



Cisco Routed PON Installation Guide



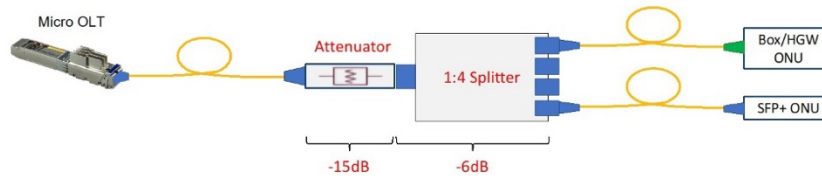
CAUTION

Refer to the Install Guide before Installation

Warranty Notice: Device Attenuation Required

Do not connect OLT directly to ONUs without proper attenuation. PON transceivers will be **permanently damaged** unless connected with **minimum 16dB** attenuation (20dB recommended). **Damage from optical overload will void warranty.**

Combination of attenuator and splitters can provide required attenuation. Refer to the example:



523682

Table of Contents

Solution Overview	4
Cisco Routed PON Manager	4
Netconf Server	5
Cisco Routed PON Controller	5
MongoDB Datastore	6
Server Sizing	6
Production Deployments	6
Installation for Production Deployments	7
MongoDB Installation	7
Installation	7
Installation steps:	7
Routed PON Manager Installation	9
Prerequisites:	9
Installation steps:	9
Configuration steps:	10
Table 1: PON Manager user_database.json file	11
Table 2: PON Manager databases.json file	12
Table 3: PON Manager recovery_email_configuration.json file	13
Creating Initial Account in PON Manager	14
Initialize Database	14
Post Installation – Verify Processes	15
Routed PON Netconf Server Installation	16
Prerequisites:	17
Installation steps:	17
Configuration steps:	17
Table 4: Netconf Initialization File	18
Appendix A: Method for creating a permanent Network configuration (Netplan)	21

Solution Overview

The Cisco Routed PON is Cisco's management solution for PON Networks. The Routed PON architecture is shown in Figure 1 and consists of the Routed PON Manager graphical user interface, Routed PON Netconf Server, and Routed PON Controller. Together these components provide a complete network management solution for provisioning and monitoring Routed PON OLT devices, as well as the subtended third-party ONU devices and third-party ONUs compliant with the 10G-EPON and XGS-PON standards.

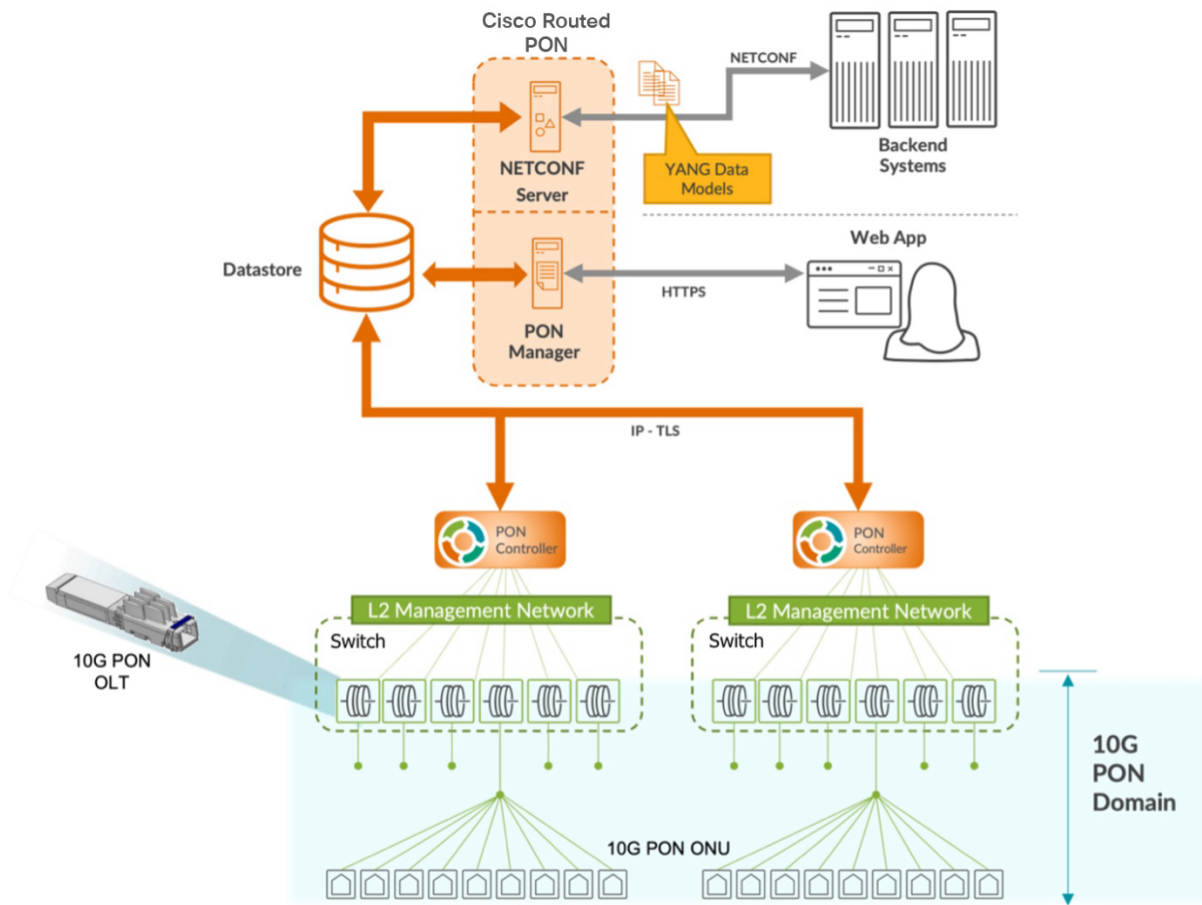


Figure 1 – Cisco Routed PON Management System Architecture

Cisco Routed PON Manager

The Cisco Routed PON Manager is a single-page web application (Web App) and an accompanying REST API that provides a graphical user interface for managing the PON Network. The Web App is built on the Angular web application framework, which provides an HTML and JavaScript front-end user interface. The REST API accompanies the Web App for the purposes of providing access to MongoDB for managing Routed PON Manager users and the PON Network.

The Routed PON Manager has the following features:

- Alarm management.
- Dashboard view with a summary of PON network conditions.
- Device monitoring and statistics.
- Device provisioning and management.
- Logging for diagnostics and troubleshooting.
- Routed PON Controller database management.
- Routed PON Manager user management.
- Polyglot graphical OMCI (and future 10G EPON OAM) service configuration tool.
- Service configuration, including VLANs, Service Level Agreements (SLAs), 802.1X Authentication, and DHCP Relay.

Note that the REST API should only be used with the Routed PON Manager. At this time, the REST API is not a published interface and will change in future releases. Future versions for Routed PON Manager may make the REST API available directly to customer applications.

Netconf Server

The Cisco Routed PON Netconf Server provides a standard Netconf interface and customer facing API for managing the PON network. The Netconf Server is built on the ConfD architecture and interfaces with MongoDB. In addition to standard YANG models, Cisco YANG models provide a complete Netconf management solution for PON Controllers, OLTs, and ONU devices.

Cisco Routed PON Controller

The Cisco Routed PON Controller is a stateless management controller and device driver application for configuring and monitoring the end points in a Cisco OLT PON network. MongoDB serves as the northbound application programming interface for the PON Controller. The PON Controller applies configuration to OLT and ONU devices from documents stored in MongoDB. The Routed PON Controller uses IEEE 1904.2 packets (Layer 2 Ethernet) to communicate with the OLT and ONU devices. The PON Controller also collects state information, statistics, and logs from devices and reports the information to higher layer applications through MongoDB. The PON Manager and Netconf interfaces manage the PON Controller through MongoDB\MongoDB Datastore.

MongoDB Datastore

The Mongo database (MongoDB) provides the datastore for the Cisco Routed PON Management System. The MongoDB datastore contains all the configuration, state, statistics, alarms, and logging data for the devices in the PON network. Northbound interfaces, such as the Routed PON Manager, Routed PON Netconf Server, and customer applications interface with MongoDB to provision and retrieve monitoring information for devices in the PON network. MongoDB serves as the interface between the PON Manager, Netconf, and the PON Controller.

Provisioning data generally flows "downstream" through the management network. The PON Manager and Netconf interfaces write device configuration to MongoDB. The PON Controller reads the configuration data from MongoDB and programs the OLT and ONU devices accordingly.

Monitoring data, including device state, statistics, alarms, and logging, is collected and flows "upstream" through the management network. The PON Controller periodically collects state information from devices in the PON network and writes the monitoring data to MongoDB. The PON Manager reads the monitoring data from MongoDB for display in the Web App.

Server Sizing

Production Deployments

It is recommended to run each of the Routed PON Management applications on separate VMs (or servers). The following table provides guidelines for deployments.

Note: The sizing requirements are for the Application process only and does not include the base OS (Ubuntu 20.04).

Application	Subscriber Size	
	3072 Subscriber	100K Subscribers
PON Manager	2 vCPU 8 GB RAM 1 GB disk space	4 vCPU 16 GB RAM 1 GB disk space
Netconf	1 Instance: 2 vCPU 1 GB RAM 50 MB disk space	3 x Netconf Instances: 6 vCPU 3 GB RAM 150 MB disk space
MongoDB	2 vCPU 8 GB RAM 10 GB disk space	16 vCPU 64 GB RAM 100 GB disk space

¹ This is an estimate. Please contact Cisco Support for details

Installation for Production Deployments

For production deployments, Cisco recommends installing each of the Routed PON Management applications on separate VMs (or servers).

Note: Cisco PON Manager package is signed by Cisco and is available on the official [Cisco Software Download](#) page. Refer to README document in the extracted Cisco PON Manager package for the instructions to verify the signing of the package:

```
├── R4.0.0-Cisco-UB2004-sign.tar.gz
├── README
└── verify.tar.gz
```

For more information on the Cisco PON Manager package, refer to the [Cisco Routed PON Manager User Guide](#).

MongoDB Installation

MongoDB serves the purpose of being the datastore for the Cisco Routed PON Management System. This includes all PON-specific information as well as user models.

For production deployments, Cisco recommends configuring multiple servers or VMs in a Replica Set using Authentication and TLS. Please reference the Cisco Tech Note, [Mongo Replication with Authentication and TLS/SSL](#) for configuration of each Mongo instance.

Installation

To install the MongoDB Community Edition on each instance,

1. Access the official MongoDB website.
2. Using the table of contents at the top of the page, select the option, “Uninstall MongoDB Community Edition”, and follow the instructions to uninstall MongoDB if installed.
3. Using the same table of contents at the top of the page, select the option; “Install MongoDB Community Edition using .deb Packages”, and follow the instructions to install the latest version of MongoDB.
 - Ensure that you are using the “Ubuntu 20.04” options.

Installation steps:

1. Import the MongoDB Public GPG (GNU Privacy Guard) Key.

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```
2. Create a List File for MongoDB.

```
echo "deb [ arch=amd64,arm64 ]
https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.4
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
```
3. Update the Package Database

```
sudo apt update
```

4. Install MongoDB Packages

```
sudo apt install mongodb-org -y
```

5. Start MongoDB Service

```
sudo systemctl start mongod.service
```

6. Enable MongoDB to Start on Boot

```
sudo systemctl enable mongod
```

7. Modify the MongoDB Configuration File

If your server has multiple IP addresses, add the IP addresses to the `/etc/mongod.conf` file.

8. Edit the `bindIp` parameter to include the server's IP addresses

```
bindIp: 127.0.0.1, <Global NW ip>, <Lab N/W ip>
```

Example:

```
bindIp: 127.0.0.1,10.106.211.115,5.0.0.209
```

9. Add the replication configuration:

replication:

```
replSetName: rs0
```

10. Restart MongoDB Service

```
sudo systemctl restart mongod.service
```

11. Check MongoDB Service Status

```
sudo systemctl status mongod
```

12. Install MongoDB Shell

```
sudo apt-get install -y mongodb-mongosh
```

13. Open MongoDB Shell

```
mongosh
```

14. Initialize the Replica Set. In the MongoDB shell, run the following command to initiate the replica set:

```
rs.initiate({
  _id: "rs0",
  members: [
    { _id: 0, host: "<Global NW ip>:27017" }
  ]
})
```

e.g:

```
rs.initiate( {
  _id : "rs0",
  members: [
    { _id: 0, host: "10.106.211.115:27017" },
  ]
})
```

15. Restart MongoDB Service Again

```
sudo systemctl restart mongod.service
```

Routed PON Manager Installation

The Routed PON Manager software is composed of a graphical user front-end web application (Web App) and a RESTful interface (REST API) that provides access to the MongoDB datastore.

Prerequisites:

- Ensure that you have an active internet connection.
- Install/upgrade existing python3 package to be equal to or greater than version 3.6.7.
- Install/upgrade existing pip3 package to be equal to or greater than version 9.0.0.
- Ensure that you have your MongoDB Server running that will manage user authentication and serve as your default database.
- Ensure you have the latest ubuntu updates

Installation steps:

4. Transfer/download the .zip file to Ubuntu host or VM.
4. From a Linux shell, unpack the .zip file and change to the following directory:

```
unzip R4.0.0-Cisco-UB2004.zip  
cd R4.0.0-Cisco-UB2004/R4.0.0-PonManager-UB8104
```
3. Optional | If desired, you can configure the self-signed certificate created during installation by modifying the configuration file within this package directory; `'tbit_ponmgr_selfsigned_cert_req.conf'`.

There are two sections that may be modified:

- [req_distinguished_name]

countryName	Country code. Ex: 'US' (United States), etc
ST	State or Province
localityName	City or Locality
organizationName	Name of organization
organizationalUnitName	Name of group/unit within organization
CN	Fully Qualified Domain Name of system

- [alt_names]

DNS.* <i>number</i> *	Name to access web server. Configure additional names by adding more 'DNS.x' lines with increasing numbers in place of 'x' in the example
-----------------------	--

	above. Ex. DNS.1 = tibitdev-web DNS.2 = tibitdev-api DNS.3 = www.MCMS.com
IP.*number*	IP of machine. Configure additional interfaces by adding more 'IP.x' lines with increasing numbers in place of 'x' in the example above. Ex. IP.1 = 127.0.0.1 IP.2 = 10.1.10.225

4. Use apt-get to install the package.

Note: By default, Routed PON Manager will deploy over HTTPS utilizing a self-signed certificate created during installation. THIS IS NOT SAFE FOR A PRODUCTION ENVIRONMENT. If installing Routed PON Manager for any use other than evaluation, you should use legitimate certificates. Step 4 describes how to update the self-signed certificate and key with valid certificate and key.

- Use the following command to install PON Manager with the default self-signed certificate:

```
sudo apt install ./tibit-ponmgr_R4.0.0_all.deb
```

- *Optional* | Use the following command to install PON Manager with the customized self-signed certificate configured in Step 2 above:

```
sudo TIBIT_PONMGR_CERT_REQ=/path/to/file.conf apt \
install ./tibit-ponmgr_R4.0.0_all.deb
```

Configuration steps:

Ensure that you have root/sudo access to modify files.

1. Navigate to the directory: '/etc/tibit/ponmgr/'
 - Within this directory, there is one PON Manager configuration file that will need to be edited with a text editor of your choice.
 - Update the PON Manager configuration files to match the settings for connectivity to MongoDB.
 - user_database.json – Stores configuration for database to be used for user authentication (see Table 1: PON Manager user_database.json file)

- databases.json – Stores list of all available PON databases in PON Manager (see Table 2: PON Manager databases.json file)
- recovery_email_configuration.json – Configures SMTP email server for sending user password recovery emails (see Table 3: PON Manager recovery_email_configuration.json file)

2. Verify the PON Manager status:

```
sudo systemctl status apache2.service
```

Table 1: PON Manager user_database.json file

Key	Description
host	Hostname/IP Address of the MongoDB server hosting your user database.
name	Name of your user database. Default user database name is 'tibat_users'. <i>(If it doesn't exist, it will be created)</i>
port	MongoDB port number.
auth_enable	Boolean value determining if the MongoDB server at <i>host:port</i> is using authentication.
auth_db	Name of your MongoDB authentication database. <i>(Used when auth_enabled = true)</i>
username	The username of the MongoDB user to authenticate with. <i>(Used when auth_enabled = true)</i>
password	The password of the specified MongoDB user. <i>(Used when auth_enabled = true)</i>
tls_enable	Boolean value specifying whether the MongoDB server at <i>host:port</i> is using encryption.
ca_cert_path	The local path to the encryption certificate.
compression	Boolean value specifying whether to enable snappy compression for MongoDB connections.
dns_srv	Boolean value specifying whether the MongoDB server is using a DNS seed list.

Key	Description
db_uri	Raw MongoDB connection URI. All Other fields are ignored if this is used.
replica_set_enable	Boolean value specifying if the MongoDB server is running as a replica set.
replica_set_name	The name of the MongoDB replica set. <i>(Used when replica_set_enabled = true)</i>
replica_set_hosts	List of hosts to be used as the MongoDB replica set. <i>(Used when replica_set_enabled = true)</i>
django_key	Cryptographic key used by django. The format of the field is a freeform string, however, Django recommends it be minimum 50 characters with minimum 5 unique characters. For more information on the Django Key, please refer to the Routed PON Manager User Guide. <i>(Note: A randomized key will be generated during installation if the 'django_key' field is not present.)</i>

Table 2: PON Manager databases.json file

Key	Description
host	Hostname/IP Address of your mongoDB server hosting your PON Controller database
name	Name of your PON Controller database. Default PON Controller database name is 'tibit_pon_controller'. <i>(If it doesn't exist, it will be created)</i>
port	MongoDB server port number.
auth_enable	Boolean value determining if the MongoDB server at <i>host:port</i> is using authentication.
auth_db	Name of your MongoDB authentication database. <i>(Used when auth_enabled = true)</i>
username	The username of the MongoDB user to authenticate with. <i>(Used when auth_enabled = true)</i>
password	The password of the specified MongoDB user. <i>(Used when auth_enabled = true)</i>

Key	Description
tls_enable	Boolean value specifying whether the MongoDB server at <i>host:port</i> is using encryption.
ca_cert_path	The local path to the encryption certificate.
dns_srv	Boolean value specifying whether the MongoDB server is using a DNS seed list.
db_uri	Raw MongoDB connection URI. All Other fields are ignored if this is used.
replica_set_enable	Boolean value specifying if the MongoDB server is running as a replica set.
replica_set_name	The name of the MongoDB replica set. <i>(Used when replica_set_enabled = true)</i>
replica_set_hosts	List of hosts to be used as the MongoDB replica set. <i>(Used when replica_set_enabled = true)</i>

Table 3: PON Manager *recovery_email_configuration.json* file

Key	Description
host	Host to use for sending email
port	Port to use for the SMTP server defined above <i>Type: String</i>
use_tls	Use TLS connection <i>'true' or 'false'</i> <i>Type: String</i>
user	Username for the SMTP server
password	Password for user

For more information on installing the Routed PON Manager, please reference the R4.0.0 Routed PON Manager User Guide. The section “Installation” on page 21 provides details on requirements and dependencies, package contents and installation methods.

Creating Initial Account in PON Manager

This section describes how to create the initial login account in PON Manager. This account will have administrative privileges and supports the ability to create new accounts for access into the PON Manager.

- Connect to PON Manager using any of the IP addresses configured on the ubuntu system
- Create an Account. Ensure the password meets the Password Requirements listed.

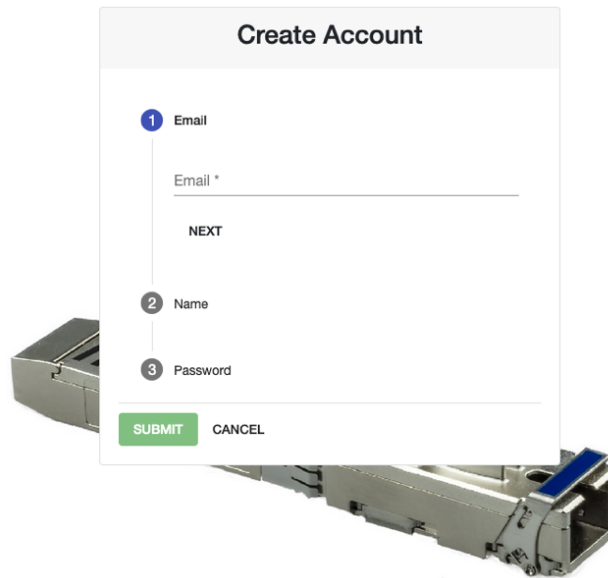


Figure 2 - Create Initial Login Account

Initialize Database

This section describes how to Initialize the Database after the upgrade process completes. Some items (specifically OLT Firmware) will need to be added into the MongoDB.

- Connect to the PON Manager
- Select the Global Config button on the main menu.
- Select Databases
- Then select the **Initialize** tab (see screenshot below)
- Select the Database to initialize (i.e. Default)
- Check (or uncheck) all the boxes for the information wanted to copy into the Database
- Click **Sart** in the **Database to initialize**

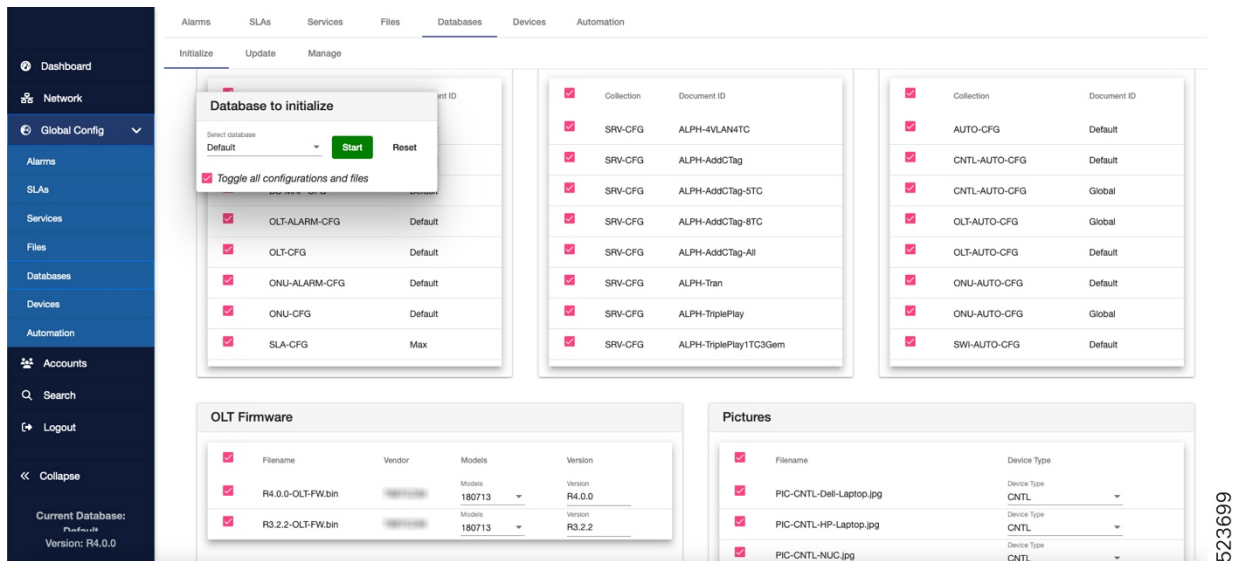


Figure 3: Initialize Database

Post Installation – Verify Processes

This section describes how to verify the PON Manager, PON Controller and MongoDB processes are running.

1. Use the script `status.sh` provided with the release to check the status of the installation.

```
user@system:~/R4.0.0-Cisco-UB2004$ ./status.sh
MCMS Component Versions:
  PON Manager: R4.0.0
  PON NETCONF: R4.0.0
  PON Controller: R4.0.0
```

- `mongod.service` - MongoDB Database Server
 - Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
 - Active: **active (running)** since Tue 2021-11-30 14:56:17 PST; 4 months 16 days ago
 - Docs: <https://docs.mongodb.org/manual>
 - Main PID: 656 (mongod)
 - CGroup: /system.slice/mongod.service
 - └─656 /usr/bin/mongod --config /etc/mongod.conf
- `apache2.service` - The Apache HTTP Server
 - Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
 - Drop-In: /lib/systemd/system/apache2.service.d
 - └─apache2-systemd.conf
 - Active: **active (running)** since Fri 2022-04-08 09:56:02 PDT; 1 weeks 2 days ago
 - Process: 11716 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
 - Process: 29373 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 - Process: 11722 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 - Main PID: 11726 (apache2)
 - Tasks: 151 (limit: 4915)
 - CGroup: /system.slice/apache2.service
- `tibit-netopeer2-server.service` - Tibit Communications, Inc. Netopeer2 Server
 - Loaded: loaded (/lib/systemd/system/tibit-netopeer2-server.service; enabled; vendor: enabled)
 - Active: **active (running)** since Fri 2022-04-08 09:54:42 PDT; 1 weeks 2 days ago
 - Main PID: 10819 (netopeer2-serve)

```

Tasks: 7 (limit: 4915)
CGroup: /system.slice/tibit-netopeer2-server.service
└─10819 /opt/tibit/netconf/bin/netopeer2-server -v 1 -t 55

● tibit-netconf.service - Tibit Communications, Inc. NetCONF Server
   Loaded: loaded (/lib/systemd/system/tibit-netconf.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-04-08 09:54:42 PDT; 1 weeks 2 days ago
 Main PID: 10812 (tibit-netconf)
    Tasks: 12 (limit: 4915)
   CGroup: /system.slice/tibit-netconf.service
           └─10812 /opt/tibit/netconf/bin/tibit-netconf
           └─10821 /opt/tibit/netconf/bin/tibit-netconf

```

2. Verify MongoDB is running

```
user@system:~$ sudo systemctl status mongod.service
```

```

● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-08-30 11:56:38 PDT; 3 days ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 15035 (mongod)
    CGroup: /system.slice/mongod.service
           └─15035 /usr/bin/mongod --config /etc/mongod.conf

```

3. Verify PON Manager Apache Web Server is running

```
user@system:~$ sudo systemctl status apache2.service
```

```

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
 Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Fri 2019-08-16 15:19:09 PDT; 1 weeks 2 days ago
 Process: 2981 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 8471 (apache2)

```

Routed PON Netconf Server Installation

The Routed PON Netconf Server provides an interface for managing the PON Controller, OLTs, and ONU devices using standard NETCONF protocols and tools.

The Routed PON NETCONF/YANG interface supports the following IETF standards:

- RFC 4742 NETCONF Protocol over Secure Shell (SSH)
- RFC 6241 Network Configuration Protocol (NETCONF)
- RFC 6020 YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)
- RFC 7950 The YANG 1.1 Data Modeling Language

The Routed PON NETCONF/YANG interface supports the following Broadband Forum standards:

- TR-383, Issue 1, Amendment 3 Common YANG Modules for Access Networks
- TR-385, Issue 2 ITU-T PON YANG Modules

Prerequisites:

- Ensure that you have an active internet connection.
- Ensure that you have your MongoDB Server running and configured as a Replica Set.
- Ensure the host or VM is running either Ubuntu 18.04 or 20.04 with the latest ubuntu updates
- Root level access is required to install the software.

Installation steps:

1. From a Linux shell, unpack the .zip file and change to the following directory:

```
cd R4.0.0-Cisco-UB2004/MCMS-R4.0.0-Netconf-ConfD-UB2004
```
2. Use apt-get to install the package.

```
sudo apt-get install ./cisco-netconf-confd_R5.0.0_amd64.deb
```

Configuration steps:

Ensure that you have root/sudo access to modify files.

1. Navigate to the directory: '/etc/tibit/netconf/'
 - Within this directory, there is one Netconf configuration file that will need to be edited with a text editor of your choice.
 - Update the Netconf configuration file to match the settings for connectivity to MongoDB.
 - See Table 4: Netconf Initialization File for details on each configuration parameter.
2. Check the status of ConfD service:

```
systemctl status cisco-netconf.service  
systemctl status cisco-confd.service  
systemctl restart mongod.service
```
3. Configure the Routed PON Netconf Server's MongoDB connection information. Edit the file '/etc/tibit/netconf/NetconfInit.json' and modify the IP address, port number, and database name as required.

```
cat /etc/tibit/netconf/NetconfInit.json  
{  
  "Logging": {  
    "Filename" : "/var/log/tibit/netconf.log",  
    "FileCount" : 3,  
    "FileSize" : 1024000,  
    "Netconf" : {  
      "Console" : "INFO",  
      "File" : "INFO",  
      "Syslog" : "INFO"  
    }  
  },  
  "MongoDB": {
```

```

    "auth_db": "tibit_users",
    "auth_enable": false,
    "ca_cert_path": "/etc/tibit/ca.pem",
    "host": "127.0.0.1",
    "name": "tibit_pon_controller",
    "password": "",
    "port": "27017",
    "tls_enable": false,
    "username": ""
  }
}

```

3. Restart the Routed PON Netconf Server using Linux systemd scripts.

```
sudo systemctl restart tibit-netconf
```

Table 4: Netconf Initialization File

	Field	Values	Description
Logging			
	Filename	"directory-path"	Directory path where log file is to be stored
	FileCount	integer	The maximum number of log files to store for each module, before overwriting the oldest.
	FileSize	integer	The maximum size in bytes of log file
	Netconf	true or false	
MongoDB			
	auth_db	"string"	Name of the MongoDB database that will be used for user authentication and authorization.
	auth_enable	true or false	Set to true if MongoDB access control is enabled. Set to false if MongoDB access control is not enabled.
	ca_cert_path	"/directory/filename"	Directory path to the file (in PEM format) containing the Root CA certificate that was used to sign the server certificate.
	db_uri	"string"	URI to use to connect to the Mongo database. If specified, all other Mongo database connection settings are ignored. Refer to the MongoDB website for more information on the Mongo Connection String URI format.
	dns_srv	true or false	Set to true to enable DNS-based Mongo database discovery. If set to true, the "host" field must contain an FQDN that will be used to lookup SRV records for the Mongo database. If set to true, then the "port" field is ignored as the port number will be acquired from the DNS SRV record.
	host	"ip-address"	IP address used to access the Mongo database. Example:"127.0.0.1".

	Field	Values	Description
			If dns_srv is set to true, then this field must contain the FQDN used to lookup DNS records for the Mongo database.
	name	"database-name"	Name of the database.
	port	"network-port"	Network port number to access the Mongo database. Example:"27017". If dns_srv is set to true, then this field is ignored.
	replica_set_enable	true or false	Set to true to use Mongo replica sets and the replica_set_name and replica_set_hosts parameters in the configuration file
	replica_set_name	"string"	nly applicable if replica_set_enable is set to true. Contains the name of the replicaSet parameter that is common to the members of the set.
	replica_set_hosts	List of host:ports	Only applicable if replica_set_enable is set to true and dns_srv is set to false. This field contains a list of hostname:port entries used to connect to the Mongo database.
	tls_enable	true or false	Set to true to enable TLS when connecting to MongoDB. Set to false to disable TLS when connecting to MongoDB.
	username	"string"	The username to use to connect to the MongoDB database.
	password	"string"	The password to use to connect to the MongoDB database.
NETCONF			
	System Name	"string"	The Netconf System Name

For more information on installing the Routed PON Netconf Server, please reference the R4.0.0 Routed PON Netconf Server User Guide (MCMS-R4.0.0-NetconfServer-UserGuide-TN037.pdf). The section "Installation" on page 14 provides details on requirements and dependencies, package contents and installation methods.



Appendix A: Method for creating a permanent Network configuration (Netplan)

In the requirements section on Page 10, it shows you the commands to create temporary interfaces to be used by the Routed PON management system. The procedure below creates permanent interfaces.

Verify the ethernet interfaces are configured on the ubuntu system

- 1) Look for your Ethernet Interfaces: "ifconfig" or "ip a".
<<< make note of interfaces listed >>>
- 2) There are multiple ways to configure interfaces on Ubuntu 20.04
- 3) This is the example for creating interfaces via Netplan
- 4) Netplan is located in the /etc/netplan directory
- 5) Located here will be a file similar in name to "01-network-manager-all.yaml"
- 6) Edit this file with your favorite editor such as "nano" or "vi"

```
user@system:~$ sudo nano /etc/netplan/<net-plan-name>.yaml
```

Sample Netplan text <<< [Make sure indentation is consistent](#) >>>

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    eno1:                                <<< Routed PON IP Interface >>>
      dhcp4: no                          <<< No DHCP >>>
      dhcp6: no                          <<< No DHCP >>>
      addresses: [172.16.41.5/24]         <<< Static IPv4 >>>
      gateway4: 172.16.41.1             <<< IPv4 default Gateway >>>
      nameservers:                       <<< DNS Addresses >>>
        search: [local]                 <<< Search on local machine first >>>
        addresses: [8.8.8.8, 8.8.4.4]   <<< external DNS servers >>>
    eno2:                                <<< PON Controller Interface >>>
      dhcp4: no                          <<< No DHCP >>>
      dhcp6: no                          <<< No DHCP >>>
  vlans:                                <<< Configured VLANs >>>
    vlan4090:                            <<< "I2EthInterfaceName" >>>
      id: 4090                          <<< VLAN number >>>
      link: eno2                         <<< PON Controller Interface >>>
```

After finished editing, Save and exit, then enter "sudo netplan apply" to enable new configuration.

-
- 1) Verify ubuntu system has connectivity to Internet.
 - 2) Run “ip a” and look for the interfaces you’ve created. You should see an interface with the IP interface you configured, and an interface called “vlan4090” shown (or the name you chose to call the "I2EthInterfaceName").

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)