

Configure Software for Cisco NCS 1002 (Rosco) with Alarm and Performance Data Monitoring

Contents

- [Introduction](#)
- [Prerequisites](#)
- [Requirements](#)
- [Components Used](#)
- [Background Information](#)
- [Configure](#)
- [Set and Verify Management IP](#)
- [Verify Software Version](#)
- [Verify Status of Hardware](#)
- [Configure and Verify Slices](#)
- [SNMP](#)
- [Telemetry](#)
- [Netconf and SSH](#)
- [Verify Optical PM \(Client QSFP and Trunk CFP\)](#)
- [Verify Ethernet PM on Client Ports](#)
- [Verify Alarms \(Current and History\) on Port and System](#)
- [Verify](#)
- [Troubleshoot](#)
- [Related Information](#)

Introduction

This document describes the basic software configuration steps along with alarm and performance data monitoring for Cisco Network Convergence System (NCS)1002 (Rosco).

Prerequisites

Requirements

Cisco recommends you have knowledge of these topics:

- NCS1002
- Cisco IOS®-XR Platform specific system with optical products knowledge

Components Used

The information in this document is based on these software and hardware versions:

- NCS1002
- Cisco IOS®-XR VM console login

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

The NCS1002 is a 2 Rollup (RU) system that delivers fully programmable, High-bandwidth capacity (up to 250 Gbps) wavelengths over distances that exceeds 3000 km with the use of current fiber. Powered by the industry-leading Cisco IOS® XR operating system, Cisco NCS 1002 offers robust functions such as third party application hosting, Machine-to-Machine (M2M) interface, telemetry and flexible package delivery.

NCS 1002 delivers these benefits:

- Supports up to 2 Tbps capacity
- Transports 100, 200, or 250 Gbps per wavelength on the same platform through software provisioning
- Transports 10 GE and 100 GE on the same platform through software provisioning
- Supports grid-less tuning for flex-grid Dense Wavelength-Division Multiplexing (DWDM)
- Supports different modulation formats (PM-QPSK or PM-16QAM)
- Supports 7% or 20% Soft Decision (SD) FEC for maximum optical performance
- Allows for automated installation, configuration and monitoring
- Supports M2M APIs based on Yet Another Next Generation (YANG) models for ease of configuration
- Supports a telemetry agent for a pub-sub model of device monitoring

Configure

Set and Verify Management IP

Before you begin:

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management port
- Ensure that the management port is connected to management network

```
RP/0/RP0/CPU0:DBX2#conf t
```

```
Thu Feb 11 07:45:28.810 UTC
```

```
RP/0/RP0/CPU0:DBX2(config)#interface mgmtEth 0/RP0/CPU0/0
```

```
RP/0/RP0/CPU0:DBX2(config-if)#ipv4 address 172.20.165.151/24
```

```
RP/0/RP0/CPU0:DBX2(config-if)#no shutdown
```

```
RP/0/RP0/CPU0:DBX2(config-if)#commit
```

```
RP/0/RP0/CPU0:DBX2#sh ipv4 interface brief
```

```
Thu Feb 11 07:44:21.811 UTC
```

Interface	IP-Address	Status	Protocol	Vrf-Name
MgmtEth0/RP0/CPU0/0	172.20.165.151 Up	Up	default	

```
RP/0/RP0/CPU0:DBX2#
```

Verify Software Version

```
RP/0/RP0/CPU0:DBX2#show version
```

```
Thu Feb 11 07:52:26.846 UTC
```

```
Cisco IOS XR Software, Version 6.0.0
```

```
Copyright (c) 2013-2015 by Cisco Systems, Inc.
```

```
Build Information:
```

```
Built By      : alnguyen
```

```
Built On     : Thu Dec 24 01:05:17 PST 2015
```

```
Build Host   : iox-lnx-005
```

```
Workspace    : /auto/srcarchive16/production/6.0.0/ncs1k/workspace
```

```
Version      : 6.0.0
```

```
Location     : /opt/cisco/XR/packages/
```

```
cisco NCS1002 () processor
```

```
System uptime is 21 hours, 2 minutes
```

```
RP/0/RP0/CPU0:DBX2#
```

Verify Status of Hardware

```
RP/0/RP0/CPU0:DBX2#show platform
```

```
Thu Feb 11 10:06:43.448 UTC
```

Node name	Node type	Node state	Admin state	Config state
-----------	-----------	------------	-------------	--------------

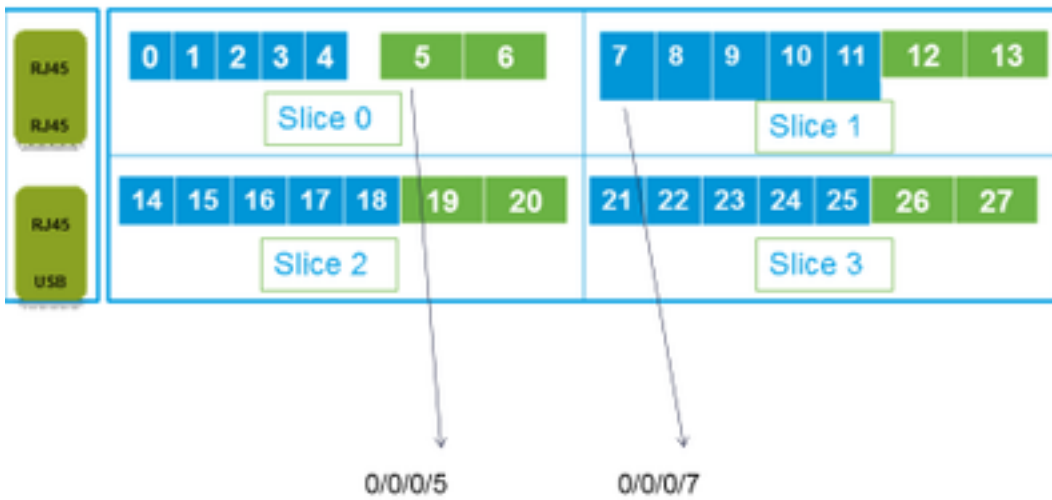
```
-----  
0/RP0          NCS1K-CNTRLR      OPERATIONAL      UP              NSHUT
```

```
RP/0/RP0/CPU0:DBX2#
```

Configure and Verify Slices

Port representation in NCS1002 is as shown in the image.

Port addressing on NCS1K



```
RP/0/RP0/CPU0:DBX2#conf t
```

```
Thu Feb 11 08:53:44.390 UTC
```

```
RP/0/RP0/CPU0:DBX2(config)#hw-module location 0/RP0/CPU0 slice 3 client bitrate 100 trunk
bitrate 200$
```

```
RP/0/RP0/CPU0:DBX2(config)#commit
```

```
Thu Feb 11 08:54:16.383 UTC
```

```
RP/0/RP0/CPU0:DBX2(config)#end
```

```
RP/0/RP0/CPU0:DBX2#show hw-module slice 3
```

```
Thu Feb 11 08:55:05.100 UTC
```

```
Slice ID:          3
Status:           Provisioning In Progress
Client Bitrate:   100
Trunk Bitrate:    200
DP FPGA Version:  H14 (CURRENT)
```

Client Port - Trunk Port	CoherentDSP0/0/0/26	CoherentDSP0/0/0/27
Traffic Split Percentage		
HundredGigEctrlr0/0/0/21	100	0
HundredGigEctrlr0/0/0/22	100	0
HundredGigEctrlr0/0/0/24	0	100
HundredGigEctrlr0/0/0/25	0	100

RP/0/RP0/CPU0:DBX2#

The Traffic Combinations supported on NCS1k:

5 x 40G à 2 x 100G
5 x 40G à 1 x 200G
5 x 40G à 1 x 250G
20 x 10G à 2 x 100G
20 x 10G à 1 x 200G
20 x 10G à 1 x 250G
2 x 100G à 2 x 100G
4 x 100G à 2 x 200G
5 x 100G à 2 x 250G

SNMP

- Configuring SNMP V2C on Cisco IOS®-XR

The configuration you need in order to enable SNMP V2c get/set requests:

```
RP/0/0/CPU0:smart-prpl6#conf t
RP/0/0/CPU0:smart-prpl6(config)#
RP/0/0/CPU0:smart-prpl6(config)#snmp-server community public RW SystemOwner
RP/0/0/CPU0:smart-prpl6(config)#commit
RP/0/0/CPU0:smart-prpl6#
RP/0/0/CPU0:smart-prpl6#show running-config snmp-server
snmp-server community public RW SystemOwner
```

Explanation of Configuration:

In configuration "public" is the community string and it be any text.

The Options RW and SystemOwner provides Read/Write to the entire system, including admin plane. If you need access to just the SDR (secure Domain Router),

then you can choose option "SDROwner"

You can also apply an Access list to the SNMP server. You can use command.

```
#snmp-server community public RW SDROwner my_acl_ravi
```

Where my_acl_ravi is an ACL.

Use command to verify the SNMP configuration.

```
RP/0/RP0/CPU0:DBX2#show snmp group
```

Thu Feb 11 09:09:48.303 UTC

```
groupname: public                security model:snmpv1
readview : vldefault            writeview:
```

```
notifyview: vldefault
row status: active
groupname: public                security model:snmpv2c
readview : vldefault            writeview:
notifyview: vldefault
row status: active
```

- **Configure SNMP V3 on Cisco IOS®-XR:**

Follow the steps:

SNMPV3

Configure an SNMP View

Command Syntax:

```
snmp-server view view-name oid-tree included
```

Where

view-name: is the name of the View
oid-tree: Object identifier (OID) of the ASN.1 subtree to be included or excluded from the view. To identify the subtree, specify a text string consisting of numbers, such as 1.3.6.2.4, or a word, such as system. Replace a single sub-identifier with the asterisk wildcard to specify a subtree family; for example 1.3.*.4.

```
RP/0/RP1/CPU0:akki(config)#snmp-server view view1 1.3 included
RP/0/RP1/CPU0:akki(config)#commit
```

```
RP/0/RP1/CPU0:akki#show snmp view
view1 org - included nonVolatile active
vldefault iso - included nonVolatile active
RP/0/RP1/CPU0:akki#
```

- **Configure an SNMP Group:**

```
RP/0/RP1/CPU0:akki(config)#snmp-server group group1 v3 priv write view1 read view1
RP/0/RP1/CPU0:akki(config)#commit
RP/0/RP1/CPU0:akki#show running-config snmp-server group snmp-server group group1 v3 priv read
view1 write view1
```

```
RP/0/RP1/CPU0:akki#show snmp group groupname: group1 security model:usm readview : view1
writeview: view1 notifyview: vldefault row status: nonVolatile
```

```
RP/0/RP1/CPU0:akki#
```

- **Configure an SNMP User:**

```
RP/0/RP1/CPU0:akki(config)#snmp-server user user1 group1 v3 auth md5 clear lab priv des56 clear
lab SystemOwner
RP/0/RP1/CPU0:akki(config)#commit
RP/0/RP1/CPU0:akki(config)#
RP/0/RP1/CPU0:akki#
```

```
RP/0/RP1/CPU0:akki#show snmp users
User name: user1
Engine ID: localSnmpID
storage-type: nonvolatile active
```

```
RP/0/RP1/CPU0:akki#show running-config snmp-server user
snmp-server user user1 group1 v3 auth md5 encrypted 13091610 priv des56 encrypted 09404F0B
SystemOwner
```

```
RP/0/RP1/CPU0:akki#
```

So far we need lines for V3 SNMP

```
RP/0/RP1/CPU0:akki#show running-config snmp-server
snmp-server user user1 group1 v3 auth md5 encrypted 13091610 priv des56 encrypted 09404F0B
SystemOwner
snmp-server view view1 1.3 included
snmp-server group group1 v3 priv read view1 write view1
```

With this you have setup the configuration on the router for V3 Requests.

Refer [Setting Up Host for V3 SNMP Requests](#).

Telemetry

Streaming telemetry lets the user's direct data to a configured receiver for analysis and troubleshooting purposes in order to maintain the health of the network. This is achieved by leveraging the capabilities of M2M communication.

Traditionally, organizations used the pull model to collect data, where a client pulls data from network elements. This pull model, however, does not scale when there is more than one network management station in the network. These traditional techniques do not cater to all the underlying information of the router, and they require manual intervention.

Follow the link:

https://www.cisco.com/c/en/us/td/docs/iosxr/Telemetry/Telemetry-Config-Guide/Telemetry-Config-Guide_chapter_011.html

Netconf and SSH

Netconf provides mechanisms to install, manipulate, and delete the configuration of network devices.

The Netconf protocol provides a set of operations to manage device configurations and retrieve device state information.

- Verify the installation of k9sec package
- Generate the crypto key for Secure Shell (SSH) with the use of the **crypto key generate dsa** command
- Configure SSH

```
RP/0/RP0/CPU0:ios# configure
```

```
RP/0/RP0/CPU0(config)# ssh server v2
```

```
RP/0/RP0/CPU0(config)# ssh server netconf port 830
```

```
RP/0/RP0/CPU0(config)# ssh server netconf vrf default
```

Note: Port 830 is the default Netconf port.

- **Configure Netconf:**

```
RP/0/RP0/CPU0:ios# configure
```

```
RP/0/RP0/CPU0(config)# netconf-yang agent ssh
```

- Display the client details for netconf-yang, run the **show netconf-yang clients** command in EXEC mode.

```
RP/0/RP0/CPU0:ios# show netconf-yang clients
```

```
Tue Dec 8 07:49:14.846 UTC
```

```
Netconf clients
```

```
client session ID| NC version| client connect time| last OP time| last OP type|
```

```
<lock>|
```

```
1188487019| 1.1| 0d 16h 56m 50s| 01:17:13| get|
```

```
No|
```

- Display the statistical details for netconf-yang, run the **show netconf-yang statistics** command in EXEC mode.

```
RP/0/RP0/CPU0:ios# show netconf-yang statistics
```

```
Tue Dec 8 07:49:45.506 UTC
```

```
Summary statistics
```

```
# requests| total time| min time per request| max time per request| avg time per request|other  
0| 0h 0m 0s 0ms| 0h 0m 0s 0ms| 0h 0m
```

```
0s 0ms| 0h 0m 0s 0ms|
```

```
0s 0ms| 0h 0m 0s 0ms|
```

```
Statistics for session with ID: 1188487019
```

```
<snip>
```

- Debug and verify Netconf, run the **show netconf-yang trace** command in EXEC mode.

```
RP/0/RP0/CPU0:ios# show netconf-yang trace
```

```
Tue Dec 8 07:50:54.590 UTC
```

```
[12/08/15 07:30:37.851 UTC 1046d3 4942] TRC: nc_sm_session_find_session_id:1386 Found session  
3027026318 0x1852f68
```


[12/08/15 07:30:37.851 UTC 1046d4 4942] DBG: nc_sm_yfw_response_cb:2816 Received OK response for session-id '3027026318', for message-id '856615', which has 'NO ERROR' and 'DATA'

[12/08/15 07:30:37.851 UTC 1046d5 4942] TRC: nc_sm_yfw_response_complete:2700 DATA element in chunk

state: CONTINUE

<snip>

Verify Optical PM (Client QSFP and Trunk CFP)

- PM for client ports:

For current PM data

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm current 15-min/24-hour optics 1
```

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm current 15-min/24-hour optics 2
```

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm current 15-min/24-hour optics 3
```

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm current 15-min/24-hour optics 4
```

For History PM data

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm history 15-min/24-hour optics 1 bucket <1-32/1-12>
```

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm history 15-min/24-hour optics 2 bucket <1-32/1-12>
```

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm history 15-min/24-hour optics 3 bucket <1-32/1-12>
```

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/0 pm history 15-min/24-hour optics 4 bucket <1-32/1-12>
```

- PM for Trunk CFP ports:

For current PM data:

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/5 pm current 15-min/24-hour optics 1
```

For history PM data:

```
RP/0/RP0/CPU0:DBX2#show controllers optics 0/0/0/5 pm history 15-min/24-hour optics 1 bucket 1
```

Verify Ethernet PM on Client Ports

- Client QSFP ports:

```
RP/0/RP0/CPU0:DBX2#show controllers hundredGigECtrlr 0/0/0/0 pm current 15-min/24-hour ether
```

```
RP/0/RP0/CPU0:DBX2#show controllers hundredGigECtrlr 0/0/0/0 pm history 15-min ether <1-32>
```

```
RP/0/RP0/CPU0:DBX2#show controllers hundredGigECtrlr 0/0/0/0 pm history 24-hour ether
```

- Trunk CFP ports:

```
RP/0/RP0/CPU0:DBX2#show controllers coherentDSP 0/0/0/5 pm current 15-min otn
RP/0/RP0/CPU0:DBX2#show controllers coherentDSP 0/0/0/5 pm current 15-min fec
RP/0/RP0/CPU0:DBX2#show controllers coherentDSP 0/0/0/5 pm history 15-min fec <1-32>
RP/0/RP0/CPU0:DBX2#show controllers coherentDSP 0/0/0/5 pm history 24-hour fec
```

Verify Alarms (Current and History) on Port and System

- For current alarms run the command:

```
RP/0/RP0/CPU0:DBX2#show alarms brief card location 0/RP0/CPU0 active
Thu Feb 11 10:12:21.886 UTC
```

Active Alarms

Location	Severity	Group	Set Time	Description
0/0 Removal	Critical	Controller	02/10/2016 10:51:33	Optics0/0/0/10 - Improper
0/0 Removal	Critical	Controller	02/10/2016 10:51:33	Optics0/0/0/11 - Improper

[snippet]

- For history alarms run the command:

```
RP/0/RP0/CPU0:DBX2#show alarms brief card location 0/RP0/CPU0 history
Thu Feb 11 10:14:13.070 UTC
```

History Alarms

Location	Severity	Group	Set Time	Description
0/0 Differential Group Delay	Minor	Controller	02/10/2016 10:51:33	Optics0/0/0/5 - Optics High

02/10/2016 10:52:01

0/0 Minor Controller 02/10/2016 10:51:33 Optics0/0/0/5 - Optics Out of Range Chromatic Dispersion

02/10/2016 10:52:01

[snippet]

Note: Output has been captured from release 6.0.0 for NSC1002.

Verify

Use this section in order to confirm that your configuration works properly.

The verification process is covered individually in the Configure section of this article.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

Related Information

- http://www.cisco.com/c/en/us/td/docs/optical/ncs1000/dwdm-system-setup-guide/b-system-setup-ncs1002_chapter_010.html#task_37FE9449C1004631B8CEB859BB191F9E
- http://www.cisco.com/c/en/us/td/docs/optical/ncs1000/dwdm-cli-reference/ncs1002commandreference_chapter_00.html
- [Technical Support & Documentation - Cisco Systems](#)