

## LAB REPORT

LAB TITLE: Designing Network Topology with GNS3

Name: Gabriel Miracle Nwauche

PSU ID: [GMN5192@psu.edu](mailto:GMN5192@psu.edu)

Course: IST 220

Instructor: Nick Giacobe

Date: 3/29/2023

## **General Context**

This particular lab was difficult just because of the countless configurations the user had to go through, but first let's talk about what went well. Drawing the network topology simulation was the core of the lab. It was so important that all of the interfaces and adapters were correct because once the user had to go to the PuTTy application and configure the commands, many mistakes could have been made. For example, on one of the instructions in section 2, when I tried to "any" IP address (using 0.0.0.0/0) to their destination using one of the Default Gateways, the result constantly gave me a warning every attempt. At that point the user has to resort to trial and error, which is inevitable when it comes like that in the real world.

During the final stages of the lab, this was where the real test began. Managing the switch ports was not very difficult, the user would just have to move around a few connections in the topology. However, when it came to rearranging the ports on the bridge, this required prior knowledge from the previous steps in the lab using the Open vSwitch commands. The last portion was the most challenging just because of the login via SSH, which was difficult to find in the first place, followed by the ping which became unreachable.

## **Technical Context**

The construction and the meticulousness of the topology was the core and the most crucial part of the lab. In fact basically the entirety of the lab was derived from the network topology in GNS3. The design and development called for different devices to be used like switches and routers. The first model involved the switches, which were connected through ethernet ports and those ethernet ports were connected to the VPCs as well. The second model was much of the same, instead it dealt with the routers instead of using switches, which in turn made the configurations in the Solar-PuTTy application more complex. Many times I had to start from the beginning of section 2 and reconfigure the commands using *conf t* then connecting the interfaces to their designated IP addresses. Even though that part proved to be difficult, it sure made the user more knowledgeable about the differences between the Layer 2 and Layer 3 switch tools.

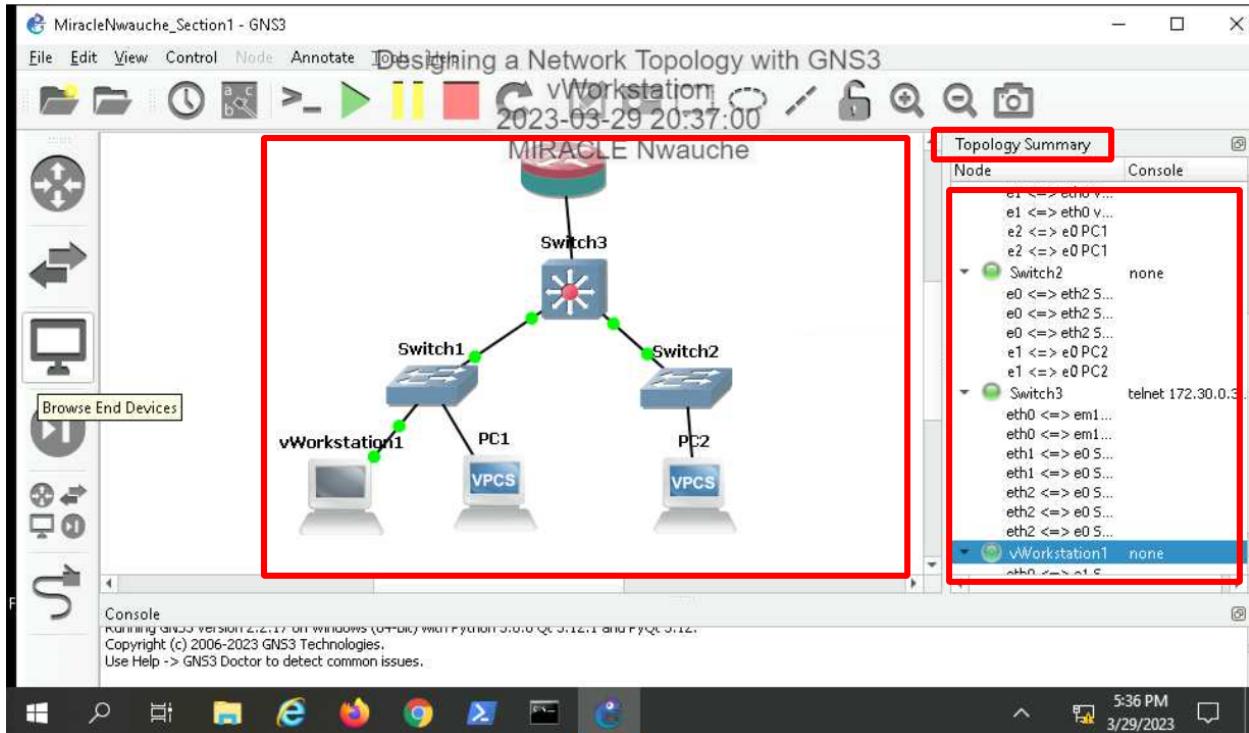
The learning curve from section 2 to section 3 was a drastic "jump." Not entirely because the content was difficult but also because the prior commands the user had to use in the lab required the opposite for section 3. For example, the Open vSwitch caused trouble for me when it was time to delete the connection of interface *eth0*. Once that was solved, more complexity was thrown into the mix, the user would now have to log into the root user via SSH. My first thought was to use a *sudo* command, but the truth was that the SSH was the core of that command. I then figured out how to use *ssh root@(ip address given)* to solve that heinous problem. After the succession of that obstacle, the user is met with a (yes/no) message, which I

continued to press enter for until realizing that the system wanted my input to login. At least it gave the user a nice message saying “Happy Configuring!”

## SECTION 1, PART 1: Configure Physical Connectivity on a Layer 2 Network

25. Make a screen capture showing the completed topology and the active nodes and interfaces displayed in the Topology Summary.

### SCREEN CAPTURES



**Figure 1 Section 1, Part 1:** Captured here are the nodes with all of their connections and topologies relating to each other using the ethernet. On the right is the Topology summary showing the active connections in the topology tree.

## SECTION 1, PART 2: Configure logical connectivity on a Layer 2 Network

8. Make a screen capture showing the interface configuration on PC1.



The screenshot shows a Solar-PuTTY window titled "MiracleNwauche\_Section1 - GNS3". The title bar also includes "Designing a Network | Topology with GNS3" and "Workstation". The window displays a terminal session for "PC1". The session output is as follows:

```
Executing the startup file
2023-03-30 14:27:39
MIRACLE Nwauche

PC1> ip 172.30.0.10
Checking for duplicate address...
PC1 : 172.30.0.10 255.255.255.0

PC1> ip 172.30.0.10/24
Checking for duplicate address...
PC1 : 172.30.0.10 255.255.255.0

PC1> show
NAME      IP/MASK        GATEWAY        MAC          LPORT    RHOST:PORT
PC1       172.30.0.10/24   0.0.0.0        00:50:79:66:68:00  20012   127.0.0.1:20013
fe80::250:79ff:fe66:6800/64

PC1> save
Saving startup configuration to startup.vpc
done

PC1>
```

A red box highlights the command "PC1> ip 172.30.0.10/24" and the resulting configuration table. Another red box highlights the "save" command and its confirmation message.

**Figure 2 Section 1, Part 2:** The configured commands are as shown in the highlighted boxes. Once the PCs are consoled using the GNS3 device, a PuTTy application opens where command lines can be transmuted. We type in the IP address to make sure there are no duplicated addresses, show the configuration of the device, then save it to the desktop.

10. Make a screen capture showing the interface configuration on PC2.

The screenshot shows a Solar-PuTTY terminal window titled "MiracleNwauche\_Section1 - GNS3". The window title bar also includes "Designing a Network | Topology with GNS3" and "vWorkstation". The status bar at the bottom right shows the date and time: "2023-03-30 14:33:16" and "MIRACLE Nwauche".

The terminal window displays the following command-line session:

```
PC2> ip 172.30.0.20/24
Checking for duplicate address...
PC2 : 172.30.0.20 255.255.255.0
PC2> show
NAME      IP/MASK      GATEWAY      MAC          LPORT      RHOST:PORT
PC2       172.30.0.20/24  0.0.0.0      00:50:79:66:68:01  20014     127.0.0.1:20015
fe80::250:79ff:fe66:6801/64
PC2> save
Saving startup configuration to startup.vpc
, done
PC2>
```

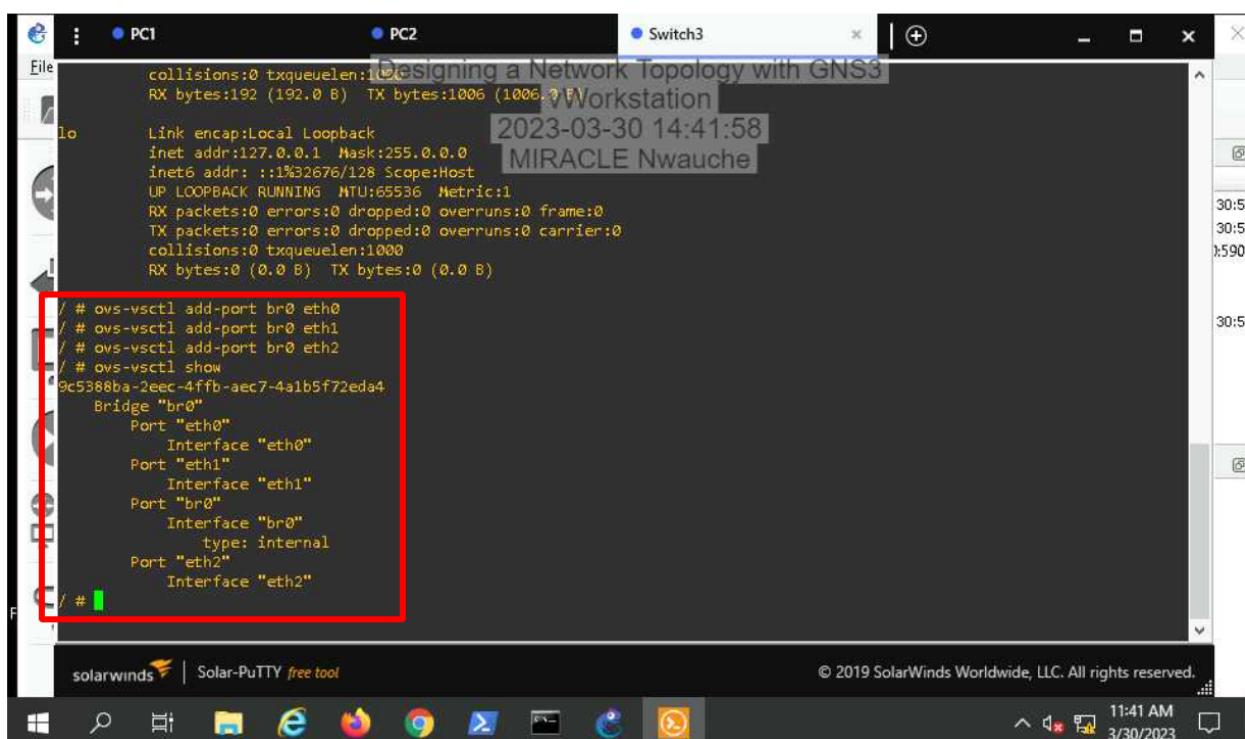
Two specific lines of the session are highlighted with red boxes:

- The first line, "PC2> ip 172.30.0.20/24", followed by its output "Checking for duplicate address..." and "PC2 : 172.30.0.20 255.255.255.0".
- The second line, "PC2> save", followed by its output "Saving startup configuration to startup.vpc" and ", done".

The Solar-PuTTY interface includes a sidebar with icons for file operations and a footer with the SolarWinds logo and copyright information: "© 2019 SolarWinds Worldwide, LLC. All rights reserved."

**Figure 3 Section 1, Part 2:** Much like the information in the previous capture, we do the same functions in the command line to get the same results. This time the IP address is not the same, so it will *show* a different configuration than the one in the prior PC device.

18. Make a screen capture showing the bridge configuration on Switch3.



The screenshot shows a Solar-PuTTY terminal window titled "Switch3". The terminal displays the output of the "ovs-vsctl" command. A red box highlights the portion of the output where the bridge "br0" is defined and its ports "eth0", "eth1", and "eth2" are listed. The terminal window is part of a larger interface with tabs for "PC1", "PC2", and "Switch3". The status bar at the bottom shows the SolarWinds logo, the text "Solar-PuTTY free tool", the date "3/30/2023", and the time "11:41 AM".

```
collisions:0 txqueuelen:1000
RX bytes:192 (192.0 B) TX bytes:1006 (1006.0 B)
Workstation
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1%32676/128 Scope:Host
      UP LOOPBACK RUNNING MTU:65536 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frames:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

/ # ovs-vsctl add-port br0 eth0
/ # ovs-vsctl add-port br0 eth1
/ # ovs-vsctl add-port br0 eth2
/ # ovs-vsctl show
9c5388ba-2eec-4ffb-aec7-4a1b5f72eda4
  Bridge "br0"
    Port "eth0"
      Interface "eth0"
    Port "eth1"
      Interface "eth1"
    Port "br0"
      Interface "br0"
        type: internal
    Port "eth2"
      Interface "eth2"
/ #
```

**Figure 4 Section 1, Part 2:** The highlighted frame shows the *ovs-vsctl* configuration of the bridge that connects all of the *eth* ports together. Once the command is executed, using the *show* command the user will be able to see the Bridge followed by the Ports that connect to that bridge. The interface name will be shown too.

22. Make a screen capture showing the successful replies from PC1 and PC2 in the Command Prompt window.

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The window is titled "Designing a Network Topology with GNS3" and has tabs for "PC1", "PC2", and "Switch3". The command being run is "ping 172.30.0.10". The output shows four successful replies from the target IP address. Below this, another ping command is run to "172.30.0.20", also showing four successful replies. The ping statistics for both commands show 0% loss and low round-trip times. The taskbar at the bottom shows various icons and the date/time as 11:47 AM 3/30/2023.

```
C:\Users\Administrator>ping 172.30.0.10 2023-03-30 14:47:31
Pinging 172.30.0.10 with 32 bytes of data:
Reply from 172.30.0.10: bytes=32 time=2ms TTL=64
Reply from 172.30.0.10: bytes=32 time=1ms TTL=64
Reply from 172.30.0.10: bytes=32 time=1ms TTL=64
Reply from 172.30.0.10: bytes=32 time=1ms TTL=64

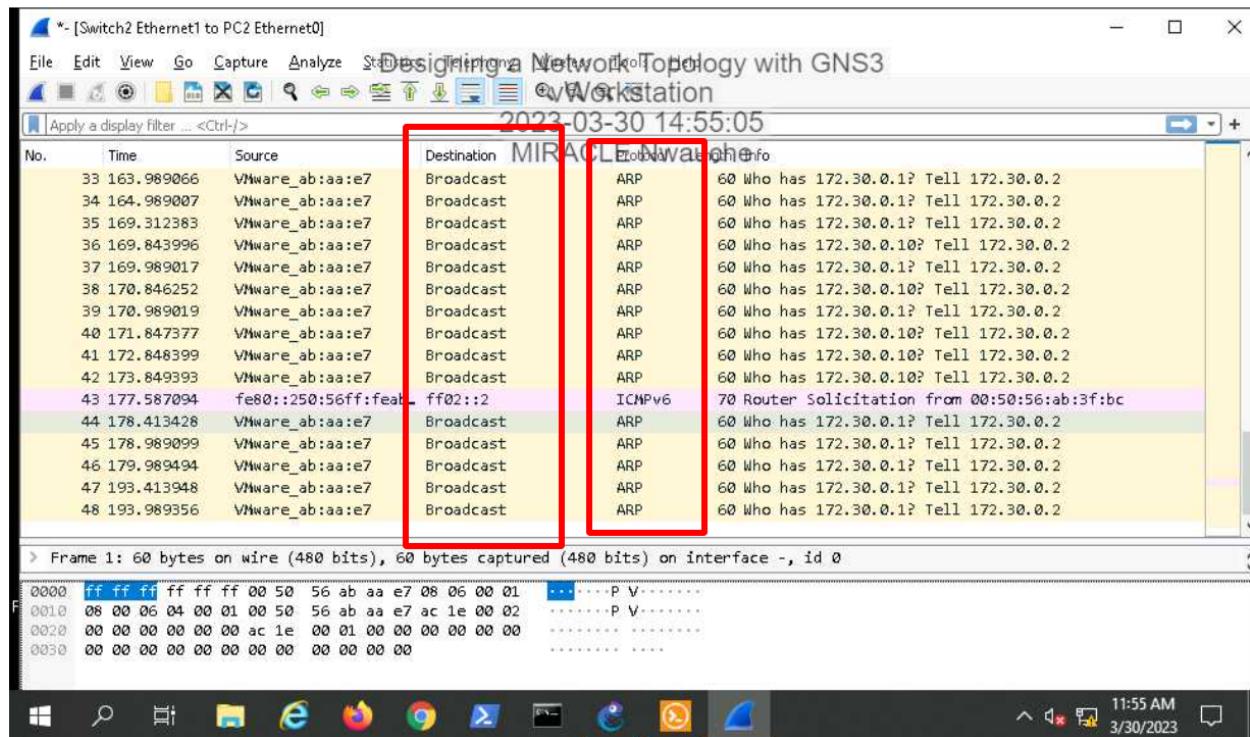
Ping statistics for 172.30.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\Administrator>ping 172.30.0.20
Pinging 172.30.0.20 with 32 bytes of data:
Reply from 172.30.0.20: bytes=32 time=4ms TTL=64
Reply from 172.30.0.20: bytes=32 time=1ms TTL=64
Reply from 172.30.0.20: bytes=32 time=1ms TTL=64
Reply from 172.30.0.20: bytes=32 time=2ms TTL=64

Ping statistics for 172.30.0.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 4ms, Average = 2ms
```

**Figure 5 Section 1, Part 2:** The next step was to go into the command prompt to see if the PCs were properly connected. To do this, the user had to ping the IP address to see if the replies were successful. From the highlighted spaces in the picture above, we can see that the ping was indeed successful, and the packet sourcing was intact because there was no packet loss.

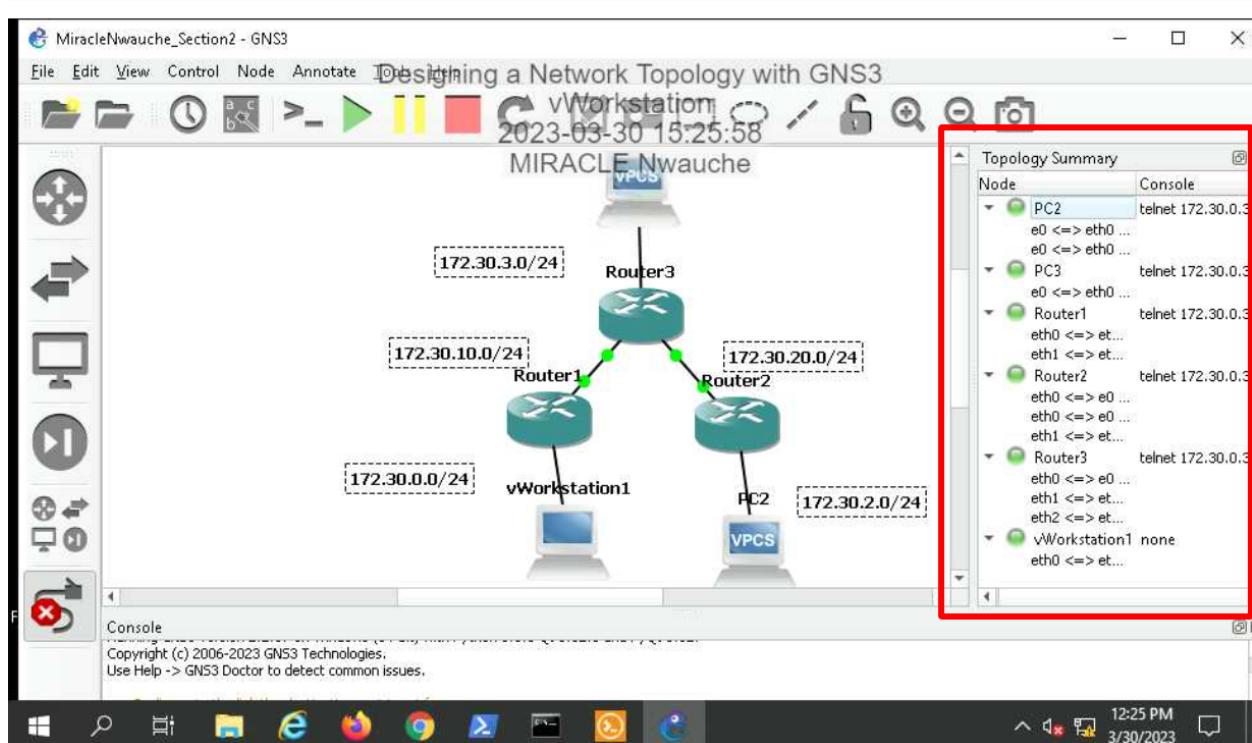
**31. Make a screen capture showing the ARP broadcast packets captured on the Switch2>PC2 link.**



**Figure 6 Section 1, Part 2:** As prompted in the lab instructions, the user had to go through the command prompt to see the arp configuration between the Switch2 device and the PC2 device. Since this connection was the farthest away from the two devices connected to Switch1, it is good to see that the ARP requests are the same, hence the reason this WireShark capture is important.

## SECTION 2: PART 2, Configure Physical Connectivity on a Layer 3 Network

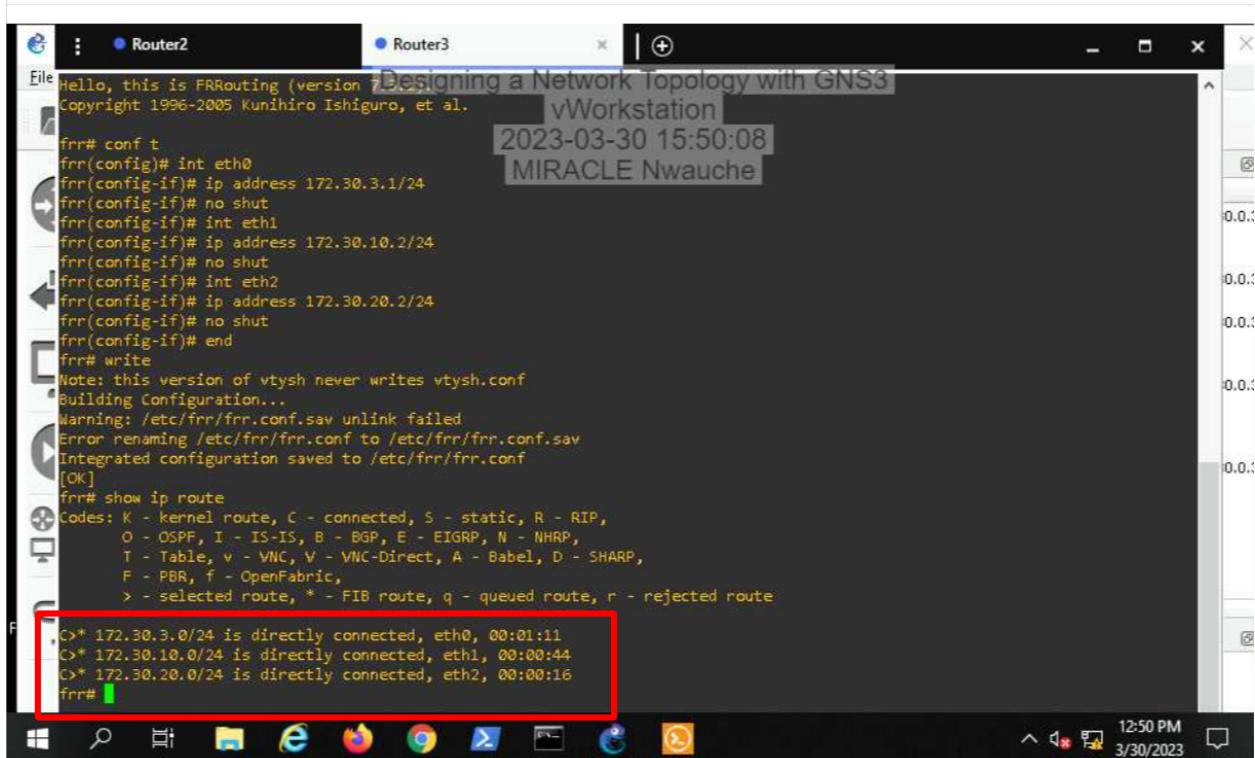
16. Make a screen capture showing the completed topology in the workspace, as well as the nodes and their links in the Topology Summary.



**Figure 7 Section 2, Part 1:** The network topology of the second network is shown above, this time with a little more information than the previous example. Instead of using the switches, the routers are substituted to show the connection of IP addresses between the devices shown. Once all of the system is up and running, the highlighted section to the right shows the connections and some information about the ports associated with those said connections.

## SECTION 2: PART 2: Configure the Logical Connectivity on a Layer 3 Network

13. Make a screen capture showing the routes currently known by Router3.



The screenshot shows a GNS3 vWorkstation window with two routers: Router2 and Router3. Router3 is active. The terminal window displays the configuration of Router3, including setting IP addresses on its three interfaces (eth0, eth1, eth2) and saving the configuration. It then shows the current IP routes:

```
frr# conf t
frr(config)# int eth0
frr(config-if)# ip address 172.30.3.1/24
frr(config-if)# no shut
frr(config-if)# int eth1
frr(config-if)# ip address 172.30.10.2/24
frr(config-if)# no shut
frr(config-if)# int eth2
frr(config-if)# ip address 172.30.20.2/24
frr(config-if)# no shut
frr(config-if)# end
frr# write
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Warning: /etc/frr/frr.conf.sav unlink failed
Error renaming /etc/frr/frr.conf to /etc/frr/frr.conf.sav
Integrated configuration saved to /etc/frr/frr.conf
[OK]
frr# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
      O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
      T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
      F - PBR, f - OpenFabric,
      > - selected route, * - FIB route, q - queued route, r - rejected route
C>* 172.30.3.0/24 is directly connected, eth0, 00:01:11
C>* 172.30.10.0/24 is directly connected, eth1, 00:00:44
C>* 172.30.20.0/24 is directly connected, eth2, 00:00:16
frr#
```

**Figure 8 Section 2, Part 2:** The IP route for the routers connected in the logically connected Layer 3 network are shown above. After configuring the IP addresses for each of the interfaces connected to all the routers, making sure they won't shut down (using the *no shut* command for that) as well. The C> shows that all of them are connected successfully.

26. Make a screen capture showing the traceroute attempt to the vWorkstation node.

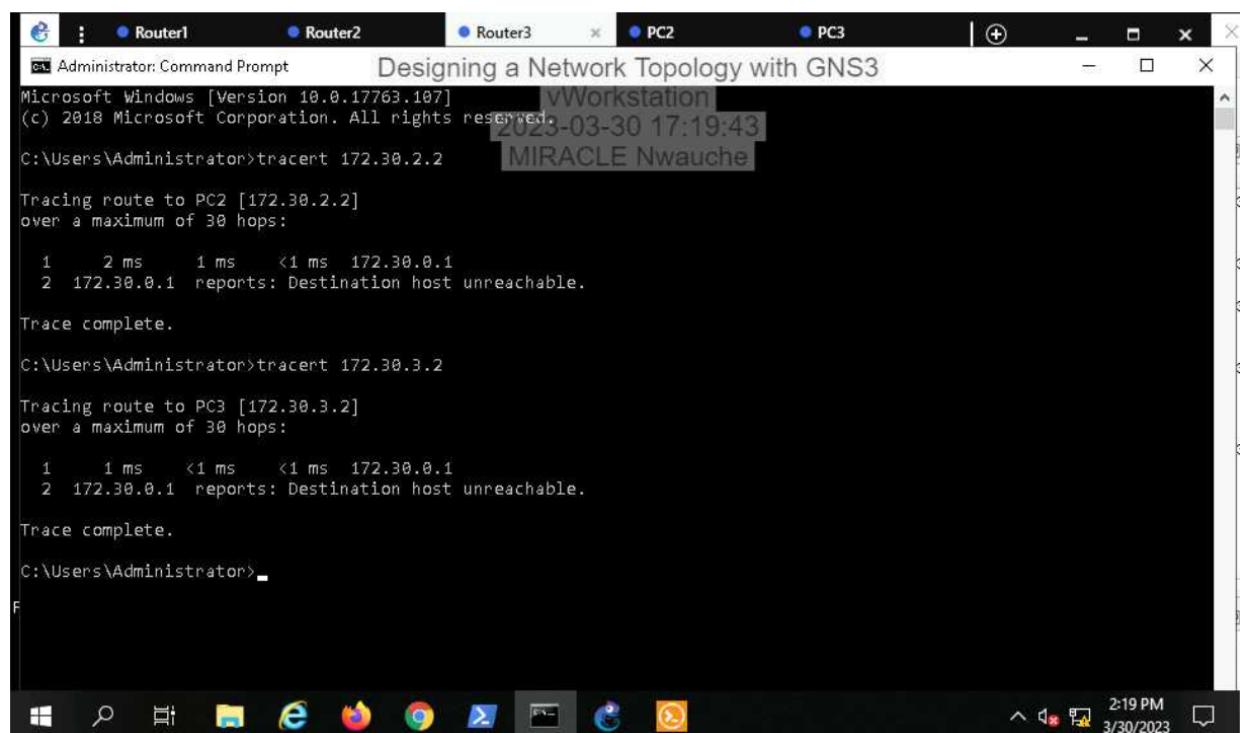
The screenshot shows a Windows desktop with a terminal window open. The title bar of the terminal window says "Building a Network Topology with GNS3". The terminal window displays the following command-line session:

```
frr(config)# ip route 0.0.0.0/0 172.30.0.2 vWorkstation
frr(config)# end
frr# write
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Warning: /etc/frr/frr.conf.sav unlink failed
Integrated configuration saved to /etc/frr/frr.conf
[OK]
frr# trace 172.30.3.2
traceroute to 172.30.3.2 (172.30.3.2), 30 hops max, 46 byte packets
1 172.30.20.2 (172.30.20.2) 3.320 ms 1.759 ms 0.776 ms
2 172.30.3.2 (172.30.3.2) 2.655 ms 2.178 ms 1.867 ms
frr# trace 172.30.0.2
traceroute to 172.30.0.2 (172.30.0.2), 30 hops max, 46 byte packets
1 172.30.20.2 (172.30.20.2) 1.286 ms !N 0.979 ms !N 0.863 ms !N
frr# conf t
frr(config)# ip route 0.0.0.0/0 172.30.20.2
frr(config)# end
frr# write
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
frr# trace 172.30.3.2
traceroute to 172.30.3.2 (172.30.3.2), 30 hops max, 46 byte packets
1 172.30.20.2 (172.30.20.2) 2.563 ms 0.720 ms 0.943 ms
2 172.30.3.2 (172.30.3.2) 2.561 ms 1.822 ms 1.746 ms
frr# trace 172.30.0.2
traceroute to 172.30.0.2 (172.30.0.2), 30 hops max, 46 byte packets
1 172.30.20.2 (172.30.20.2) 1.224 ms 1.454 ms 1.003 ms
2 172.30.20.2 (172.30.20.2) 3133.328 ms !H 3149.215 ms !H 3119.278 ms !H
frr#
```

A red box highlights the last two traceroute commands and their output. The terminal window is part of a GNS3 interface, with tabs for Router2, Router3, PC2, PC3, and Router1 visible at the top. The status bar at the bottom right shows the time as 1:14 PM and the date as 3/30/2023.

**Figure 9 Section 2, Part 2:** This capture is a little complex because of the way the traceroute worked with this specific command. Since the trace route for the second IP is only supposed to reach Router3 and then fail, there are exclamation points next to the time sensitive information after the command is configured.

**35. Make a screen capture showing the results from your tracert executions to PC2 and PC3.**



The screenshot shows a Windows Command Prompt window with the title "Designing a Network Topology with GNS3". The window contains the following text:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.
2023-03-30 17:19:43
C:\Users\Administrator>tracert 172.30.2.2
Tracing route to PC2 [172.30.2.2]
over a maximum of 30 hops:
1  2 ms    1 ms    <1 ms  172.30.0.1
2  172.30.0.1 reports: Destination host unreachable.

Trace complete.

C:\Users\Administrator>tracert 172.30.3.2
Tracing route to PC3 [172.30.3.2]
over a maximum of 30 hops:
1  1 ms    <1 ms    <1 ms  172.30.0.1
2  172.30.0.1 reports: Destination host unreachable.

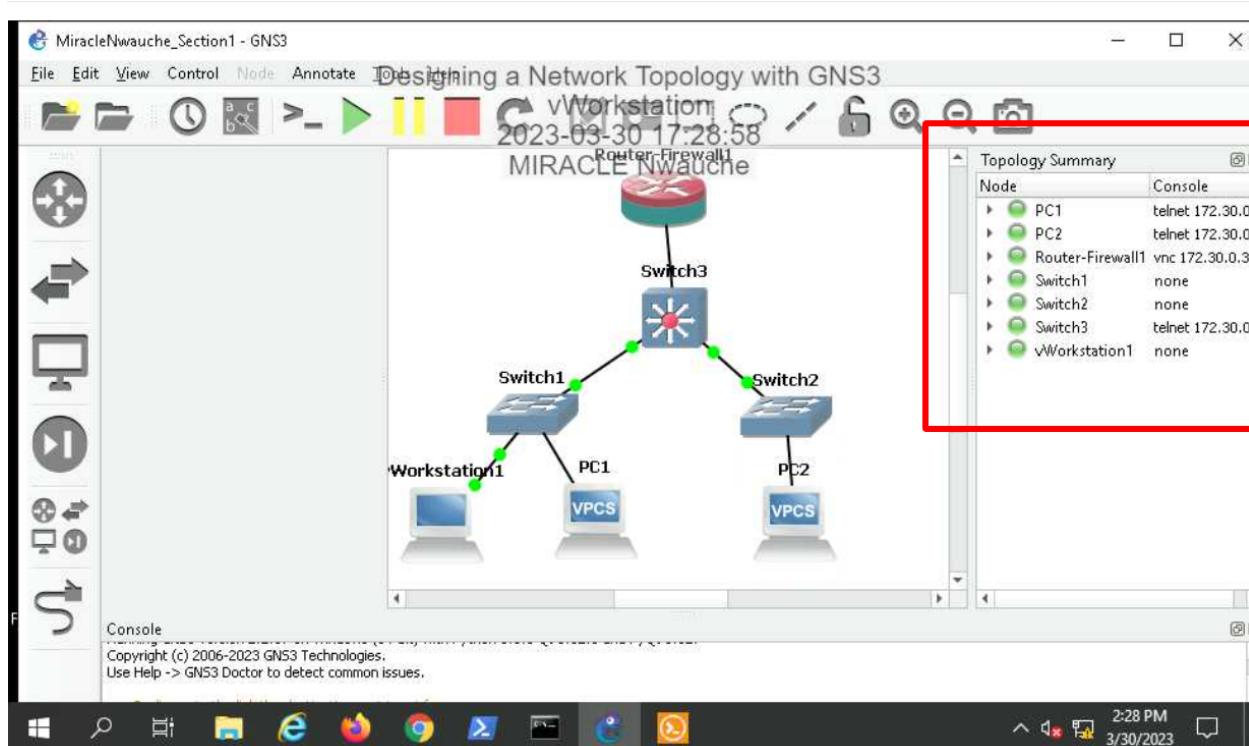
Trace complete.

C:\Users\Administrator>
```

**Figure 10 Section 2, Part 2:** These were my results from the traceroute. I'm not sure why but whenever I tried to configure the IP address with the Gateway 172.30.10.2/24 it would give me a warning. I'm guessing that is why the second destination is unreachable for both of the routers.

## SECTION 3: PART 3: Manage Switch Ports

Make a screen capture showing Switch3's connections in the Topology Summary.



**Figure 11 Section 3, Part 3:** Here is the network topology for the connection in Section 1. Switch3 with an extra adapter. After everything is powered on the right side of the screen shows that all of the connections are active. This topology shows that switch 3 is a four way connection for switches 1 and 2, with addition to the Firewall.

## PART 2: Rearrange Ports on a Bridge.

Make a screen capture showing the current Open vSwitch bridge configuration.

The screenshot shows a Solar-PuTTY terminal window titled "Designing a Network Topology with GNS3" with a timestamp of "2023-03-30 17:55:55" and a watermark for "MIRACLE Nwauche". The terminal session is running on a Windows host, as indicated by the taskbar at the bottom. The terminal content displays the following command-line session:

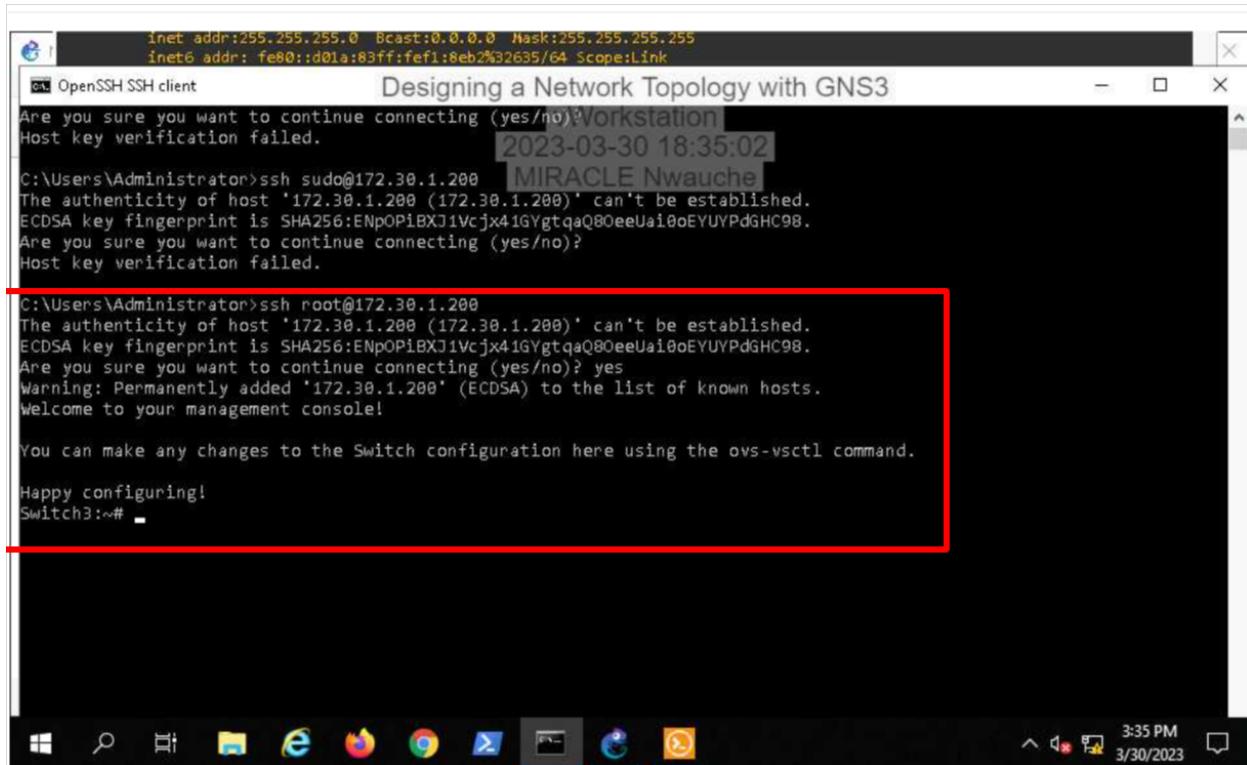
```
/bin/sh: no: not found
/ # ovs-vsctl show
9c5388ba-2ecc-4ffb-aec7-4a1b5f72eda4
  Bridge "br0"
    Port "eth0"
      Interface "eth0"
    Port "eth1"
      Interface "eth1"
    Port "br0"
      Interface "br0"
        type: internal
    Port "eth2"
      Interface "eth2"
/ # no br0 int eth0
/bin/sh: no: not found
/ # ovs-vsctl del-port br0 eth0
/ # ovs-vsctl add-port br0 eth3
/ # ovs-vsctl show
9c5388ba-2ecc-4ffb-aec7-4a1b5f72eda4
  Bridge "br0"
    Port "eth3"
      Interface "eth3"
    Port "eth1"
      Interface "eth1"
    Port "br0"
      Interface "br0"
        type: internal
    Port "eth2"
      Interface "eth2"
/ #
```

A red box highlights the command `ovs-vsctl add-port br0 eth3` and its resulting output, which shows the bridge now contains port "eth3".

**Figure 12 Section 3, Part 2:** The rearranged ports on the bridge are shown in the highlighted box. Before the results, the user had to delete the interface `eth0` from the bridge. Using the command `ovs-vsctl del-port br0 eth0` made all of that possible in the first place. After that command was executed then you would do the reverse to add the `eth3` port into the bridge.

## SECTION 3, PART 3: Assign an IP address to a Managed Switch

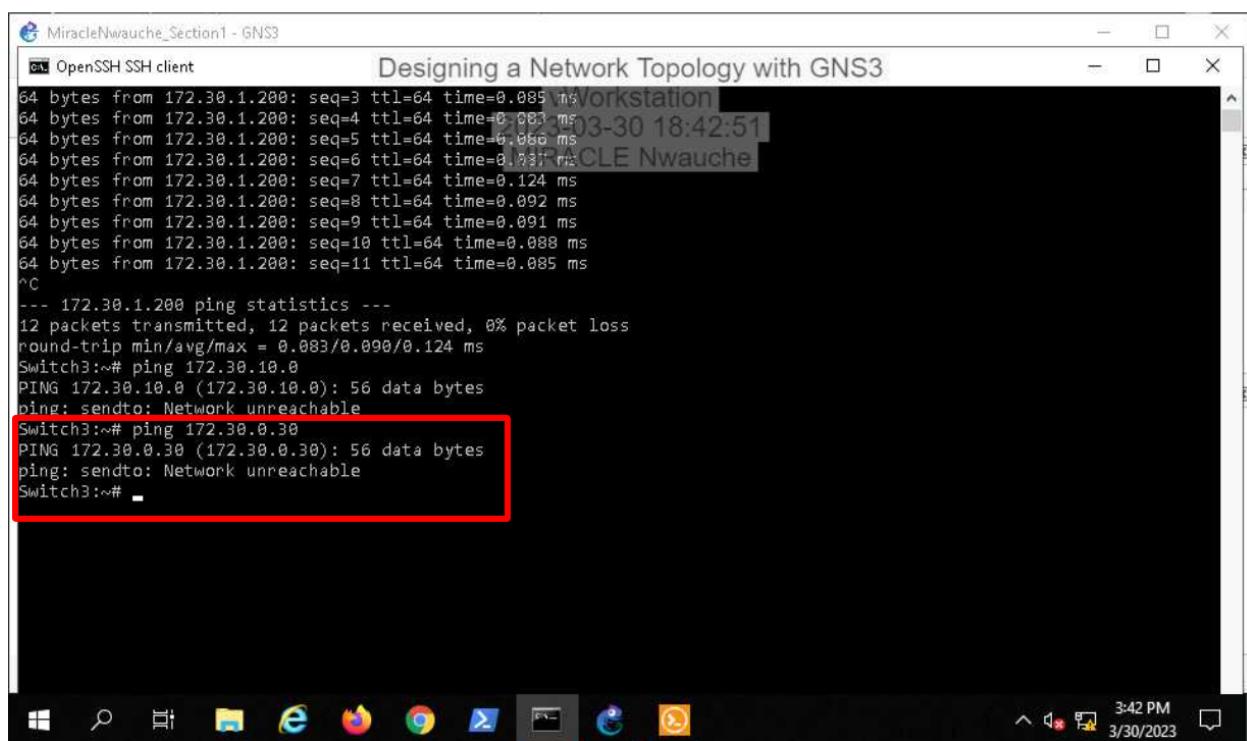
Make a screen capture showing the successful SSH login.



The screenshot shows a Windows terminal window titled "Designing a Network Topology with GNS3". The terminal is running an OpenSSH SSH client. The user has entered the command "ssh root@172.30.1.200". The terminal displays several messages related to host key verification and fingerprint matching. A red rectangular box highlights the portion of the terminal where the user is prompted to "Are you sure you want to continue connecting (yes/no)?". The user has responded with "yes", and the terminal shows the message "Warning: Permanently added '172.30.1.200' (ECDSA) to the list of known hosts." and "Welcome to your management console!". At the bottom of the terminal, it says "Happy configuring!" followed by a prompt "Switch3:~#". The taskbar at the bottom of the screen shows various icons for other applications like File Explorer, Edge, and Google Chrome.

**Figure 13 Section 3, Part 3:** After a couple tries, I finally got into the root account via SSH using the command in the above highlighted box. The thing I forgot to do was type in the “yes” or “no,” once I did that the configuration worked! It even left a nice message saying “Happy configuring!”

**Make a screen capture showing the results of your ping.**



```
MiracleNwauche_Section1 - GNS3
OpenSSH SSH client      Designing a Network Topology with GNS3
64 bytes from 172.30.1.200: seq=3 ttl=64 time=0.085 ms
64 bytes from 172.30.1.200: seq=4 ttl=64 time=0.083 ms
64 bytes from 172.30.1.200: seq=5 ttl=64 time=0.060 ms
64 bytes from 172.30.1.200: seq=6 ttl=64 time=0.085 ms
64 bytes from 172.30.1.200: seq=7 ttl=64 time=0.124 ms
64 bytes from 172.30.1.200: seq=8 ttl=64 time=0.092 ms
64 bytes from 172.30.1.200: seq=9 ttl=64 time=0.091 ms
64 bytes from 172.30.1.200: seq=10 ttl=64 time=0.088 ms
64 bytes from 172.30.1.200: seq=11 ttl=64 time=0.085 ms
^C
--- 172.30.1.200 ping statistics ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 0.083/0.090/0.124 ms
Switch3:~# ping 172.30.10.0
PING 172.30.10.0 (172.30.10.0): 56 data bytes
ping: sendto: Network unreachable
Switch3:~# ping 172.30.0.30
PING 172.30.0.30 (172.30.0.30): 56 data bytes
ping: sendto: Network unreachable
Switch3:~#
```

**Figure 15, Section 3, Part 3:** In the command prompt, the last command the user needs to execute is to confirm that the destination is unreachable because of the connection that was terminated (eth0). Once this is confirmed, we now know that the commands in the PuTTY application worked correctly and the network topology was managed correctly.