Configuring NCS 1002 Using Data Models

This section includes examples for configuring NCS 1002 using Data models.

Supported YANG Models in NCS 1002

The supported config and oper YANG models for NCS 1002 are listed below:

Cfg. yang	Oper. yang
Cisco-IOS-XR-pmengine-cfg.yang	Cisco-IOS-XR-pmengine-oper.yang
Cisco-IOS-XR-controller-optics-cfg.yang	Cisco-IOS-XR-controller-optics-oper.yang
Cisco-IOS-XR-controller-otu-cfg.yang	Cisco-IOS-XR-controller-otu-oper.yang
Cisco-IOS-XR-ncs1k-mxp-cfg	Cisco-IOS-XR-alarmgr-server-oper.yang
Cisco-IOS-XR-lib-keychain-macsec-cfg	Cisco-IOS-XR-ncs1k-mxp-headless-oper.yang
Cisco-IOS-XR-crypto-macsec-mka-cfg	Cisco-IOS-XR-plat-chas-invmgr-oper.yang
Cisco-IOS-XR-ifmgr-cfg	Cisco-IOS-XR-ncs1k-mxp-lldp-oper.yang Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang

Cfg. yang	Oper. yang
Cisco-IOS-XR-crypto-macsec-mka-if-cfg	Cisco-IOS-XR-crypto-macsec-mka-oper.yang Cisco-IOS-XR-crypto-macsec-secy-oper.yang

The supported versions of Open Config model are listed below:

- openconfig-platform.yang 0.4.0
- openconfig-platform-transceiver.yang 0.1.0
- openconfig-terminal-device.yang 0.3.0
- openconfig-interfaces.yang 1.0.2

Configure Slice

Step 1 Use the Cisco-IOS-XR-ncs1k-mxp-cfg.yang YANG model for provisioning the slice with traffic on the client and trunk ports.

All the five client ports of the slice need to be configured at the same bitrate except for mixed mode configuration. Both the trunk ports are always set with the same FEC mode. In mixed mode configuration, the client ports are configured at different bitrates.

YANG model	Example
Cisco-IOS-XR-ncs1k-mxp-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <hardware-module xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-cfg"> <node> <location>0_RP0_CPU0</location> <values> <value> <slice-id >3</slice-id> <client-rate>ten-gig</client-rate> <trunk-rate>two-hundred-gig</trunk-rate> <fec>sd20</fec> </value> <value> <slice-id >2</slice-id> <client-rate>ten-gig</client-rate> <trunk-rate>two-hundred-gig</trunk-rate> <fec>sd20</fec> </value> <value> <slice-id >1</slice-id> <client-rate>ten-gig</client-rate> <trunk-rate>two-hundred-gig</trunk-rate> <fec>sd20</fec> </value>

YANG model	Example
	<value> <slice-id >0</slice-id> <client-rate>ten-gig</client-rate> <trunk-rate>two-hundred-gig</trunk-rate> <fec>sd20</fec> </value> </values> </node> </hardware-module> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/2</interface-name> <optics xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-optics-cfg"> <optics-dwdm-carrier> <grid-type>50g-hz-grid</grid-type> <param-type>itu-ch</param-type> <param-value>1</param-value> </optics-dwdm-carrier> </optics> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

YANG model	Example
Cisco-IOS-XR-ncs1k-mxp-cfg.yang	<?xml version="1.0"?> <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <hardware-module xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-cfg"> <node> <location>0_RP0_CPU0</location> <slice> <values> <client-rate>ten-and-hundred-gig</client-rate> <trunk-rate>two-hundred-gig</trunk-rate> <fec>sd7</fec> </values> <slice-id>0</slice-id> </slice> </node> </hardware-module> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-ncs1k-mxp-oper.yang YANG model to verify the slice configuration.

YANG model	Example
Cisco-IOS-XR-ncs1k-mxp-oper.yang	<?xml version="1.0" ?> <rpc message-id="856612" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <hw-module xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-oper" > <slice-all> <slice-info> <slice-id>0</slice-id> </slice-info> </slice-all> <slice-all> <slice-info> <slice-id>1</slice-id> </slice-info> </slice-all> <slice-all> <slice-info> <slice-id>2</slice-id> </slice-info> </slice-all> <slice-all> <slice-info> <slice-id>3</slice-id> </slice-info> </slice-all> </hw-module> </filter> </get> </rpc>

Configure Optics Controller

Step 1 Use the Cisco-IOS-XR-ifmgr-cfg.yang YANG model for configuring the optics controller.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active>

YANG model	Example
	<interface-name>Optics0/0/0/5</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/6</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/12</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/13</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/19</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/20</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/26</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/27</interface-name> <shutdown></shutdown> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-controller-optics-cfg.yang YANG model for configuring the wavelength on the trunk port.

YANG model	Example
Cisco-IOS-XR-controller-optics-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>Optics0/0/0/2</interface-name> <optics xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-optics-cfg">

YANG model	Example
	<optics-dwdm-carrier> <grid-type>50g-hz-grid</grid-type> <param-type>itu-ch</param-type> <param-value>1</param-value> </optics-dwdm-carrier> </optics> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 3 Use the Cisco-IOS-XR-controller-optics-oper.yang YANG model to verify the wavelength and channel mapping for trunk optics controllers.

YANG model	Example
Cisco-IOS-XR-controller-optics-oper.yang	<?xml version="1.0" ?> <rpc message-id="8566" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter type="subtree"> <optics-oper xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-optics-oper"> <optics-ports> <optics-port> <name>Optics0/0/0/13</name> <optics-dwdm-carrrier-channel-map> </optics-dwdm-carrrier-channel-map> </optics-port> </optics-ports> </optics-oper> </filter> </get> </rpc>

Step 4 Use the Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang YANG model to display the name, status, and port description of the optics controller.

YANG model	Example
Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang	<?xml version="1.0" ?> <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <controllers xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper"> <controllers> <controller> <interafce-name>Optics0/0/0/12 </interafce-name> </controller> </controllers> </controllers> </filter>

YANG model	Example
	</get> </rpc>

Configure Ethernet and Coherent DSP Controllers

Step 1 Use the Cisco-IOS-XR-ifmgr-cfg.yang YANG model to configure the Ethernet controller.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>TenGigECtrlr0/0/0/0/1</interface-name> <shutdown xc:operation="delete" /> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-ifmgr-cfg.yang YANG model to configure the Coherent DSP controller.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>CoherentDSP0/0/0/6</interface-name> <shutdown xc:operation="delete" /> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>CoherentDSP0/0/0/13</interface-name> <shutdown></shutdown> </interface-configuration>

YANG model	Example
	<interface-configuration> <active>act</active> <interface-name>CoherentDSP0/0/0/20</interface-name> <shutdown></shutdown> </interface-configuration> <interface-configuration> <active>act</active> <interface-name>CoherentDSP0/0/0/27</interface-name> <shutdown></shutdown> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 3 Use the Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang YANG model to display the name, status, and port description of the Ethernet controller.

YANG model	Example
Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang	<?xml version="1.0" ?> <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <controllers xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper"> <controllers> <controller> <interafce-name>HundredGigECtrlr0/0/0/8 </interafce-name> </controller> </controllers> </controllers> </filter> </get> </rpc>

Step 4 Use the Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang YANG model to display the name, status, and port description of the Coherent DSP controller.

YANG model	Example
Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang	<?xml version="1.0" ?> Query: <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <controllers xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper"> <controllers> <controller> <interafce-name>CoherentDSP0/0/0/19 </interafce-name> </controller> </controllers> </controllers>

YANG model	Example
	</filter> </get> </rpc>

Configure Performance Monitoring

Step 1 Use the Cisco-IOS-XR-ifmgr-cfg.yang and Cisco-IOS-XR-pmengine-cfg.yang YANG models for configuring the performance monitoring parameters for the Optics, Ethernet, and coherentDSP controllers.

Step 2 Use the Cisco-IOS-XR-pmengine-oper.yang YANG models to view the performance monitoring parameters for the Optics, Ethernet, and coherentDSP controllers.

The table below shows an example that displays all the PM parameters for the optics controller. You can use specific filters for the required the output.

YANG model	Example
Cisco-IOS-XR-pmengine-oper.yang	<?xml version="1.0" ?> <rpc message-id="856612" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter type="subtree"> <performance-management xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-pmengine-oper"> <optics> <optics-ports> <optics-port>Optics0/0/0/1</optics-port> </optics-ports> </optics> </performance-management> </filter> </get> </rpc>

The table below shows an example that displays current 15 minute FEC PM for the Coherent DSP controller.

YANG model	Example
Cisco-IOS-XR-pmengine-oper.yang	<?xml version="1.0" ?> <rpc message-id="856612" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter type="subtree"> <performance-management xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-pmengine-oper"> <otu> <otu-ports> <otu-port> <name>CoherentDSP0/0/0/12</name> <otu-current> <otu-minute15> <otu-minute15fecs/> </otu-minute15>

YANG model	Example
	</otu-current> </otu-port> </otu-ports> </otu> </performance-management> </filter> </get> </rpc>

Configure Loopback

Step 1 Use the Cisco-IOS-XR-ifmgr-cfg.yang and Cisco-IOS-XR-controller-otu-cfg YANG models for configuring Loopback.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg.yang Cisco-IOS-XR-controller-otu-cfg.yang	<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>CoherentDSP0/1/0/0</interface-name> <otu xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-otu-cfg"> <otn-send-tti> <string-type>send-tti-full-ascii/full-ascii</string-type> <full-ascii-string>test1234</full-ascii-string> </otn-send-tti> <otn-expected-tti> <string-type>exp-tti-full-ascii/full-ascii</string-type> <full-ascii-string>test1234</full-ascii-string> </otn-expected-tti> </otu> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-ifmgr-cfg.yang and Cisco-IOS-XR-drivers-media-eth-cfg.yang YANG models for configuring the maintenance mode and loopback on an Ethernet controller.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg.yang Cisco-IOS-XR-drivers-media-eth-cfg.yang	<rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:34d98974-474a-4396-ad1a-6dd4ddfa20bc">

YANG model	Example
	<ok/> </rpc-reply>

Configure MACsec Encryption

Step 1 Use the Cisco-IOS-XR-ncs1k-mxp-cfg.yang YANG model to create an encrypted slice.

YANG model	Example
Cisco-IOS-XR-ncs1k-mxp-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <hardware-module xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-cfg"> <node> <location>0_RP0_CPU0</location> <values> <value> <slice-id>1</slice-id> <client-rate>hundred-gig</client-rate> <trunk-rate>two-hundred-gig</trunk-rate> <fec>sd20</fec> <encrypted>true</encrypted> </value> </values> </node> </hardware-module> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-lib-keychain-macsec-cfg.yang YANG model to configure the MACsec key chain.

YANG model	Example
Cisco-IOS-XR-lib-keychain-macsec-cfg.yang	<?xml version="1.0"?> <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config > <mac-sec-keychains xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-lib-keychain-macsec-cfg"> <mac-sec-keychain> <chain-name>keychain1</chain-name> <keies> <key> <key-id>kc1</key-id>

YANG model	Example
	<key-string> <string>055A575E701D1F58485446435A5D557B7A7579626473425647525007080902055F52494708090602030405 5A0A57560906554257550A5A575E701D1F5848544643</string> <cryptographic-algorithm>aes-256-cmac</cryptographic-algorithm> </key-string> <lifetime> <start-hour>10</start-hour> <start-minutes>10</start-minutes> <start-seconds>10</start-seconds> <start-date>1</start-date> <start-month>jan</start-month> <start-year>2016</start-year> <infinite-flag>true</infinite-flag> </lifetime> </key> <key> <key-id>kc2</key-id> <key-string> <string>0553515974181D5B485D40445E5857787A757A606177455 04E5253050D0D0503565B4F400C0C0401030406580F53510F0F5C4450510F58545E701E1D5D4C53404A</string> <cryptographic-algorithm>aes-256-cmac</cryptographic-algorithm> </key-string> <lifetime> <start-hour>10</start-hour> <start-minutes>10</start-minutes> <start-seconds>10</start-seconds> <start-date>13</start-date> <start-month>sep</start-month> <start-year>2016</start-year> <life-time>86400</life-time> </lifetime> </key> <key> <key-id>kc3</key-id> <key-string> <string>00554155500E5D5157701E1D5D4C53404A5A5E577E7E727F6B647040534355560E010F05015A504A47010F01060606065A0351510D035741575C0C5D535B721E1F</string> <cryptographic-algorithm>aes-256-cmac</cryptographic-algorithm> </key-string> <lifetime> <start-hour>10</start-hour> <start-minutes>10</start-minutes> <start-seconds>10</start-seconds> <start-date>25</start-date> <start-month>dec</start-month> <start-year>2016</start-year> <end-hour>10</end-hour> <end-minutes>10</end-minutes> <end-seconds>10</end-seconds> <end-date>1</end-date> <end-month>jan</end-month> <end-year>2017</end-year> </lifetime> </key> </keies> </mac-sec-keychain> </mac-sec-keychains> </config>

YANG model	Example
	</edit-config> </rpc>

Step 3 Use the Cisco-IOS-XR-crypto-macsec-mka-cfg.yang YANG model to configure a MACsec policy.

YANG model	Example
Cisco-IOS-XR-crypto-macsec-mka-cfg.yang	<?xml version="1.0"?> <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config > <macsec xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-macsec-mka-cfg"> <policy> <name>mac_policy</name> <key-server-priority>255</key-server-priority> <conf-offset>conf-off-set-0</conf-offset> <security-policy>must-secure</security-policy> <window-size>100</window-size> <cipher-suite>gcm-aes-xpn-256</cipher-suite> </policy> </macsec> </config> </edit-config> </rpc>

Step 4 Use the Cisco-IOS-XR-ifmgr-cfg.yang and Cisco-IOS-XR-crypto-macsec-mka-if-cfg.yang YANG model to configure MACsec on a MACsec controller.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg Cisco-IOS-XR-crypto-macsec-mka-if-cfg.yang	<?xml version="1.0"?> <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1. <edit-config> <target> <candidate/> </target> <config > <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr-cfg"> <interface-configuration> <active>act</active> <interface-name>MACSecCtrlr0/0/0/10</interface-name> <macsec xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-macsec-mka-if-cf <psk-key-chain> <key-chain-name>kc</key-chain-name> <policy-name>mac_policy</policy-name> </psk-key-chain> </macsec> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 5 Use the Cisco-IOS-XR-crypto-macsec-mka-oper.yang YANG model to verify the MACsec configuration and MKA session details of all the configured interfaces.

YANG model	Example
Cisco-IOS-XR-crypto-macsec-mka-oper.yang	<?xml version="1.0"?> rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <macsec xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-macsec-mka-oper" > <mka> </mka> </macsec> </filter> </get> </rpc>

Step 6 Use the Cisco-IOS-XR-crypto-macsec-secy-oper.yang YANG model to verify the MACsec SecY statistics for all the MACsec Key Agreement protocol (MKA) sessions.

YANG model	Example
Cisco-IOS-XR-crypto-macsec-secy-oper.yang	<?xml version="1.0"?> <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <macsec xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-crypto-macsec-secy-oper" > <secy> </secy> </macsec> </filter> </get> </rpc>

Configure Breakout Patch Panel

Step 1 Use the Cisco-IOS-XR-patch-panel-cfg.yang YANG model to configure the breakout patch panel.

YANG model	Example
Cisco-IOS-XR-patch-panel-cfg.yang	<?xml version="1.0"?> <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config type="subtree"> <patch-panel xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-patch-panel-cfg">

YANG model	Example
	<ipv4>169.254.1.4</ipv4> <user-name>SysAdmin</user-name> <password>!Password1</password> </patch-panel> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-patch-panel-cfg.yang YANG model to delete the breakout patch panel.

YANG model	Example
Cisco-IOS-XR-patch-panel-cfg.yang	<?xml version="1.0"?> <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101"> <edit-config> <target> <candidate/> </target> <config> <patch-panel xmlns:ns0="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-patch-panel-cfg" ns0:operation="delete"/> </config> </edit-config> </rpc>

Configure LLDP Drop

Step 1 Use the Cisco-IOS-XR-ncs1k-mxp-cfg.yang YANG model to configure LLDP drop.

YANG model	Example
Cisco-IOS-XR-ncs1k-mxp-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <hardware-module xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-cfg"> <node> <location>0_RP0_CPU0</location> <slice> <slice-id>0</slice-id> <lldp>true</lldp> </slice> </node> </hardware-module> </config> </edit-config> </rpc>

Step 2 Use the Cisco-IOS-XR-ncs1k-mxp-cfg.yang YANG model to delete LLDP drop configuration.

YANG model	Example
Cisco-IOS-XR-ncs1k-mxp-cfg.yang	<?xml version="1.0"?> <rpc message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <hardware-module xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-cfg"> <node> <location>0_RP0_CPU0</location> <slice> <slice-id>0</slice-id> <lldp>false</lldp> </slice> </node> </hardware-module> </config> </edit-config> </rpc>

Step 3 Use the Cisco-IOS-XR-ncs1k-mxp-cfg.yang YANG model to retrieve operational data for LLDP drop.

YANG model	Example
Cisco-IOS-XR-ncs1k-mxp-cfg.yang	<?xml version="1.0"?> <rpc message-id="856615" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <lldp-snoop-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-lldp-oper"/> </filter> </get> </rpc>

Configure PRBS on Trunk Ports

Step 1 Use Cisco-IOS-XR-ifmgr-cfg and Cisco-IOS-XR-controller-otu-cfg yang models to configure PRBS feature on trunk ports.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg.yang Cisco-IOS-XR-controller-otu-cfg.yang	<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target><candidate/></target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifmgr- xmlns:controller-otu-cfg="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-ot <interface-configuration> <active>act</active> <interface-name>CoherentDSP0/0/0/27</interface-name>

YANG model	Example
	<controller-otu-cfg:otu> <controller-otu-cfg:prbs> <controller-otu-cfg:mode-value>mode-source-sink</controller-otu-cfg:mode-v <controller-otu-cfg:patternvalue>pattern-pn31</controller-otu-cfg:patternv </controller-otu-cfg:prbs> <controller-otu-cfg:secondary-admin-state>maintenance</controller-otu-cfg:s </controller-otu-cfg:otu> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 2 Use Cisco-IOS-XR-ifmgr-cfg and Cisco-IOS-XR-controller-otu-cfg yang models to retrieve PRBS configuration on the trunk ports.

YANG model	Example
Cisco-IOS-XR-ifmgr-cfg.yang Cisco-IOS-XR-controller-otu-cfg.yang	<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target><candidate/></target> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <interface-configurations xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ifm xmlns:controller-otu-cfg="http://cisco.com/ns/yang/Cisco-IOS-XR-controller <interface-configuration> <active>act</active> <interface-name>CoherentDSP0/0/0/27</interface-name> <controller-otu-cfg:otu> <controller-otu-cfg:prbs> <controller-otu-cfg:mode-value>mode-source-sink</controller-otu-cfg:mode-v <controller-otu-cfg:patternvalue>pattern-pn31</controller-otu-cfg:patternv </controller-otu-cfg:prbs> <controller-otu-cfg:secondary-admin-state>maintenance</controller-otu-cfg:s </controller-otu-cfg:otu> </interface-configuration> </interface-configurations> </config> </edit-config> </rpc>

Step 3 Use Cisco-IOS-XR-controller-otu-oper yang model to retrieve PRBS status on the trunk ports.

YANG model	Example
Cisco-IOS-XR-controller-otu-oper.yang	<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get> <filter> <otu xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-controller-otu-oper"> <controllers> <controller> <prbs/> </controller> </controllers>

Monitor Headless Statistics

In the headless mode, the data path and statistics are maintained for at least 72 hours. The collected statistics are preserved for a slice until the hardware module configuration is removed or changed on that slice. These statistics are automatically cleared during the next reload or CPU-OIR operation.

Use the Cisco-IOS-XR-ncs1k-mxp-headless-oper YANG model for monitoring the headless statistics.

YANG model	Example

Cisco-IOS-XR-ncs1k-mxp-headless-oper	`<?xml version="1.0" ?>`
	`<rpc message-id="856615"`
	`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
	`<get>`
	`<filter>`
	`<headless-func-data`
	`xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ncs1k-mxp-headless-oper"`
	`/>`
	`</filter>`
	`</get>`
	`</rpc>`

Open Configuration Model for Client FEC and Laser-Squelch

Table 1: Feature History

Feature Name	Release	Description
OC (Open Configuration) Model for Client FEC and Laser Squelch	Cisco IOS XR Release 7.3.1	The OC model for configuring client FEC and Laser Squelch is available. This feature enables you to perform the configuration using scripts, which is less time-consuming. Also, the Open Configuration model supports the use of vendor-neutral data models to configure and manage the network.

Step 1 You can enable FEC (Forward Error Correction) on clients using the following scripts:

```
"openconfig-platform:components": {
 "component": [
  {
   "name": "0/0-Optics0/0/0/2",
   "config": {
    "name": "0/0-Optics0/0/0/2"
   },
```

```
   "openconfig-platform-transceiver:transceiver": {
    "config": {
     "fec-mode": "openconfig-platform-types:FEC_ENABLED"
    }
   }
  }
```

Step 2 You can get operational data using GNMI.

```
"state": {
    "connector-type": "openconfig-transport-types:LC_CONNECTOR",
    "date-code": "2019-08-05T00:00:00Z+00:00",
    "fault-condition": false,
    "fec-mode": "openconfig-platform-types:FEC_ENABLED",
    "fec-uncorrectable-words": 0,
    "form-factor": "openconfig-transport-types:QSFP28",
    "otn-compliance-code": "openconfig-transport-types:OTN_UNDEFINED",
    "present": "PRESENT",
    "serial-no": "INL23321878",
    "sonet-sdh-compliance-code": "openconfig-transport-types:SONET_UNDEFINED",
    "vendor": "CISCO-INNOLIGHT",
    "vendor-part": "10-3220-02",
    "vendor-rev": "1C"
}
```

IPv4 PING Over NETCONF

Use the Cisco-IOS-XR-ping-act YANG model to do the ping test to the destination IPv4 addresses.The following example shows the RPC request and RPC response messages for a successful ping test. The destination host is reachable and the success rate is 100%.

YANG Model	Example
Cisco-IOS-XR-ping-act.yang	<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:28170002-365f-45be-a8e1-e1f54d8b64b5"><ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act"> <destination> <destination>10.127.60.1</destination> </destination> </ping> </nc:rpc> <rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:28170002-365f-45be-a8e1-e1f54d8b64b5"> <ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act"> <ipv4>

YANG Model	Example
	<destination>10.127.60.1</destination> <data-size>100</data-size> <timeout>2</timeout> <pattern>abcd</pattern> <rotate-pattern>false</rotate-pattern> <replies> <reply> <reply-index>1</reply-index> <result>!</result> </reply> <reply> <reply-index>2</reply-index> <result>!</result> </reply> <reply> <reply-index>3</reply-index> <result>!</result> </reply> <reply> <reply-index>4</reply-index> <result>!</result> </reply> <reply> <reply-index>5</reply-index> <result>!</result> </reply> </replies> <hits>5</hits> <total>5</total> <success-rate>100</success-rate> <rtt-min>1</rtt-min> <rtt-avg>1</rtt-avg> <rtt-max>2</rtt-max>

YANG Model	Example
	</ipv4> </ping-response> </rpc-reply>

The following example shows the RPC request and RPC response messages for a failure ping test. The destination host is not reachable and the success rate is 0%.

YANG model	Example
Cisco-IOS-XR-ping-act.yang	<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:28170002-365f-45be-a8e1-e1f54d8b64b5"><ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act"> <destination> <destination>10.127.60.1</destination> </destination> </ping> </nc:rpc> <rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:02800209-6ebf-4955-8588-f6cdfd6f2750"> <ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act"> <ipv4> <destination>10.127.60.171</destination> <data-size>100</data-size> <timeout>2</timeout> <pattern>abcd</pattern> <rotate-pattern>false</rotate-pattern> <replies> <reply> <reply-index>1</reply-index> <result>.</result> </reply> <reply> <reply-index>2</reply-index> <result>.</result> </reply> <reply>

YANG model	Example
	<reply-index>3</reply-index> <result>.</result> </reply> <reply> <reply-index>4</reply-index> <result>.</result> </reply> <reply> <reply-index>5</reply-index> <result>.</result> </reply> </replies> <hits>0</hits> <total>5</total> <success-rate>0</success-rate> </ipv4> </ping-response> </rpc-reply>

IPv6 PING Over NETCONF

Table 2: Feature History

Feature Name	Release	Description
NETCONF Support for READ, WRITE, and Execute or Administrative Commands.	Cisco IOS XR Release 7.3.1	Support for IPv4 and IPv6 Ping test using the Cisco-IOS-XR-ping-act YANG model, instead of using CLI commands, is available. RPC (Remote Procedure Call) Request and Response messages are used to do the ping test, which is automated using scripts. This enables you to perform the ping test in a less time-consuming manner and to enhance network scalability.

Use the Cisco-IOS-XR-ping-act YANG model to do the ping test to the destination IPv6 addresses. The following example shows the RPC request and RPC response messages for a successful ping test. The destination host is reachable and the success rate is 100%.

YANG model	Example
Cisco-IOS-XR-ping-act.yang	<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:28170002-365f-45be-a8e1-e1f54d8b64b5"><ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act"> <destination> <destination>2001:420:5446:2014::281:178</destination> </destination> </ping> </nc:rpc> <rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:15798adc-f9f9-41b2-9aa5-a1c88dd788e8"> <ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act"> <ipv6> <destination>2001:420:5446:2014::281:178</destination> <repeat-count>50</repeat-count> <data-size>100</data-size> <timeout>2</timeout> <pattern>abcd</pattern> <rotate-pattern>false</rotate-pattern> <replies> <reply> <reply-index>1</reply-index> <result>!</result> </reply> <reply> <reply-index>2</reply-index> <result>!</result> </reply> <reply> <reply-index>3</reply-index> <result>!</result>

YANG model	Example
	</reply> <reply> <reply-index>4</reply-index> <result>!</result> </reply> <reply> <reply-index>5</reply-index> <result>!</result> </reply> </replies> <hits>5</hits> <total>5</total> <success-rate>100</success-rate> <rtt-min>1</rtt-min> <rtt-avg>1</rtt-avg> <rtt-max>2</rtt-max> </ipv6> </ping-response> </rpc-reply>

The following example shows the RPC request and RPC response messages for a failure ping test. The destination host is not reachable and the success rate is 0%.

YANG model	Example
Cisco-IOS-XR-ping-act.yang	<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:28170002-365f-45be-a8e1-e1f54d8b64b5"><ping xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act"> <destination> <destination>2001:420:5446:2014::281:178</destination> </destination> </ping> </nc:rpc> <rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:02800209-6ebf-4955-8588-f6cdfd6f2750"> <ping-response xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ping-act">

YANG model	Example
	<ipv6> <destination>2001:420:5446:2014::281:178</destination> <data-size>100</data-size> <timeout>2</timeout> <pattern>abcd</pattern> <replies> <reply> <reply-index>1</reply-index> <result>.</result> </reply> <reply> <reply-index>2</reply-index> <result>.</result> </reply> <reply> <reply-index>3</reply-index> <result>.</result> </reply> <reply> <reply-index>4</reply-index> <result>.</result> </reply> <reply> <reply-index>5</reply-index> <result>.</result> </reply> </replies> <hits>0</hits> <total>5</total> <success-rate>0</success-rate> </ipv6> </ping-response>

YANG model	Example
	</rpc-reply>

Examples Using gRPC

Example—Verify the Slice Configuration Using gRPC

Set-up:

- Client—client_v3
- Client IP address and configured grpc port—192.0.2.198:57500

```
./client_v3 -server 192.0.2.198:57500 -oper show-cmd-text -cli_input_file show-hw-module
```

The slice configuration is displayed.

```
{
  "Response": "{\"ResReqId\":753690684504425618,\"output\":\"\\n--------------------------
 show hw-module slice all -------------------------\\nSlice ID:              1\\nStatus:
                 Provisioned\\nClient Bitrate:          100\\nTrunk Bitrate:
100\\nDP FPGA Version:         H201 (NEED UPG)\\n\\nClient Port -   Trunk Port\\t
CoherentDSP0/0/0/12\\t CoherentDSP0/0/0/13\\nTraffic Split
Percentage\\n\\nHundredGigECtrlr0/0/0/7  \\t                100
0\\nHundredGigECtrlr0/0/0/11 \\t                  0                100\\n\\n\\n\"}",
  "FatalErrors": ""
}
```

Example—View the Optics Controller Configuration Using gRPC and Yang

Set-up:

- Client—client_v3
- Client IP address and configured grpc port—192.0.2.198:57500
- Yang model—Cisco-IOS-XR-ifmgr-cfg

```
./client -server_addr=192.0.2.198:57500 -username=root -password=lab -oper=get-config
-yang_path='{"Cisco-IOS-XR-ifmgr-cfg:interface-configurations": [null]}'
```

The optics controller configuration is displayed.

```
{
 "Cisco-IOS-XR-ifmgr-cfg:interface-configurations": {
  "interface-configuration": [
   {
    "active": "act",
    "interface-name": "Optics0/0/0/5",
    "shutdown": [null]
   },
   {
    "active": "act",
    "interface-name": "Optics0/0/0/6",
```

```
 "Cisco-IOS-XR-controller-optics-cfg:optics": {
  "optics-dwdm-carrier": {
   "grid-type": "100mhz-grid",
   "param-type": "frequency",
   "param-value": 1927000
  }
 },
 "secondary-admin-state": "maintenance"
},
{
 "active": "act",
 "interface-name": "Optics0/0/0/12",
 "shutdown": [
  null
 ]
},
{
 "active": "act",
 "interface-name": "Optics0/0/0/13",
 "Cisco-IOS-XR-controller-optics-cfg:optics": {
  "optics-dwdm-carrier": {
   "grid-type": "100mhz-grid",
   "param-type": "frequency",
   "param-value": 1927000
  }
 },
 "secondary-admin-state": "maintenance"
},
{
 "active": "act",
 "interface-name": "Optics0/0/0/14",
 "Cisco-IOS-XR-controller-optics-cfg:optics": {
  "rx-thresholds": {
   "rx-threshold": [
    {
     "rx-threshold-type": "low",
     "rx-threshold": -120
    },
    {
     "rx-threshold-type": "high",
     "rx-threshold": 49
    }]}}}
 ,
{
 "active": "act",
 "interface-name": "Optics0/0/0/18",
 "Cisco-IOS-XR-controller-optics-cfg:optics": {
  "rx-thresholds": {
   "rx-threshold": [
    {
     "rx-threshold-type": "low",
     "rx-threshold": -120
    },
    {
     "rx-threshold-type": "high",
     "rx-threshold": 49
    }]}}
   ,
{
 "active": "act",
 "interface-name": "Optics0/0/0/19",
 "shutdown": [
  null
 ],
```

```
 "Cisco-IOS-XR-controller-optics-cfg:optics": {
  "optics-dwdm-carrier": {
   "grid-type": "50g-hz-grid",
   "param-type": "frequency",
   "param-value": 19270
  }}}
,
{
 "active": "act",
 "interface-name": "Optics0/0/0/20",
 "Cisco-IOS-XR-controller-optics-cfg:optics": {
  "optics-dwdm-carrier": {
   "grid-type": "50g-hz-grid",
   "param-type": "frequency",
   "param-value": 19270
  },
  "rx-thresholds": {
   "rx-threshold": [
    {
     "rx-threshold-type": "low",
     "rx-threshold": -120
    },
    {
     "rx-threshold-type": "high",
     "rx-threshold": 49
    }]}}}
   ],
{
 "active": "act",
 "interface-name": "Optics0/0/0/26",
 "shutdown": [
  null
 ]
},
{
 "active": "act",
 "interface-name": "Optics0/0/0/27",
 "shutdown": [
  null
 ]
},
{
 "active": "act",
 "interface-name": "MgmtEth0/RP0/CPU0/0",
 "Cisco-IOS-XR-ipv4-io-cfg:ipv4-network": {
  "addresses": {
   "primary": {
    "address": "10.77.132.165",
    "netmask": "255.255.255.0"
   }}}}
  ,
{
 "active": "act",
 "interface-name": "TenGigECtrlr0/0/0/0/1",
 "Cisco-IOS-XR-pmengine-cfg:performance-management": {
  "ethernet-minute15": {
   "minute15-ether": {
    "minute15-ether-reports": {
     "minute15-ether-report": [
      {
       "ether-report": "report-fcs-err"
      }
     ]
    },
```

```
    "minute15-ether-thresholds": {
     "minute15-ether-threshold": [
      {
       "ether-threshold": "thresh-fcs-err",
       "ether-threshold-value": 1000
      }
     ]
    }
   }
  }
 }
},
{
 "active": "act",
 "interface-name": "TenGigECtrlr0/0/0/0/2",
 "Cisco-IOS-XR-pmengine-cfg:performance-management": {
  "ethernet-minute15": {
   "minute15-ether": {
    "minute15-ether-reports": {
     "minute15-ether-report": [
      {
       "ether-report": "report-fcs-err"
      }
     ]
    },
    "minute15-ether-thresholds": {
     "minute15-ether-threshold": [
      {
       "ether-threshold": "thresh-fcs-err",
       "ether-threshold-value": 1000
      }
     ]
    }
   }
  }
 }
},
{
 "active": "act",
 "interface-name": "TenGigECtrlr0/0/0/0/3",
 "Cisco-IOS-XR-pmengine-cfg:performance-management": {
  "ethernet-minute15": {
   "minute15-ether": {
    "minute15-ether-reports": {
     "minute15-ether-report": [
      {
       "ether-report": "report-fcs-err"
      }
     ]
    },
    "minute15-ether-thresholds": {
     "minute15-ether-threshold": [
      {
       "ether-threshold": "thresh-fcs-err",
       "ether-threshold-value": 1000
      }
     ]
    }
   }
  }
 }
},
{
 "active": "act",
```

```
 "interface-name": "TenGigECtrlr0/0/0/0/4",
 "Cisco-IOS-XR-pmengine-cfg:performance-management": {
  "ethernet-minute15": {
   "minute15-ether": {
    "minute15-ether-reports": {
     "minute15-ether-report": [
      {
       "ether-report": "report-fcs-err"
      }
     ]
    },
    "minute15-ether-thresholds": {
     "minute15-ether-threshold": [
      {
       "ether-threshold": "thresh-fcs-err",
       "ether-threshold-value": 1000
      }
     ]
    }
   }
  }
 }
},
{
 "active": "act",
 "interface-name": "TenGigECtrlr0/0/0/11/1",
 "Cisco-IOS-XR-pmengine-cfg:performance-management": {
  "ethernet-minute15": {
   "minute15-ether": {
    "minute15-ether-reports": {
     "minute15-ether-report": [
      {
       "ether-report": "report-fcs-err"
      }
     ]
    },
    "minute15-ether-thresholds": {
     "minute15-ether-threshold": [
      {
       "ether-threshold": "thresh-fcs-err",
       "ether-threshold-value": 1000
      }
     ]
    }
   }
  }
 }
},
{
 "active": "act",
 "interface-name": "TenGigECtrlr0/0/0/11/2",
 "Cisco-IOS-XR-pmengine-cfg:performance-management": {
  "ethernet-minute15": {
   "minute15-ether": {
    "minute15-ether-reports": {
     "minute15-ether-report": [
      {
       "ether-report": "report-fcs-err"
      }
     ]
    },
    "minute15-ether-thresholds": {
     "minute15-ether-threshold": [
      {
```

```
            "ether-threshold": "thresh-fcs-err",
            "ether-threshold-value": 1000
           }
          ]
         }
        }
       }
      }
     },
     {
      "active": "act",
      "interface-name": "TenGigECtrlr0/0/0/11/3",
      "Cisco-IOS-XR-pmengine-cfg:performance-management": {
       "ethernet-minute15": {
        "minute15-ether": {
         "minute15-ether-reports": {
          "minute15-ether-report": [
           {
            "ether-report": "report-fcs-err"
           }
          ]
         },
         "minute15-ether-thresholds": {
          "minute15-ether-threshold": [
           {
            "ether-threshold": "thresh-fcs-err",
            "ether-threshold-value": 1000
           }
          ]
         }
        }
       }
      }
     },
     {
      "active": "act",
      "interface-name": "TenGigECtrlr0/0/0/11/4",
      "Cisco-IOS-XR-pmengine-cfg:performance-management": {
       "ethernet-minute15": {
        "minute15-ether": {
         "minute15-ether-reports": {
          "minute15-ether-report": [
           {
            "ether-report": "report-fcs-err"
           }
          ]
         },
         "minute15-ether-thresholds": {
          "minute15-ether-threshold": [
           {
            "ether-threshold": "thresh-fcs-err",
            "ether-threshold-value": 1000
           }
          ]
         }
        }
       }
      }
     }
    ]
   }
}
emsGetConfig: ReqId 1, byteRecv: 7455
```

```
----------------- gRPC Summary ----------------------

Operation: get-config
Number of iterations: 1
Total bytes transferred: 7455
Number of bytes per second: 124482
Ave elapsed time in seconds: 0.059888
Min elapsed time in seconds: 0.059888
Max elapsed time in seconds: 0.059888

--------------- End gRPC Summary ----------------------
```