> /oc-platform:components/component/name
|    +--rw port-spectrum-power-profiles
|    |  +--rw port* [name]
|    |   |   +--rw name                      -> ../config/name
|    |    |  +--rw config
|    |   |  |  +--rw name?   -> /oc-platform:components/component/name
|    |  |  +--ro state
|    | |  |  +--ro name?   -> /oc-platform:components/component/name
```

From R25.2.1, these leaves are supported for NETCONF/GNMI operations, including get, edit-config, and telemetry:

```
module: openconfig-wavelength-router
 +--rw wavelength-router
    +--rw media-channels
    |  +--rw channel* [index]
    |     +--rw index                     -> ../config/index
    |     +--rw config
    |     |  +--rw index?                           uint32
    |     |  +--rw lower-frequency?                 oc-opt-types:frequency-type
    |     |  +--rw upper-frequency?                 oc-opt-types:frequency-type
    |     |  +--rw admin-status?                    oc-opt-types:admin-state-type
    |     +--ro state
    |     |  +--ro index?                           uint32
    |     |  +--ro lower-frequency?                 oc-opt-types:frequency-type
    |     |  +--ro upper-frequency?                 oc-opt-types:frequency-type
    |     |  +--ro admin-status?                    oc-opt-types:admin-state-type
    |     |  +--ro oper-status?                     enumeration
    +--rw port-spectrum-power-profiles
    |  +--rw port* [name]
    |    |   +--rw name                      -> ../config/name
    |     |  +--rw config
    |    |  |  +--rw name?   -> /oc-platform:components/component/name
    |  |  +--ro state
    | |  |  +--ro name?   -> /oc-platform:components/component/name
    | |  +--rw oc-wave-ext:extended
    |  | |  +--rw config
    |  |  |  |  +--rw oc-wave-ext:regulation-enable?   boolean
    |  |  |   +--ro state
    |  |  |   |  +--rw oc-wave-ext:regulation-enable?   boolean
    |    |  +--rw spectrum-power-profile
    |     |   |  +--rw distribution* [lower-frequency upper-frequency]
    |     |   |  |  +--rw lower-frequency    -> ../config/lower-frequency
     |    |   |  |  +--rw upper-frequency    -> ../config/upper-frequency
     |    |   |  |  +--rw config
     |    |   |  |  |  +--rw lower-frequency?   oc-opt-types:frequency-type
     |     |  |  |  |  +--rw upper-frequency?   oc-opt-types:frequency-type
     |     |  |  |  |  +--rw target-power?      decimal64
     |     |  |  |  +--ro state
     |    |  |   |  |     +--ro lower-frequency?   oc-opt-types:frequency-type
     |    |  |       |     +--ro upper-frequency?   oc-opt-types:frequency-type
     |    |  |       |     +--ro target-power?      decimal64


module: openconfig-wavelength-router
  +--rw wavelength-router
     +--rw media-channels
```

```
|  +--rw channel* [index]
|     +--rw index                      -> ../config/index
|     +--ro state
|     |  +--ro index?                            uint32
|     |  +--ro lower-frequency?                  oc-opt-types:frequency-type
|     |  +--ro upper-frequency?                  oc-opt-types:frequency-type
|     |  +--ro admin-status?                     oc-opt-types:admin-state-type
|     |  +--ro oper-status?                      enumeration
|    +--rw port-spectrum-power-profiles
|  +--rw port* [name]
|   |   +--rw name                       -> ../config/name
|    |  +--rw config
|   |  |  +--rw name?   -> /oc-platform:components/component/name
|  |  +--ro state
| |  |  +--ro name?   -> /oc-platform:components/component/name
| |  +--rw oc-wave-ext:extended
|  |  |  |   +--ro state
|  |  |  |   |  +--rw oc-wave-ext:regulation-enable?   boolean
|   |  +--rw spectrum-power-profile
|    |   |  +--rw distribution* [lower-frequency upper-frequency]
|    |   |  |  +--rw lower-frequency    -> ../config/lower-frequency
 |   |   |  |  +--rw upper-frequency    -> ../config/upper-frequency
 |   |   |  |  +--rw config
 |   |   |  |  |  +--rw lower-frequency?   oc-opt-types:frequency-type
 |    |  |  |  |  +--rw upper-frequency?   oc-opt-types:frequency-type
 |    |  |  |  |  +--rw target-power?      decimal64
 |    |  |  |  +--ro state
 |   |  |  |   |  |     +--ro lower-frequency?   oc-opt-types:frequency-type
 |   |  |    +--ro upper-frequency?   oc-opt-types:frequency-type
 |   |  |    +--ro target-power?      decimal64
```

EDT supported leaf

From R25.2.1, telemetry EDT supports both channel addition and deletion. These OpenConfig leaves are included in the EDT functionality:

- `openconfig-wavelength-router/wavelength-router/media-channels/channel[index]/state/index`
- `openconfig-wavelength-router/wavelength-router/media-channels/channel[index]/state/upper-frequency`
- `openconfig-wavelength-router/wavelength-router/media-channels/channel[index]/state/lower-frequency`

Supported leaves for OpenConfig channel monitor model

From R25.2.1, these leaves are supported for get-oper and telemetry:

```
+--rw channel-monitors
      +--rw channel-monitor* [name]
         +--rw name        -> ../config/name
         +--rw config
         +--ro state
         |  +--ro name?           -> /oc-platform:components/component/name
         |  +--ro monitor-port?   -> /oc-platform:components/component/name
         +--rw channels
            +--ro channel* [lower-frequency upper-frequency]
               +--ro lower-frequency    -> ../state/lower-frequency
               +--ro upper-frequency    -> ../state/upper-frequency
               +--ro state
                  +--ro lower-frequency?   oc-opt-types:frequency-type
                  +--ro upper-frequency?   oc-opt-types:frequency-type
                  +--ro power?             decimal64

        +--ro state
         |  +--ro name?           -> /oc-platform:components/component/name
```

```
|  +--ro monitor-port?   -> /oc-platform:components/component/name
+--rw channels
   +--ro channel* [lower-frequency upper-frequency]
      +--ro state
         +--ro lower-frequency?   oc-opt-types:frequency-type
         +--ro upper-frequency?   oc-opt-types:frequency-type
         +--ro power?             decimal64
```

Supported leaves for OpenConfig transport line common model

From R25.2.1, these leaves are supported for get-oper and telemetry:

```
module: openconfig-transport-line-common
  augment /oc-platform:components/oc-platform:component/oc-platform:port:
    +--rw optical-port
       +--ro state
          +--ro admin-state?         oc-opt-types:admin-state-type
          |  +--ro instant?    decimal64
```

```
module: openconfig-transport-line-common
  augment /oc-platform:components/oc-platform:component/oc-platform:port:
    +--rw optical-port
       +--ro state
          +--ro admin-state?         oc-opt-types:admin-state-type
          |  +--ro instant?    decimal64
```

Sample configurations

This section presents sample configurations and telemetry outputs for the OC models supported on the EDFA2 card.

OpenConfig attenuator model

This is a sample to configure and get operational data using the OC-Attenuator model:

```
{
    "openconfig-optical-attenuator:optical-attenuator": {
       "attenuators": {
           "attenuator": [
               {
                   "name":"0/0-VOA-EGRESS0",
"state":{
                   }
              }


{
    "openconfig-optical-attenuator:optical-attenuator": {
       "attenuators": {
           "attenuator": [
               {
                   "name":"0/0-VOA-OSC0",
                   "state":{
                   }
               }
```

OpenConfig amplifier model

This is a sample to configure and get operational data using the OC-Amplifier model:

```
 <get>
<filter>
```

```
<optical-amplifier xmlns="http://openconfig.net/yang/optical-amplfier">
<amplifiers/>
 <supervisory-channels/>
</optical-amplifier>
</filter>
</get>
```

```
<edit-config>
<target>
<candidate/>
 </target>
<config>
<optical-amplifier xmlns="http://openconfig.net/yang/optical-amplfier">
   <amplifiers>
    <amplifier>
     <name>0/0-AMP-EGRESS0</name>
     <config>
      <name>0/0-AMP-EGRESS0</name>
      <amp-mode>CONSTANT_GAIN</amp-mode>
      <type>EDFA</type>
<fiber-type-profile>SSMF</fiber-type-profile>
      <target-gain>16.00</target-gain>
      <target-gain-tilt>0.00</target-gain-tilt>
      </config>
  <cisco xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-openconfig-optical-amplifier-ext">
      <extended>
       <config>
        <regulation-enabled>true</regulation-enabled>
       </config>
      </extended>
     </cisco>
    </amplifier>
   </amplifiers>
  </optical-amplifier>
</config>
</edit-config>
```

OpenConfig wavelength model

This is a sample to configure and get operational data using the OC-Wavelength model:

```
"openconfig-wavelength-router:wavelength-router": {
  "port-spectrum-power-profiles": {
   "port": [
    {
     "name": "Ots0/0/0/0",
     "Cisco-IOS-XR-openconfig-wavelength-router-ext:extended": {
      "state": {
       "regulation-enable": false
      }
     },
     "state": {
      "name": "Ots0/0/0{
/0"

  },
  "media-channels": {
   "channel": [
    {
     "index": 100,
     "state": {
      "index": 100,
      "lower-frequency": "193950000",
      "upper-frequency": "194050000",
```

```
      "admin-status": "enabled",
      "oper-status": "up"
     }
```

Wavelength router model

This is a sample to configure and get operational data for frequency convention in spectrum-power-profile and distribution for wavelength router model:

```
<edit-config>
<target><candidate/></target>
<config>
    <wavelength-router xmlns="http://openconfig.net/yang/wavelength-router">
      <port-spectrum-power-profiles>
        <port>
          <name>Ots0/0/0/0</name>
            <config>
              <name>Ots0/0/0/0</name>
            </config>
          <spectrum-power-profile>
            <distribution>
              <lower-frequency>191200000</lower-frequency>
              <upper-frequency>196175000</upper-frequency>
<config>
                <lower-frequency>191200000</lower-frequency>
                <upper-frequency>196175000</upper-frequency>
                <target-power xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
nc:operation="create">-1.25</target-power>
              </config>
            </distribution>
          </spectrum-power-profile>
        </port>
      </port-spectrum-power-profiles>
    </wavelength-router>
</config>
</edit-config>
```

```
Request:
<get>
  <filter type="subtree">
    <wavelength-router xmlns="http://openconfig.net/yang/wavelength-router">
      <port-spectrum-power-profiles>
        <port>
          <name>Ots0/0/0/0</name>
          <spectrum-power-profile>
          </spectrum-power-profile>
        </port>
      </port-spectrum-power-profiles>
    </wavelength-router>
  </filter>
</get>
Response:
<?xml version="1.0"?>
<rpc-reply message-id="d78bc8c8-4e99-431b-9a8c-cab3d87e25cc"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <data>
  <wavelength-router xmlns="http://openconfig.net/yang/wavelength-router">
   <port-spectrum-power-profiles>
    <port>
     <name>Ots0/0/0/0</name>
     <spectrum-power-profile>
      <distribution>
       <lower-frequency>191200000</lower-frequency>
       <upper-frequency>196175000</upper-frequency>
```

```
       <state>
        <target-power>-1.20</target-power>
        <upper-frequency>196175000</upper-frequency>
        <lower-frequency>191200000</lower-frequency>
       </state>
       <config>
        <target-power>-1.25</target-power>
        <upper-frequency>196175000</upper-frequency>
        <lower-frequency>191200000</lower-frequency>
       </config>
      </distribution>
     </spectrum-power-profile>
    </port>
   </port-spectrum-power-profiles>
  </wavelength-router>
 </data>
</rpc-reply>
```

OpenConfig channel monitor model

This is a sample to get operational data using the OC-Channel monitor model:

```
{
  "openconfig-channel-monitor:channel-monitors": {
                "channel-monitor": [
                        {
                "name": "0/0-CHMON-RX-0",
                "channels":
                {
                    "channel":{}
                }
```

```
{
  "openconfig-channel-monitor:channel-monitors": {
                "channel-monitor": [
                        {
                "name": "0/0-CHMON-TX-0",
                "channels":
                {
                    "channel":{}
                }
```

This is a sample output to get data for single channel:

```
get data xpath:
/channel-monitors/channel-monitor/channels/channel[lower-frequency=191400000][upper-frequency=191406250]/state

Response:
[
  {
    "source": "10.127.126.174:57400",
    "timestamp": 1749368365802571706,
    "time": "2025-06-08T13:09:25.802571706+05:30",
    "updates": [
      {
        "Path":
"openconfig:channel-monitors/channel-monitor[name=0/0-CHMON-RX-0]/channels/channel[lower-frequency=191400000][upper-frequency=191406250]/state",

        "values": {
          "channel-monitors/channel-monitor/channels/channel/state": {
            "lower-frequency": "191400000",
            "upper-frequency": "191406250"
          }
        }
```

```
      },
      {
        "Path":
"openconfig:channel-monitors/channel-monitor[name=0/0-CHMON-TX-0]/channels/channel[lower-frequency=191400000][upper-frequency=191406250]/state",

        "values": {
          "channel-monitors/channel-monitor/channels/channel/state": {
            "lower-frequency": "191400000",
            "power": "-31.30",
            "upper-frequency": "191406250"
          }
        }
      },
      {
        "Path":
"openconfig:channel-monitors/channel-monitor[name=0/0-CHMON-RX-1]/channels/channel[lower-frequency=191400000][upper-frequency=191406250]/state",

        "values": {
          "channel-monitors/channel-monitor/channels/channel/state": {
            "lower-frequency": "191400000",
            "power": "0.00",
            "upper-frequency": "191406250"
          }
        }
      },
      {
        "Path":
"openconfig:channel-monitors/channel-monitor[name=0/0-CHMON-TX-1]/channels/channel[lower-frequency=191400000][upper-frequency=191406250]/state",

        "values": {
          "channel-monitors/channel-monitor/channels/channel/state": {
            "lower-frequency": "191400000",
            "power": "-26.60",
            "upper-frequency": "191406250"
          ]
```


OpenConfig transport line common model

This is a sample output for EDFA2 line interface:


```
{
    "source": "10.127.126.174:57400",
    "timestamp": 1749368046243415426,
    "time": "2025-06-08T13:04:06.243415426+05:30",
    "updates": [
      {
       "Path": "openconfig:components/component[name=Ots0/0/0/0]/port/optical-port/state",

        "values": {
          "components/component/port/optical-port/state": {
            "admin-state": "enabled",
            "input-power": {
              "instant": "-9.65"
                  }
      }
```

gNOI for OTDR

The OTDR gNOI protocol is designed to manage and monitor optical networks. This service is responsible for initiating and controlling OTDR scans, reporting scan results and handling errors during a scan based on the gnoi otdr proto definition.. In Release 25.1.1 OTDR gNOI supports only manual initiation of OTDR.

This table lists the responses received for the OTDR proto and the corresponding OTDR status shown in the CLI and the Cisco-IOS-XR-controller-ots-oper. yang model.

OTDR proto fields	**Corresponding OTDR status of OTDR in CLI and native yang model**
Error codes corresponding to OTDR initiation	
InitiateError_HARDWARE_FAILURE	OTDR_SCAN_STATUS_COMM_FAILED OTDR_SCAN_STATUS_ERROR
InitiateError_ALREADY_IN_PROGRESS	OPTICS_CERR_OTDR_SCAN_ALREADY_IN_PROGRESS
InitiateError_UNSPECIFIED	OPTICS_CERR_OTDR_SCAN_NOT_SUPPORTED OPTICS_CERR_OTDR_MODULE_NOT_DISCOVERED_OR_INVALID_ERRORS OPTICS_CERR_OTDR_PLUGGABLE_NOT_PRESENT OPTICS_CERR_OTDR_PATCH_CONNECTION_MISSING OPTICS_CERR_OTDR_LOCKED_BY_ANOTHER_ENTITY OTDR_SCAN_STATUS_STOPPED OTDR_SCAN_STATUS_ERROR OTDR_SCAN_STATUS_TIMEOUT OTDR_SCAN_STATUS_OTDR_LOCAL_RES_NOT_AVAILABLE OTDR_SCAN_STATUS_SPAN_RES_FAILED OTDR_SCAN_STATUS_SCAN_NOT_ALLOWED OTDR_SCAN_STATUS_SCAN_UNKNOWN
Status corresponding to OTDR progression	
InitiateProgress_PENDING	OTDR_SCAN_STATUS_WAITING_SPAN_RESERVATION
InitiateProgress_RUNNING	OTDR_SCAN_STATUS_MEASURING OTDR_SCAN_STATUS_DATA_PROCESSING
InitiateProgress_COMPLETE	OTDR_SCAN_STATUS_DATA_READY
OTDR measurements	
total_loss_db	not supported
total_length_m	not supported

OTDR proto fields	Corresponding OTDR status of OTDR in CLI and native yang model
optical_return_loss_db	optical_return_loss
local_path	Sor file
discovered_fiber_type	not supported
average_loss_db_km	not supported
Events	
distance_m	location
loss_db	In case of non-reflective event, loss_db is equivalent to magnitude
Reflection_db	In case reflective event, reflection_db is equivalent to magnitude