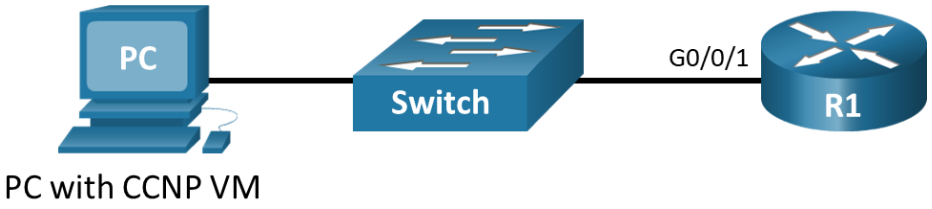
Lab 20: Sử dụng NETCONF kết nối đến thiết bị IOS XE

Sơ đồ



Bảng địa chỉ

Thiết bị	Cổng	IP Address	Subnet Mask
R1	G0/0/1	192.168.1.1	255.255.255.0
PC	NIC	DHCP	DHCP

Mục tiêu

Part 1: Xây dựng mạng và kiểm tra kết nối

Part 2: Sử dụng NETCONF Session để thu thập thông tin

Part 3: Sử dụng ncclient để kết nối NETCONF

Part 4: Sử dụng ncclient để lấy cấu hình

Part 5: Sử dụng ncclient để cấu hình

Part 6: Sửa đổi chương trình được sử dụng trong lab này

Bối cảnh/ Kịch bản

Giao thức cấu hình mạng (NETCONF), được xác định trong RFC 4741 và 6241, sử dụng các mô hình dữ liệu YANG để liên lạc với các thiết bị khác nhau trên mạng. YANG (một thế hệ tiếp theo khác) là một ngôn ngữ mô hình hóa dữ liệu. Ngôn ngữ này xác định dữ liệu được gửi qua các giao thức quản lý mạng, như NETCONF. Khi sử dụng NETCONF để truy cập thiết bị IOS XE, dữ liệu được trả về ở định dạng XML.

Trong lab này, chúng ta sẽ sử dụng NETCONF client, ncclient, là mô-đun Python để tạo kịch bản (script) phía client. Dùng ncclient để xác minh NETCONF đã được cấu hình, truy xuất cấu hình thiết bị và sửa đổi cấu hình thiết bị.

Các tài nguyên cần thiết

- 1 Router (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 Switch
- 1 PC (Choice of operating system with Cisco Networking Academy CCNP VM running in a virtual machine client and terminal emulation program)
- Cáp Ethernet

Hướng dẫn

Part 1: Xây dựng mạng và kiểm tra kết nối

Ghi chú: Bỏ qua Phần 1 nếu bạn đã hoàn thành nó trong lab trước đó. Tuy nhiên, hãy chắc chắn máy ảo CCNP của bạn có thể ping R1.

Trong Phần 1, bạn sẽ nối cáp các thiết bị, khởi động CCNP VM, cấu hình R1 để truy cập NETCONF và RESTCONF qua kết nối SSH. Sau đó, bạn sẽ xác minh kết nối giữa CCNP VM và R1 cũng như kiểm tra kết nối SSH với R1.

Step 1: Cáp mạng như được hiển thị trong cấu trúc liên kết.

Kết nối các thiết bị như sơ đồ cấu trúc liên kết và cáp nếu cần..

Step 2: Mở máy ảo CCNP VM.

Lưu ý: Nếu bạn chưa hoàn thành Lab - Cài đặt Máy ảo CCNP, hãy thực hiện ngay trước khi tiếp tục với lab này.

a. Mở VirtualBox. Khởi động máy ảo CCNP VM.

b. Nhập mật khẩu **StudentPass** để đăng nhập VM nếu cần.

Step 3: Cấu hình R1.

Đi vào R1 và dán cấu hình sau vào CLI để cấu hình các cài đặt cơ bản và bật NETCONF, RESTCONF và SSH.

```
enable
configure terminal
hostname R1
no ip domain lookup
line con 0
logging synchronous
 exec-timeout 0 0
 logging synchronous
line vty 0 15
 exec-t 0 0
 logg sync
 login local
 transport input ssh
ip domain name example.netacad.com
crypto key generate rsa modulus 2048
username cisco priv 15 password cisco123!
interface GigabitEthernet0/0/1
 description Link to PC
 ip address 192.168.1.1 255.255.255.0
 no shutdown
ip dhcp excluded-address 192.168.1.1 192.168.1.10
!Configure a DHCP server to assign IPv4 addressing to the CCNP VM
ip dhcp pool LAN
```

```
 network 192.168.1.0 /24
 default-router 192.168.1.1
 domain-name example.netacad.com
 end
copy run start
```

Step 4: Xác minh CCNP VM có thể ping default gateway.

a. Trong **CCNP VM**, mở cửa sổ terminal

b. Xác minh CCNP VM được kết nối với R1 bằng cách nhập địa chỉ IP để xác minh rằng CCNP VM đã nhận địa chỉ IP từ máy chủ DHCP, hoặc đơn giản bằng cách ping R1 ở 192.168.1.1. Nhập Ctrl + C để thoát ra khỏi ping, như thể hiện trong ví dụ bên dưới.

```
student@CCNP:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 00:50:56:b3:72:3b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.15/24 brd 192.168.1.255 scope global dynamic noprefixroute ens160
       valid_lft 79564sec preferred_lft 79564sec
    inet6 fe80::1ae4:952f:402d:6b1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 00:50:56:b3:26:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.183/24 brd 192.168.50.255 scope global dynamic noprefixroute
ens192
       valid_lft 70687sec preferred_lft 70687sec
    inet6 fe80::4c87:a2b3:aa9:5470/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

student@CCNP:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=255 time=0.703 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=255 time=0.748 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=255 time=0.757 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.703/0.736/0.757/0.023 ms
```

c. Nếu máy ảo CCNP chưa nhận được địa chỉ IPv4, hãy kiểm tra các kết nối vật lý của bạn giữa PC chủ và R1. Ngoài ra, xác minh rằng R1 được cấu hình đúng theo bước trước.

Step 5: Thiết lập kết nối SSH trên R1.

a. Mở PuTTY SSH Client.

b. NHập địa chỉ IP cho default gateway, 192.168.1.1, va nhấn **Open**.

c. Chúng ta có thể đăng nhập với username **cisco** và password **cisco123!**. Nếu không kiểm tra lại phần cấu hình SSH trên R1.

d. Giữ phiên PUTTY mở.

Part 2: Sử dụng NETCONF Session để thu thập thông tin

Trong Phần 2, bạn sẽ kiểm tra xem NETCONF đã chạy chưa, bật NETCONF nếu không và xác minh rằng NETCONF đã sẵn sàng cho kết nối SSH. Sau đó, bạn sẽ kết nối quy trình NETCONF, bắt đầu phiên NETCONF, thu thập thông tin giao diện và đóng phiên.

Step 1: Kiểm tra NETCONF đang chạy trên R1.

a. NETCONF có thể đã chạy nếu một người khác kích hoạt nó hoặc nếu phiên bản iOS mới hơn kích hoạt nó theo mặc định. Từ thiết bị đầu cuối PuTTY, sử dụng lệnh quy trình quản lý yang phần mềm nền tảng chương trình để xem daemon NETCONF SSH (ncsshd) có đang chạy không.

```
R1# show platform software yang-management process
confd            : Not Running
nesd             : Not Running
syncfd           : Not Running
ncsshd           : Not Running
dmiauthd         : Not Running
nginx            : Running
ndbmand          : Not Running
pubd             : Not Running
```

b. Nếu NETCONF không chạy như hiển thị trong đầu ra ở trên, hãy nhập lệnh cấu hình netconf-yang.

```
R1# config t
R1(config)# netconf-yang
```

c. Bây giờ kiểm tra lại NETCONF đang chạy chưa.

```
R1# show platform software yang-management process
confd            : Running
nesd             : Running
syncfd           : Running
ncsshd           : Running
dmiauthd         : Running
nginx            : Running
ndbmand          : Running
pubd             : Running
```

Step 2: Truy nhập tiến trình NETCONF qua SSH terminal.

Bạn có thể thiết lập phiên SSH với R1 bằng PuTTY. Tuy nhiên, chức năng sao chép và dán trong cửa sổ thiết bị đầu cuối PuTTY bên trong CCNP VM có thể gặp vấn đề. Do đó, sử dụng cửa sổ dòng lệnh để bắt đầu phiên SSH với R1.

a. Nhập lệnh sau trong cửa sổ terminal. Sau đó nhập cisco123! làm mật khẩu. Nếu bạn nhập lệnh không chính xác, hãy sử dụng phím mũi tên lên để gọi lại lệnh bạn nhập, sau đó tìm và chỉnh sửa lỗi của bạn.

```
student@CCNP:~$ ssh -oHostKeyAlgorithms=+ssh-dss cisco@192.168.1.1 -p 830 -s
netconf
cisco@192.168.1.1's password:
ssh vnpro@10.215.28.132 -p 830 -s netconf
```

b. R1 sẽ trả lời bằng một tin nhắn xin chào bao gồm tất cả các capabilities của nó. Phần cuối của tin nhắn được xác định bằng **]]>]]>**.

```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability>
<capability>urn:ietf:params:netconf:capability:notification:1.0</capability>
<capability>urn:ietf:params:netconf:capability:interleave:1.0</capability>
<capability>urn:ietf:params:netconf:capability:with-defaults:1.0?basic-
mode=explicit&amp;also-supported=report-all-tagged</capability>
<capability>urn:ietf:params:netconf:capability:yang-library:1.0?revision=2016-06-
21&amp;module-set-id=d26d4d9ff23c0afcad7994ca2b832a9b</capability>
<capability>http://tail-f.com/ns/netconf/actions/1.0</capability>
<capability>http://tail-f.com/ns/netconf/extensions</capability>
(output omitted)
<capability>urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults?module=ietf-
netconf-with-defaults&amp;revision=2011-06-01</capability>
<capability>
        urn:ietf:params:netconf:capability:notification:1.1
<output omitted>
      </capability>
</capabilities>
<session-id>20</session-id></hello>]]>]]>
```

Step 3: Bắt đầu phiên NETCONF bằng cách gửi tin nhắn xin chào từ client.

Để bắt đầu một phiên NETCONF, khách hàng cần gửi tin nhắn xin chào của riêng mình. Thông báo xin chào phải bao gồm phiên bản capabilities cơ bản của NETCONF mà khách hàng muốn sử dụng.

a. Sao chép và dán mã XML sau vào phiên SSH. Lưu ý rằng phần cuối của thông điệp xin chào của client được xác định bằng **]]>]]>**.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <capabilities>
   <capability>urn:ietf:params:netconf:base:1.0</capability>
 </capabilities>
</hello>
]]>]]>
```

b. Trên R1, sử dụng lệnh **show netconf-yang sessions** để kiểm tra phiên NETCONF đã được bắt đầu.

```
R1# show netconf-yang sessions
R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore

Number of sessions : 1

session-id  transport    username              source-host           global-lock
--------------------------------------------------------------------------------
20          netconf-ssh  cisco                 192.168.1.12          None
```

Step 4: Gửi tin nhắn RPC đến thiết bị IOS XE.

Trong phiên SSH, NETCONF client có thể dùng tin nhắn Remote Procedure Call (RPC) để gửi NETCONF operations đến thiết bị IOS XE. Bảng sau sẽ liệt kê những NETCONF operations thường được sử dụng.

Operation	Description
<get>	Truy xuất cấu hình đang chạy và thông tin trạng thái thiết bị
<get-config>	Truy xuất tất cả hoặc một phần lưu trữ của dữ liệu cấu hình đã chỉ định
<edit-config>	Tải tất cả hoặc một phần cấu hình vào kho dữ liệu cấu hình đã chỉ định
<copy-config>	Thay thế toàn bộ kho dữ liệu cấu hình bằng một kho khác
<delete-config>	Xóa kho dữ liệu cấu hình
<commit>	Sao chép kho dữ liệu ứng viên đến kho dữ liệu đang chạy
<lock> / <unlock>	Khóa hoặc mở khóa toàn bộ hệ thống lưu trữ dữ liệu cấu hình
<close-session>	Kết thúc phiên NETCONF
<kill-session>	Buộc chấm dứt phiên NETCONF

a. Sao chép và dán RPC sau nhận mã XML thông báo vào phiên SSH cuối cùng để truy xuất thông tin về các giao diện trên R1.

```
<rpc message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <get>
  <filter>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"/>
   </filter>
 </get>
</rpc>
]]>]]>
```

b. Hãy nhớ lại rằng XML không yêu cầu thụt lề hoặc khoảng trắng. Do đó, R1 sẽ trả về một chuỗi dữ liệu XML dài.

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-
id="103"><data><interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-
interfaces"><interface><name>GigabitEthernet0</name><type
xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
```

type">ianaift:ethernetCsmacd</type><enabled>true</enabled><ipv4
xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv4><ipv6
xmlns="urn:ietf:params:xml:ns:yang:ietf-
ip"></ipv6></interface><interface><name>GigabitEthernet0/0/0</name><type
xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type><enabled>true</enabled><ipv4
xmlns="urn:ietf:params:xml:ns:yang:ietf-ip"></ipv4><ipv6
xmlns="urn:ietf:params:xml:ns:yang:ietf-
ip"></ipv6></interface><interface><name>GigabitEthernet0/0/1</name><description>Link
to PC</description><type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type><enabled>true</enabled><ipv4
xmlns="urn:ietf:params:xml:ns:yang:ietf-
ip"><address><ip>192.168.1.1</ip><netmask>255.255.255.0</netmask></address></ipv4><ipv
6 xmlns="urn:ietf:params:xml:ns:yang:ietf-
ip"></ipv6></interface></interfaces></data></rpc-reply>]]>]]>

c. Sao chép XML đã được trả về, nhưng không bao gồm các ký tự cuối cùng]]>]]>. Các ký tự này không phải là một phần của XML được bộ định tuyến trả về.

d. Tìm kiếm trên internet để cải thiện XML dễ nhìn hơn. Tìm một trang web phù hợp và sử dụng công cụ của nó để chuyển đổi XML sang định dạng dễ đọc hơn, chẳng hạn như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="103">
   <data>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
         <interface>
            <name>GigabitEthernet0</name>
            <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
            <enabled>true</enabled>
            <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip" />
            <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip" />
         </interface>
         <interface>
            <name>GigabitEthernet0/0/0</name>
            <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
            <enabled>true</enabled>
            <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip" />
            <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip" />
         </interface>
         <interface>
            <name>GigabitEthernet0/0/1</name>
            <description>Link to PC</description>
            <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-
type">ianaift:ethernetCsmacd</type>
            <enabled>true</enabled>
            <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
               <address>
                  <ip>192.168.1.1</ip>
                  <netmask>255.255.255.0</netmask>
               </address>
            </ipv4>
```

```
            <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip" />
          </interface>
        </interfaces>
      </data>
    </rpc-reply>
```

Step 5: Đóng phiên NETCONF.

Để đóng phiên NETCONF, client cần gửi tin nhắn RPC như sau:

```
<rpc message-id="9999999" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <close-session />
</rpc>
```

Chúng ta sẽ được đưa trở lại dấu nhắc terminal. Quay trở lại dấu nhắc của router và hiển thị các phiên netconf đang mở. Chúng ta sẽ thấy rằng phiên đã được đóng lại.

Part 3: Sử dụng ncclient để kết nối NETCONF

Làm việc với NETCONF không yêu cầu làm việc với các thông điệp RPC và XML thô của NETCONF. Trong Phần 3, bạn sẽ tìm hiểu cách sử dụng mô-đun Python ncclient để dễ dàng tương tác với các thiết bị mạng bằng NETCONF. Ngoài ra, bạn sẽ tìm hiểu cách xác định mô hình YANG nào được thiết bị hỗ trợ. Thông tin này hữu ích khi xây dựng hệ thống tự động hóa mạng sản xuất yêu cầu các mô hình YANG cụ thể được hỗ trợ bởi thiết bị mạng đã cho.

Step 1: Kiểm tra thư viện nccleint đã được cài đặt và sẵn sàng để sử dụng.

Nhập lệnh danh sách pip3 --format = cột để xem tất cả các mô-đun Python hiện được cài đặt trong CCNP VM. Đầu ra của bạn có thể khác với sau đây. Nhưng bạn sẽ thấy ncclient được liệt kê, như được hiển thị. Nếu không, sử dụng lệnh sudo pip3 install ncclient để cài đặt nó.

```
student@CCNP:~$ pip3 list --format=columns
Package               Version
--------------------- -------------------
apturl                0.5.2
asn1crypto            0.24.0
(output omitted)
ncclient              0.6.7
netifaces             0.10.4
netmiko               3.0.0
oauth                 1.0.1
olefile               0.45.1
paramiko              2.7.1
pexpect               4.2.1
Pillow                5.1.0
pip                   9.0.1
(output omitted)
requests              2.22.0
requests-unixsocket   0.1.5
scp                   0.13.2
(output omitted)
xmltodict             0.12.0
zope.interface        4.3.2
```

Step 2: Tạo script sử dụng ncclient để kết nối đến dịch vụ NETCONF.

Mô đun ncclient cung cấp một lớp trình quản lý với phương thức connect () để thiết lập các kết nối NETCONF từ xa. Sau khi kết nối thành công, đối tượng được trả về biểu thị kết nối NETCONF cho thiết bị từ xa.

a. Trong Python IDLE, tạo một tệp script Python mới được gọi là **ncclient-netconf.py**.

b. Trong trình soạn thảo script Python mới, nhập lớp quản lý từ mô đun ncclient. Sau đó tạo một biến m để biểu diễn phương thức connection (). connection() bao gồm tất cả các thông tin được yêu cầu để kết nối với dịch vụ NETCONF chạy trên R1. Lưu ý rằng cổng là 830 cho NETCONF.

```
from ncclient import manager

m = manager.connect(
    host="192.168.1.1",
    port=830,
    username="cisco",
    password="cisco123!",
    hostkey_verify=False
    )
```

Nếu hostkey_verify được đặt thành True, R1 sẽ yêu cầu bạn xác minh dấu vân tay SSH. Trong lab này, chúng ta sẽ đặt giá trị này thành False, như đã thực hiện ở trên.

c. Lưu và chạy chương trình để xác minh rằng không có lỗi. Bạn sẽ không thấy bất kỳ đầu ra nào. Nhưng bạn có thể xác minh rằng phiên NETCONF đang hoạt động trên R1 bằng cách nhập lệnh hiển thị phiên netconf-yang.

```
================= RESTART: /home/student/ncclient-netconf.py =================
>>>
```

Step 3: Thêm chức năng in vào script để capabilities NETCONF cho R1 được liệt kê..

Đối tượng m được trả về bởi hàm manager.connect () đại diện cho phiên từ xa NETCONF. Như bạn đã thấy trong Phần 2, trong mỗi phiên NETCONF, trước tiên máy chủ sẽ gửi danh sách capabilities của nó là một danh sách, ở định dạng XML của các mô hình YANG được hỗ trợ. Với mô đun ncclient, danh sách capabilities nhận được được lưu trữ trong danh sách m.server_capabilities.

a. Sử dụng vòng lặp **for** và chức năng in để hiển thị capabilities của thiết bị:

```
print("#Supported Capabilities (YANG models):")
for capability in m.server_capabilities:
    print(capability)
```

b. Lưu và chạy chương trình. Đầu ra là cùng một đầu ra mà bạn nhận được từ việc gửi tin nhắn xin chào phức tạp trong Part 2, Step 3, nhưng không có thẻ XML mở và đóng <capability> trên mỗi dòng.

```
================= RESTART: /home/student/ncclient-netconf.py =================
#Supported Capabilities (YANG models):
urn:ietf:params:netconf:base:1.0
urn:ietf:params:netconf:base:1.1
urn:ietf:params:netconf:capability:writable-running:1.0
urn:ietf:params:netconf:capability:xpath:1.0
urn:ietf:params:netconf:capability:validate:1.0
urn:ietf:params:netconf:capability:validate:1.1
urn:ietf:params:netconf:capability:rollback-on-error:1.0
```

```
urn:ietf:params:netconf:capability:notification:1.0
Urn:ietf:params:netconf:capability:interleave:1.0
<output omitted>
```

Part 4: Sử dụng ncclient để lấy cấu hình

Trong Phần 4, sử dụng NETCONF ncclient để truy xuất cấu hình của R1, cập nhật cấu hình và tạo giao diện mới. Chúng ta cũng sẽ tìm hiểu lý do tại sao hỗ trợ giao dịch của NETCONF lại quan trọng để nhận được các thay đổi mạng nhất quán.

Step 1: Sử dụng hàm get_config () để truy xuất cấu hình đang chạy cho R1.

a. Chúng ta có thể sử dụng phương thức **get_config ()** của đối tượng phiên m NETCONF để truy xuất cấu hình cho R1. Phương thức **get_config ()** cần chuỗi nguồn thông số chỉ định nguồn kho dữ liệu NETCONF . Sử dụng chức năng in để hiển thị kết quả. Kho dữ liệu NETCONF duy nhất hiện có trên R1 là kho dữ liệu đang chạy. Có thể xác minh điều này bằng lệnh **show netconf-yang datastores**. Nếu muốn bỏ qua việc hiển thị đầu ra từ Phần 3, hãy đánh dấu comment khối câu lệnh in capabilities, như được hiển thị trong phần sau:

```
'''
print("#Supported Capabilities (YANG models):")
for capability in m.server_capabilities:
    print(capability)
'''

netconf_reply = m.get_config(source="running")
print(netconf_reply)
```

b. Lưu và chạy chương trình của bạn. Đầu ra sẽ có hơn 100 dòng, vì vậy IDLE có thể nén chúng. Bấm đúp vào Squeezed text trong cửa sổ IDLE shell để mở rộng đầu ra.

```
================= RESTART: /home/student/ncclient-netconf.py =================
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-
id="urn:uuid:b146b877-ba9a-45bc-8219-31732d1708dd"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"><data><native
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native"><version>16.9</version><boot-
start-marker/><boot-end-
marker/><service><timestamps><debug><datetime><msec></msec></datetime></debug><log><da
tetime><msec/></datetime></log></timestamps></service><hostname>R1</hostname>
<output omitted>
```

c. Lưu ý rằng XML trả về không được định dạng. Bạn có thể sao chép nó ra cùng một trang mà bạn đã tìm thấy trong Phần 2 để cải tiến XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:b146b877-ba9a-
45bc-8219-31732d1708dd">
   <data>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
         <version>16.9</version>
         <boot-start-marker />
         <boot-end-marker />
         <service>
```

```
            <timestamps>
               <debug>
                  <datetime>
                     <msec />
                  </datetime>
               </debug>
               <log>
                  <datetime>
                     <msec />
                  </datetime>
               </log>
            </timestamps>
         </service>
         <hostname>R1</hostname>
<output omitted>
```

Step 2: Sử dụng Python để định dạng XML dễ nhìn hơn.

Python đã tích hợp hỗ trợ để làm việc với các tệp XML. Mô-đun xml.dom.minidom có thể được sử dụng để làm đẹp đầu ra với hàm toprettyxml().

a. Khi bắt đầu tập lệnh của bạn, hãy thêm một câu lệnh để nhập mô-đun xml.dom.minidom. Sau đó, thay thế chức năng in đơn giản print(netconf_reply) bằng phiên bản in đầu ra XML được chỉnh sửa.

```
import xml.dom.minidom
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```

b. Lưu và chạy chương trình. Mở rộng đầu ra để xem XML được hiển thị ở định dạng dễ đọc hơn.

```
================= RESTART: /home/student/ncclient-netconf.py =================
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:0fc605b5-bf01-44c2-9830-1c092db3afa7"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
        <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
            <version>16.9</version>
            <boot-start-marker/>
            <boot-end-marker/>
            <service>
                <timestamps>
                    <debug>
                        <datetime>
                            <msec/>
                        </datetime>
                    </debug>
                    <log>
                        <datetime>
                            <msec/>
                        </datetime>
                    </log>
                </timestamps>
```

```
            </service>
            <hostname>R1</hostname>
```

Step 3: Sử dụng bộ lọc filter với get_config() để chỉ lấy mô hình YANG chỉ định.

Quản trị viên mạng chỉ có thể muốn truy xuất một phần cấu hình đang chạy trên thiết bị. NETCONF chỉ hỗ trợ trả về dữ liệu được xác định trong tham số bộ lọc của hàm get_conf().

a. Tạo một biến có tên là netconf_filter chỉ lấy dữ liệu được xác định bởi mô hình Cisco IOS XE Native YANG.

```
netconf_filter = """
<filter xmlns="urn:ietf:params:xml:ns:netconf:base:1.0>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native" />
</filter>
"""
netconf_reply = m.get_config(source="running", filter=netconf_filter)
```

b. Lưu và chạy chương trình của bạn. Mở rộng đầu ra để xem XML được hiển thị ở định dạng dễ đọc hơn. Sự bắt đầu của đầu ra là như nhau, như được hiển thị dưới đây. Tuy nhiên, phần còn lại của đầu ra chỉ bao gồm các mô hình YANG được chỉ định.

```
================= RESTART: /home/student/ncclient-netconf.py =================
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:a3e46a28-07b4-4e8d-964c-647cb565e7a1"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
        <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
            <version>16.9</version>
            <boot-start-marker/>
            <boot-end-marker/>
            <service>
                <timestamps>
                    <debug>
                        <datetime>
                            <msec/>
                        </datetime>
                    </debug>
                    <log>
                        <datetime>
                            <msec/>
                        </datetime>
                    </log>
                </timestamps>
            </service>
            <hostname>R1</hostname>
(output omitted)
```

c. Lọc dữ liệu được truy xuất để chỉ hiển thị mô-đun YANG gốc làm giảm đáng kể đầu ra của bạn. Điều này là do mô-đun YANG riêng chỉ bao gồm một tập hợp con của tất cả các mô hình Cisco IOX XE YANG.

Part 5: Sử dụng ncclient để cấu hình thiết bị

Trong Phần 5, bạn sẽ sử dụng **ncclient** để cấu hình R1 bằng phương thức **edit_config()** của mô đun quản lý.

Step 1: Sử dụng ncclient để chỉnh sửa hostname cho R1.

Ghi chú: Sẽ ổn nếu nhiều học viên truy cập cùng một lúc và thay đổi tên máy chủ. Bạn sẽ tìm kiếm phản hồi <ok /> từ NETCONF để xác nhận cấu hình của bạn đã được gửi và áp dụng thành công. Tên máy chủ hiện tại không quan trọng.

a. Để cập nhật cài đặt hiện có trong cấu hình cho R1, bạn có thể trích xuất vị trí cài đặt từ cấu hình được truy xuất trong Phần 4. Đối với bước này, bạn sẽ đặt một biến để thay đổi giá trị <hostname>.

```
================= RESTART: /home/student/ncclient-netconf.py =================
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:a3e46a28-07b4-4e8d-964c-647cb565e7a1"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
        <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
(output omitted)
            <hostname>R1</hostname>
(output omitted)
```

b. Trước đây, bạn đã xác định biến <filter>. Để sửa đổi cấu hình thiết bị, bạn sẽ xác định biến <config>. Thêm biến sau vào tập lệnh ncclient_netconf.py của bạn. Bạn có thể sử dụng NEWHOSTNAME hoặc bất kỳ tên máy chủ nào bạn muốn.

```
netconf_hostname = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0”>
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
     <hostname>NEWHOSTNAME</hostname>
  </native>
</config>
"""
```

c. Sử dụng hàm edit_config() của đối tượng phiên m NETCONF để gửi cấu hình và lưu trữ các kết quả trong biến netconf_reply để chúng có thể được in. Các tham số cho hàm edit_config() như sau:

- **target** – đối tượng kho lưu trữ NETCONF cần được cập nhật.
- **config** – cấu hình chỉnh sửa cần được gửi

```
netconf_reply = m.edit_config(target="running", config=netconf_hostname)
```

d. Hàm edit_config () trả về thông báo trả lời RPC XML với <ok /> cho biết thay đổi đã được áp dụng thành công. Lặp lại câu lệnh in trước để hiển thị kết quả.

```
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```

e. Lưu và chạy chương trình của bạn. Bạn sẽ nhận được đầu ra tương tự như đầu ra được hiển thị dưới đây. Bạn cũng có thể xác minh rằng tên máy chủ cho R1 đã thay đổi bằng cách truy cập R1 CLI.

```
================= RESTART: /home/student/ncclient-netconf.py =================
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:8d8e1f04-1aa4-47c9-92d5-2a8981322f8f"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```
    <ok/>
</rpc-reply>

>>>
```

f. Thay đổi hostname trở về lại R1.

Step 2: Sử dụng ncclient để tạo loopback interface mới trên R1.

a. Đánh dấu commnet code từ bước trước nếu bạn muốn tránh thay đổi hostname một lần nữa.

b. Tạo một biến <config> mới để giữ cấu hình cho giao diện loopback mới. Thêm phần sau vào tập lệnh ncclient_netconf.py của bạn. Nếu nhiều học viên đang truy cập R1 cùng một lúc, hãy sử dụng loopback được chỉ định bởi người hướng dẫn. Nếu không, bạn có thể sử dụng loopback 1, như hiển thị bên dưới.

Ghi chú: Thay thế [Tên sinh viên] bằng tên của bạn. Để lại dấu gạch chéo ngược (\) trong lệnh mô tả. Ký tự thoát, dấu gạch chéo ngược (\), cho phép giải thích thay thế cho trích dẫn đơn (') là dấu nháy đơn. Các ký tự thoát cho phép các ký tự đặc biệt được sử dụng bên trong khai báo chuỗi. Trích dẫn duy nhất thường được sử dụng để ghi chú bắt đầu hoặc kết thúc khai báo chuỗi.

```
netconf_loopback = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0”>
 <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface>
   <Loopback>
    <name>1</name>
    <description>[Student Name]\'s loopback</description>
    <ip>
     <address>
      <primary>
       <address>10.1.1.1</address>
       <mask>255.255.255.0</mask>
      </primary>
     </address>
    </ip>
   </Loopback>
  </interface>
 </native>
</config>
"""
```

c. Nhận xét biến netconf_repply trước đó nếu bạn muốn tránh thay đổi tên máy chủ một lần nữa. Thêm chức năng edit_config() sau vào ncclient_netconf.py của bạn để gửi cấu hình loopback mới tới R1 và sau đó in ra kết quả.

```
netconf_reply = m.edit_config(target="running", config=netconf_loopback)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```

d. Lưu và chạy chương trình. Chúng ta có thể lấy được kết quả đầu ra như bên dưới:

```
================ RESTART: /home/student/ncclient-netconf.py ================
<?xml version="1.0" ?>
```

```
<rpc-reply message-id="urn:uuid:6407b416-2851-4eca-bbfa-7afbc6e8fbcf"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>

>>>
```

e. Trên R1, kiểm tra loopback interface đã được tạo.

```
R1# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES unset  down                  down
GigabitEthernet0/0/1   192.168.1.1     YES manual up                    up
Serial0/1/0            unassigned      YES unset  up                    up
Serial0/1/1            unassigned      YES unset  up                    up
GigabitEthernet0       unassigned      YES unset  down                  down
Loopback1              10.1.1.1        YES other  up                    up
```

Step 3: Cố gắng tạo giao diện loopback mới có cùng địa chỉ IPv4.

a. Tạo một biến mới gọi là netconf_newloop. Nó sẽ giữ một cấu hình tạo giao diện loopback 2 mới nhưng có cùng địa chỉ IPv4 như trên loopback 1: 10.1.1.1./24. Tại router CLI, điều này sẽ tạo ra lỗi do cố gắng gán địa chỉ IP trùng lặp cho cổng.

Ghi chú: Nếu nhiều sinh viên đang truy cập cùng một lúc, hãy tăng số hiệu của loopback thêm 1 giá trị được chỉ định bởi người hướng dẫn của bạn. Nếu không, bạn có thể sử dụng loopback 2, như hiển thị bên dưới.

```
netconf_newloop = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0>
 <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface>
   <Loopback>
    <name>2</name>
    <description>[Student Name]'s loopback</description>
    <ip>
     <address>
      <primary>
       <address>10.1.1.1</address>
       <mask>255.255.255.0</mask>
      </primary>
     </address>
    </ip>
   </Loopback>
  </interface>
 </native>
</config>
"""
```

b. Thêm chức năng edit_config () sau vào ncclient_netconf.py của bạn để gửi cấu hình loopback mới tới R1. Bạn không cần một câu lệnh in cho bước này.

```
netconf_reply = m.edit_config(target="running", config=netconf_newloop)
```

c. Lưu và chạy chương trình. Bạn sẽ nhận được đầu ra lỗi tương tự như sau:

```
============ RESTART: /home/student/Documents/ncclient-netconf.py ============
Traceback (most recent call last):
  File "/home/student/Documents/ncclient-netconf.py", line 71, in <module>
    netconf_reply2=m.edit_config(target='running', config = netconf_newloop)
  File "/home/student/.local/lib/python3.6/site-packages/ncclient/manager.py", line
236, in execute
     huge_tree=self._huge_tree).request(*args, **kwds)
  File "/home/student/.local/lib/python3.6/site-packages/ncclient/operations/edit.py",
line 69, in request
     return self._request(node)
ncclient.operations.rpc.RPCError: inconsistent value: Device refused one or more
commands
>>>
```

d. Không giống như sử dụng netmiko hoặc sao chép và dán script, NETCONF sẽ không áp dụng bất kỳ cấu hình nào được gửi nếu một hoặc nhiều lệnh bị từ chối. Để xác minh điều này, hãy nhập lệnh show giao diện ip trên R1. Lưu ý rằng giao diện mới của bạn chưa được tạo.

```
R1# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES unset  down                  down
GigabitEthernet0/0/1   192.168.1.1     YES manual up                    up
Serial0/1/0            unassigned      YES unset  up                    up
Serial0/1/1            unassigned      YES unset  up                    up
GigabitEthernet0       unassigned      YES unset  down                  down
Loopback1              10.1.1.1        YES other  up                    up
```

Part 6: Chương trình được sử dụng trong lab này

Sau đây là chương trình hoàn chỉnh được tạo ra trong lab này. Code từ Part 5, Step 3 được đánh dấu comment để bạn có thể chạy script mà không gặp lỗi. Script của bạn có thể trông khác nhau. Thực hành các kỹ năng Python của bạn bằng cách sửa đổi chương trình để gửi các lệnh xác minh và cấu hình khác nhau.

```
from ncclient import manager
import xml.dom.minidom

m = manager.connect(
    host="192.168.1.1",
    port=830,
    username="cisco",
    password="cisco123!",
    hostkey_verify=False
    )

print("#Supported Capabilities (YANG models):")
```

```
for capability in m.server_capabilities:
    print(capability)


netconf_reply = m.get_config(source="running")
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())

netconf_filter = """
<filter xmlns="urn:ietf:params:xml:ns:netconf:base:1.0”>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native" />
</filter>
"""
netconf_reply = m.get_config(source="running", filter=netconf_filter)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())

netconf_hostname = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0”>
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
     <hostname>NEWHOSTNAME</hostname>
  </native>
</config>
"""

netconf_reply = m.edit_config(target="running", config=netconf_hostname)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())

netconf_loopack = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0”>
 <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface>
   <Loopback>
    <name>1</name>
    <description>[Student Name]'s loopback</description>
    <ip>
     <address>
      <primary>
       <address>10.1.1.1</address>
       <mask>255.255.255.0</mask>
      </primary>
     </address>
    </ip>
   </Loopback>
  </interface>
 </native>
```

```
</config>
"""

netconf_reply = m.edit_config(target="running", config=netconf_loopack)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())

'''
netconf_newloop = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0”>
 <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface>
   <Loopback>
    <name>2</name>
    <description>[Student Name]’s loopback</description>
    <ip>
     <address>
      <primary>
       <address>10.1.1.1</address>
       <mask>255.255.255.0</mask>
      </primary>
     </address>
    </ip>
   </Loopback>
  </interface>
 </native>
</config>
"""

netconf_reply = m.edit_config(target="running", config=netconf_newloop)
'''
```

Bảng tổng hợp cổng Router

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: Để tìm hiểu cách cấu hình router, hãy xem các interface để xác định loại router và số lượng cổng của router. Không có cách nào để liệt kê hiệu quả tất cả các kết hợp cấu hình cho từng lớp router. Bảng này bao gồm các mã định danh cho các kết hợp cổng Ethernet và Serial có thể có trong thiết bị. Bảng không bao gồm bất kỳ loại giao diện nào khác, mặc dù một router cụ thể có thể chứa một loại interface. Một ví dụ về điều này có thể là cổng ISDN BRI. Chuỗi trong ngoặc đơn là tên viết tắt hợp pháp có thể được sử dụng trong các lệnh Cisco IOS để thể hiện cổng.