Validate Routed PON Deployment 24.1.2

Contents

Introduction

Prerequisites

- Requirements

Components Used

Validation Steps - XR Router

- Configuration Checks
  - Linux Networking
  - Physical and Sub-interface Configuration
  - PON-Controller Configuration
- Verifications
  - JSON File
  - Connectivity Checks
  - Container Status
  - Date and Time
  - Trace Messages

Validation Steps - Linux VM

- Verifications
  - status.sh Script
  - Netplan
  - IP Connectivity
  - MongoDB Configuration File
  - System Level Log Locations

Reference Documentation

Introduction

This document describes validation of the Cisco Routed PON (Passive Optical Network) Solution on a Virtual Machine (VM) and XR Router.

Prerequisites

Requirements

Cisco recommends knowledge on these topics.

- Cisco IOS® XR Software
- Linux
- Virtual Machine Environment

Components Used

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

- NCS-540-28Z4C-SYS-A XR Router
- Cisco IOS® XR Software 24.1.2
- Routed PON Version 24.1.2
- Ubuntu Version 20.04.06 LTS

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.

Validation Steps - XR Router

Configuration Checks

Linux Networking

Ensure that the Virtual Routing and Forwarding (VRF) used for connectivity, is reflected within the linux networking configuration. For this example, VRF Mgmt-intf has been configured. Additionally, ensure that the source-hint default-route is set for the correct uplink interface. Connectivity in the listed example uses the interface MgmtEth0/RP0/CPU0/0.

Configuration Example:

```
linux networking
 vrf Mgmt-intf
  address-family ipv4
   default-route software-forwarding
    source-hint default-route interface MgmtEth0/RP0/CPU0/0
```

Physical and Sub-interface Configuration

Ensure that the interface the Optical Line Terminal (OLT) Pluggable is inserted into is correct and not shutdown in configuration. Additionally, confirm that the sub-interface is dot1q tagged with 4090 and is applied to the associated physical interface.

Configuration example:

```
interface TenGigE0/0/0/0
description PON OLT
!
interface TenGigE0/0/0/0.4090
encapsulation dot1q 4090
```

Command verification:

```
<#root>
RP/0/RP0/CPU0:F340.16.19.N540-1#
show ip interface brief

Tue Jul 16 15:08:28.786 UTC
Interface            IP-Address  Status Protocol Vrf-Name
TenGigE0/0/0/0
      unassigned
Up


Up
       default
TenGigE0/0/0/0.4090
  unassigned
Up


Up
       default
RP/0/RP0/CPU0:F340.16.19.N540-1#
show interface TenGigE0/0/0/0.4090

Wed Jul 17 13:17:07.754 UTC
TenGigE0/0/0/0.4090 is up, line protocol is up

Interface state transitions: 5
Hardware is VLAN sub-interface(s), address is c47e.e0b3.9b04
Internet address is Unknown
MTU 1518 bytes, BW 10000000 Kbit (Max: 10000000 Kbit)
reliability 255/255, txload 0/255, rxload 0/255
Encapsulation 802.1Q Virtual LAN, VLAN Id 4090
, loopback not set
```

Ensure LLDP is enabled in global configuration.

```
<#root>
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

```
show run | include lldp

Thu Jul 18 20:16:12.073 UTC
lldp
```

PON-Controller Configuration

Ensure that the xr-pon-ctrl RPM is installed and is an active. If not, confirm the **NCS540l-iosxr-optional-RPMs-24.2.11.tar** exists on the harddisk (in the Linux shell, the path is /misc/disk1/), and the local-repo containing the software matched RPMs is referenced correctly.



Note: Information on the installation and management on system wide RPMs can be found at this link: [System Setup and Software Installation Guide for Cisco NCS 540 Series Routers, IOS XR Release 24.1.x, 24.2.x]

Example:

```
<#root>

RP/0/RP0/CPU0:F340.16.19.N540-2#

show install active summary | include xr-pon

Tue Jul 16 14:59:16.082 UTC

xr-pon-ctlr 24.1.2v1.0.0-1
```

```
<#root>

install
 repository local-repo
  url file:///

harddisk:/optional-RPMs-2412
```

Ensure that the PON-Controller is configured with the correctly associated file, file path and VRF.

Example:

```
<#root>

pon-ctlr
 cfg-file
```

harddisk:/PonCntlInit.json vrf Mgmt-intf

Verifications

JSON File



Note: The **PonCntlInit.json** file example is included with the installation of **Routed PON Manager** software on the VM.



Note: With a single VM installation of PON Manager, the MongoDB IP and the VM IP are one in the same.



Note: The listed example does NOT use TLS. If you are using TLS, ensure that the **username** and **password** are set correctly for your installation.

Ensure that the IP of the MongoDB is set in the **host:** section to match what the PON controller connects to. Additionally, confirm the configured port matches that of the **mongod.conf** file in the VM.

Example:

<#root>

```
{
   "CNTL": {
       "Auth": false,
       "CFG Version": "R4.0.0",

"DHCPv4": true, <- DHCP set to true for CPE devices, Default is false.

       "DHCPv6":

true

,

<- DHCP set to true for CPE devices, Default is false.

       "PPPoE": false,
       "UMT interface": "tibitvirt",

"

Maximum CPEs Allowed": 0,
       "Maximum CPE Time": 0
   },
```

```
   "DEBUG": {},
   "JSON": {
       "databaseDir": "/opt/tibit/poncntl/database/",
       "defaultDir": "/opt/tibit/poncntl/database/"
   },
   "Local Copy": {
       "CNTL-STATE": false,
       "OLT-STATE": false,
       "ONU-STATE": false
   },
   "Logging": {
       "Directory": "/var/log/tibit",
       "FileCount": 3,
       "FileSize": 10240000,
       "Tracebacks": false,
       "Timestamp": false,
       "Facility" : "user"
   },
   "MongoDB": {
       "auth_db": "tibit_users",
       "auth_enable": false,
       "ca_cert_path": "/etc/cisco/ca.pem",
       "compression": false,
       "write_concern": "default",

"host": "10.122.140.232", <- MongoDB IP

       "name": "tibit_pon_controller",

"password": "", <- Left Empty - Not using TLS

       "port": "27017", <- MongoDB TCP Port

       "tls_enable": false, <- Set to False to leave TLS disabled

       "username": "", <- Left Empty - Not using TLS

       "dns_srv": false,
       "db_uri": "",
       "replica_set_enable": false,
       "validate_cfg": true
   },
   "databaseType": "MongoDB",
   "interface": "veth_pon_glb"
}
```

Connectivity Checks

From the XR router, ping the **MongoDB/VM Hosting Routed PON Manager**. If you are using a VRF, source from the VRF.

Example:

```
<#root>

RP/0/RP0/CPU0:F340.16.19.N540-1#

ping vrf Mgmt-intf 10.122.140.232


Tue Jul 16 15:09:52.780 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.122.140.232 timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

Container Status

The PON Controller runs on a docker container on the XR router. Check the status of the container by logging into the linux shell in the XR router, then run the command **docker ps.** This shows the currently up and active container if there is one.

Example:

```
<#root>

RP/0/RP0/CPU0:F340.16.19.N540-1#

run


Tue Jul 16 15:14:26.059 UTC
[node0_RP0_CPU0:~]$docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED     STATUS PORTS   NAMES

2e700f202ee3

   tibit-poncntl.xr:R4.0.0 "/usr/bin/supervisor…" 3 days ago Up     3 days

pon_ctlr
```

If the docker container is NOT running, check the contents and file structure on the JSON file. Check logs of the docker container for any active errors. The log example shows an ONU registering with the controller. This also prints any docker level errors in regards to the container and OLT. Additionally, guidance can be gained from running a simple **show logging** to check for error messages.



Note: The usage of **--follow** displays the latest log content within docker.

Example:

<#root>

```
[node0_RP0_CPU0:~]$
```

docker logs pon_ctlr

```
2024-07-16 15:05:11.630 PonCntl System Status
{
"e0:9b:27:36:aa:76": {
"OLT State": "Primary",
"ONU Active Count": 1,
"ONUs": {
"CIGG2410503f": "Registered"
```

Date and Time

Ensure that the time and date on the XR Router and the VM hosting Routed PON Manager match. If possible, use the same NTP servers for optimal accuracy.

⚠ **Caution**: NTP being out of sync between the VM and XR Router directly impacts OLT visibility in Routed PON Manager.

Example:

<#root>

```
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

show clock

```
Tue Jul 16 15:25:03.781 UTC
15:25:03.827 UTC Tue Jul 16 2024
```

Configuration Example:

```
ntp
 server vrf Mgmt-intf 172.18.108.14 source MgmtEth0/RP0/CPU0/0
 server vrf Mgmt-intf 172.18.108.15 prefer source MgmtEth0/RP0/CPU0/0
```

Trace Messages

The PON process generates additional logging through ltrace. Check these logs for any errors related to this process.

Example:

<#root>

```
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

show pon-ctlr ltrace all reverse location all

```
Wed Jul 17 13:25:43.747 UTC
670 wrapping entries (4224 possible, 896 allocated, 0 filtered, 670 total)
Jul 10 19:17:55.066 pon_ctlr/event 0/RP0/CPU0 t6986 pon_ctlr_config_sysdb.c:117:Successfully connected
Jul 10 19:17:55.039 pon_ctlr/event 0/RP0/CPU0 t6986 pon_ctlr_main.c:372:Succeessfully registered with i
Jul 10 19:17:55.006 pon_ctlr/event 0/RP0/CPU0 t7082 pon_ctlr_utls.c:353:IP LINK: ip link delete veth_po
```

Validation Steps - Linux VM

Verifications

status.sh Script

Within the Routed PON Manager installation directory, there is a shell script (**status.sh**) to display the current status of each associated process. Run this script with elevated privilege to verify each of the listed services is up and running. In the event that one of the services is not running, first check the installation script that was ran when performing the install and ensure the proper arguments were set per the installation guide.

Note: The Cisco Routed PON Manager Installation Guide can be found at this link: Cisco Routed PON Manager Installation Guide

---

mongod.service
apache2.service
netconf.service
netopeer2-server.service

Example:

<#root>

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CCO/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

sudo ./status.sh

```
[sudo] password for rpon:
MCMS Component Versions:
PON Manager: R4.0.0
```

PON NETCONF: R4.0.0
PON Controller: Not Installed

•

mongod.service

- MongoDB Database Server
Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)

Active: active (running) since Thu 2024-06-27 08:46:25 EDT; 2 weeks 5 days ago

Main PID: 52484 (mongod)
Memory: 1.5G
CGroup: /system.slice/mongod.service
└─52484 /usr/bin/mongod --config /etc/mongod.conf

•

apache2.service

- The Apache HTTP Server
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2024-07-12 06:33:30 EDT; 4 days ago

Process: 103015 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
Main PID: 96525 (apache2)
Tasks: 123 (limit: 9403)
Memory: 27.0M
CGroup: /system.slice/apache2.service
├─ 96525 /usr/sbin/apache2 -k start
├─103029 /usr/sbin/apache2 -k start
├─103030 /usr/sbin/apache2 -k start
└─103031 /usr/sbin/apache2 -k start

•

tibit-netconf.service

- Tibit Communications, Inc. NetCONF Server
Loaded: loaded (/lib/systemd/system/tibit-netconf.service; enabled; vendor preset: enabled)

Active: active (running) since Thu 2024-06-27 08:47:44 EDT; 2 weeks 5 days ago

Main PID: 60768 (tibit-netconf)
Tasks: 17 (limit: 9403)
Memory: 60.7M
CGroup: /system.slice/tibit-netconf.service
├─60768 /opt/tibit/netconf/bin/tibit-netconf
└─60786 /opt/tibit/netconf/bin/tibit-netconf

•

tibit-netopeer2-server.service

- Tibit Communications, Inc. Netopeer2 Server
Loaded: loaded (/lib/systemd/system/tibit-netopeer2-server.service; enabled; vendor preset: enabled)

Active: active (running) since Thu 2024-06-27 08:47:44 EDT; 2 weeks 5 days ago

Main PID: 60772 (netopeer2-serve)

```
Tasks: 7 (limit: 9403)
Memory: 6.0M
CGroup: /system.slice/tibit-netopeer2-server.service
└─60772 /opt/tibit/netconf/bin/netopeer2-server -v 1 -t 55
```

Netplan

Validate the Netplan and ensure that the IP information is valid, the VM network interface name is correct, VLAN id 4090 is created and assigned, and that it is using a valid Netplan YAML tree structure.



Note: The netplan YAML file is located in /etc/netplan/.

Example:

<#root>

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CCO/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

cat /etc/netplan/01-network-manager-all.yaml

```
network:
  version: 2
  Renderer: Network Manager
  ethernets:
```

ens192: <- VM Network Adapter

dhcp4: no <- No DHCP as the IP is set statically

dhcp6: no

addresses: [10.122.140.232/28] <- IP of the VM Network adapter

gateway4: 10.122.140.225 <- GW of the IP Network

nameservers:

addresses: [172.18.108.43,172.18.108.34] <- Network DNS

```
  vlans:
    vlan.4090:
```

id: 4090

link: ens192 <- VM Network adapter

```
      dhcp4: no
      dhcp6: no
```

Verify the IP configuration of the VM and that the configured network adapter matches what is listed in the netplan YAML file.



Note: Usage of **sudo netplan --debug apply** is useful when testing the netplan prior to application.

Example:

<#root>

```
rpon@rpon-mgr:~$ ifconfig
```

ens192

```
: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

inet 10.122.140.232

```
 netmask 255.255.255.240 broadcast 10.122.140.239
        inet6 fe80::df4d:8d4d:4836:82aa prefixlen 64 scopeid 0x20<link>
        ether 00:50:56:84:3f:8f txqueuelen 1000 (Ethernet)
        RX packets 68933231 bytes 21671670389 (21.6 GB)
        RX errors 0 dropped 129 overruns 0 frame 0
        TX packets 36820200 bytes 71545432788 (71.5 GB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
-- snipped for brevity --
```

vlan.4090

```
: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
           inet6 fe80::250:56ff:fe84:3f8f prefixlen 64 scopeid 0x20<link>
           ether 00:50:56:84:3f:8f txqueuelen 1000 (Ethernet)
           RX packets 0 bytes 0 (0.0 B)
           RX errors 0 dropped 0 overruns 0 frame 0
           TX packets 1044 bytes 140547 (140.5 KB)
           TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

IP Connectivity

Verify IP connectivity to the XR Router hosting the PON controller via ping.

Example:

<#root>

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CCO/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

ping 10.122.140.226

```
PING 10.122.140.226 (10.122.140.226) 56(84) bytes of data.
64 bytes from 10.122.140.226: icmp_seq=1 ttl=255 time=1.01 ms
64 bytes from 10.122.140.226: icmp_seq=2 ttl=255 time=1.03 ms
64 bytes from 10.122.140.226: icmp_seq=3 ttl=255 time=1.13 ms
^C
--- 10.122.140.226 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.009/1.054/1.128/0.052 ms
```

Verify that the MongoDB TCP Port 27017 is open. If you are using a non-standard port for the MongoDB, verify it is open/listening via **netstat -tunl**.

Note: The standard MongoDB TCP port is 27017.

Note: The configuration file listed in step 4 also sets the TCP port configuration for the MongoDB to use.

Example:

<#root>

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CCO/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

netstat -tunl

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q  Local Address        Foreign Address State
tcp   0      0       127.0.0.53:53        0.0.0.0:*       LISTEN
tcp   0      0
```

127.0.0.1:27017

```
     0.0.0.0:*       LISTEN
tcp   0      0
```

10.122.140.232:27017

```
 0.0.0.0:*       LISTEN
```

MongoDB Configuration File

Verify the **mongod.conf** file is accurate, and has the correct IP listed under **bindIP:**.

Note: The MongoDB configuration file is located at /etc/mongod.conf

Example:

<#root>

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CCO/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

cat /etc/mongod.conf

```
# mongod.conf

storage:
dbPath: /var/lib/mongodb
journal:
enabled: true

systemLog:
destination: file
logAppend: true
path: /var/log/mongodb/mongod.log
logRotate: reopen

# network interfaces
net:
```

port: 27017

bindIp: 127.0.0.1,10.122.140.232

```
processManagement:
pidFilePath: /var/run/mongodb/mongod.pid
timeZoneInfo: /usr/share/zoneinfo

replication:
replSetName: "rs0"

-- snipped for brevity --
```

System Level Log Locations

System level logs for each service are managed within linux. These logs are stored within the /var/log directory, specifically under these trees.

MongoDB logs: **/var/log/mongod/mongod.log**
Apache logs: **/var/log/apache2/<filename>.log**
Virtual Machine Syslog: **/var/log/syslog**

Reference Documentation

- Cisco Support and Downloads Page
- Cisco Routed PON Solution Page
- Cisco Routed PON Installation Guide
- Cisco Routed PON Deployment Guide
- Release Notes for Cisco Routed PON, Cisco IOS XR Release 24.1.1 and 24.1.2