LAB REPORT

LAB TITLE: Designing Network Topology with GNS3

Name: Gabriel Miracle Nwauche

PSU ID: 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

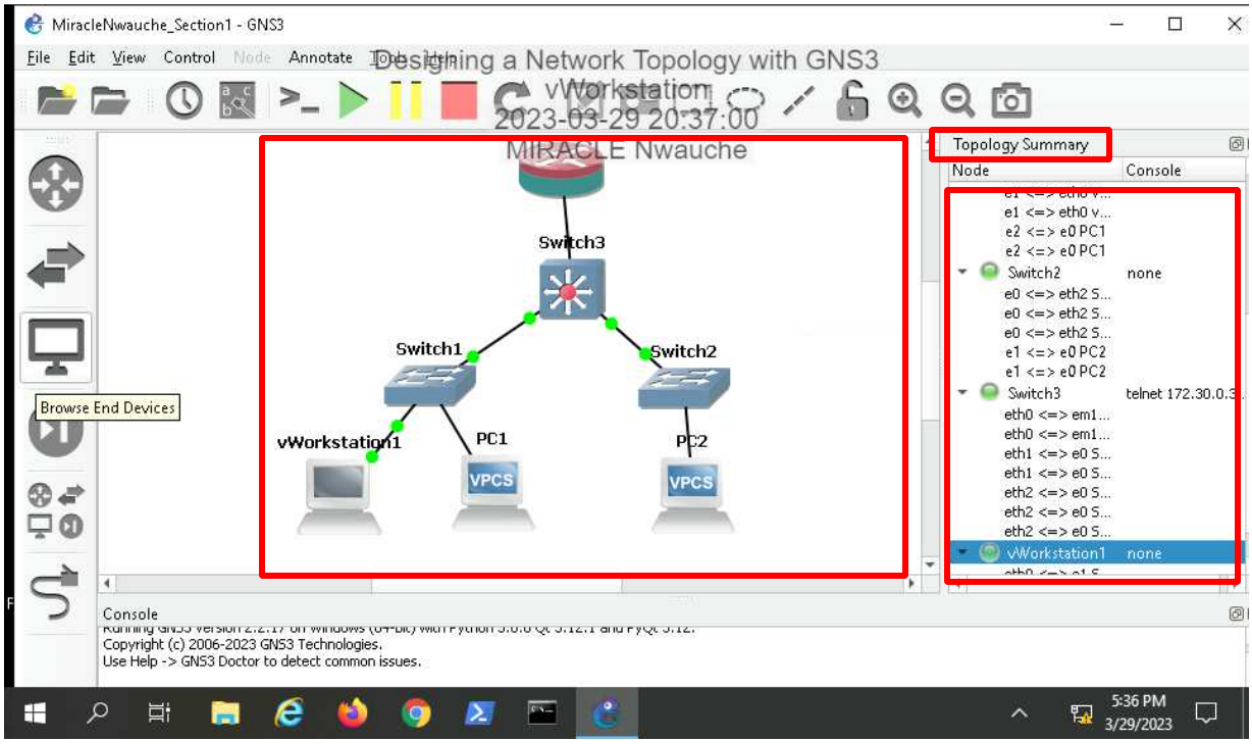

Figure 1 Section 1, Part 1: Captured here are the nodes with all of their connections and topologies relating to each other using the ethernet. On the right is the Topology summary showing the active connections in the topology tree.

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

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

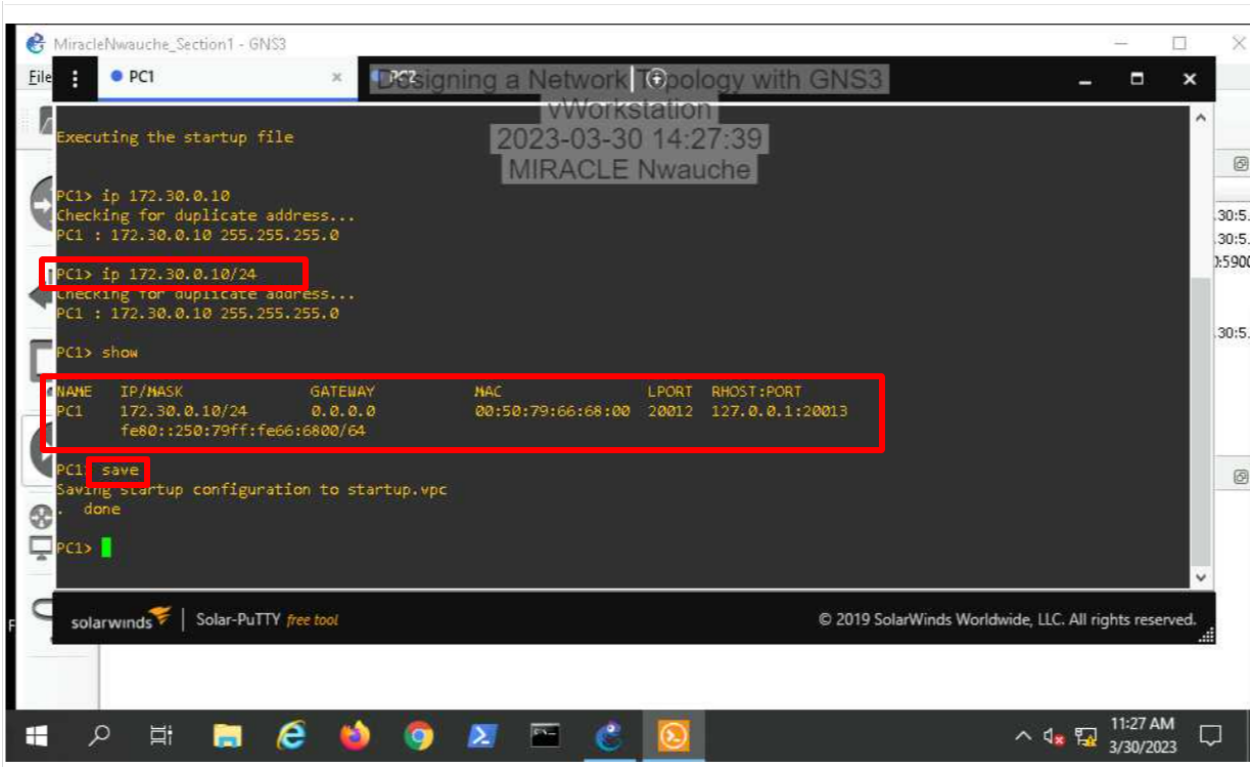

Figure 2 Section 1, Part 2: The configured commands are as shown in the highlighted boxes. Once the PCs are consoled using the GNS3 device, a PuTTy application opens where command lines can be transmuted. We type in the IP address to make sure there are no duplicated addresses, show the configuration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

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

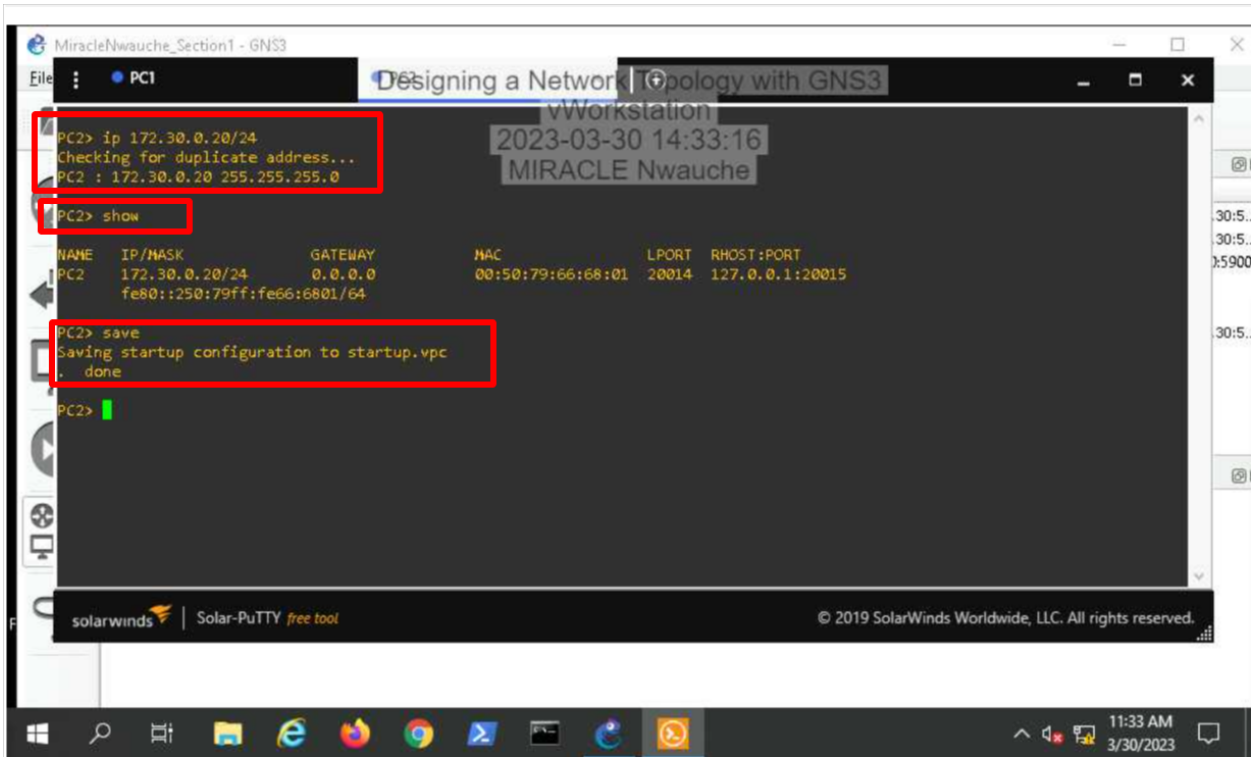

Figure 3 Section 1, Part 2: Much like the information in the previous capture, we do the same functions in the command line to get the same results. This time the IP address is not the same, so it will *show* a different configuration than the one in the prior PC device.

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

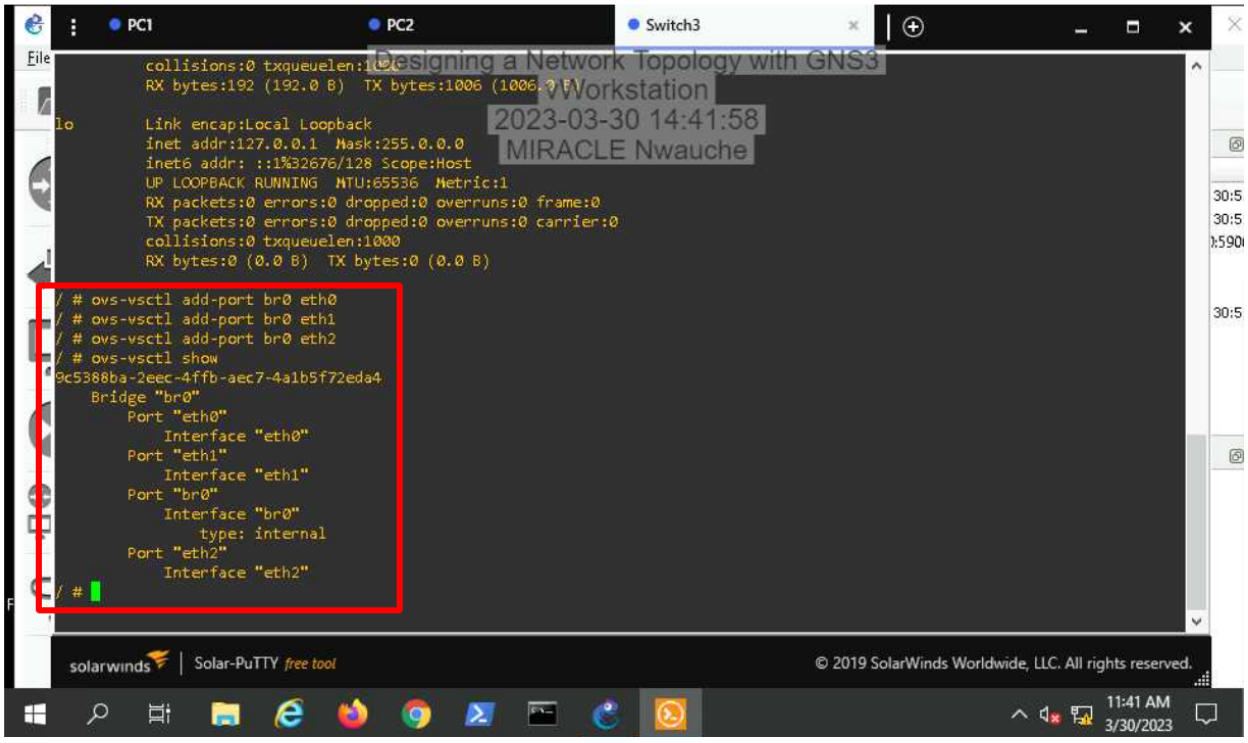

Figure 4 Section 1, Part 2: The highlighted frame shows the *ovs-vsctl* configuration of the bridge that connects all of the *eth* ports together. Once the command is executed, using the *show* command the user will be able to see the Bridge followed by the Ports that connect to that bridge. The interface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

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

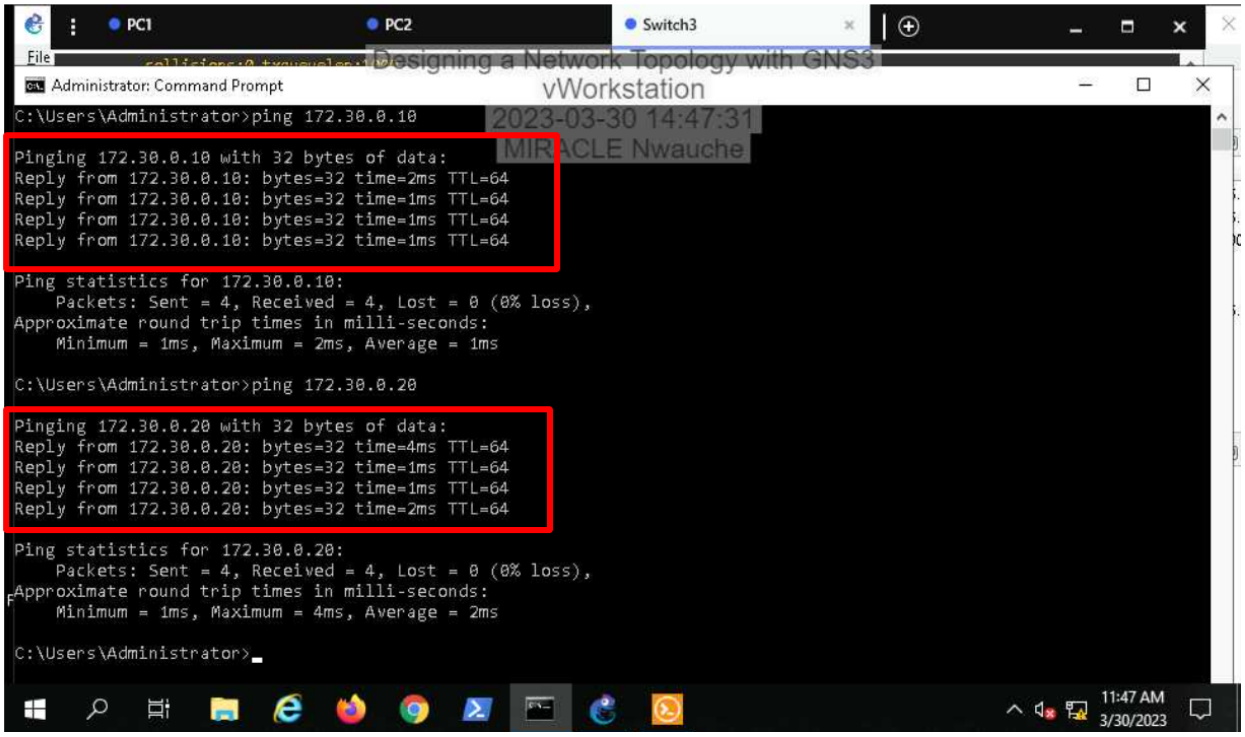

Figure 5 Section 1, Part 2: The next step was to go into the command prompt to see if the PCs were properly connected. To do this, the user had to ping the IP address to see if the replies were successful. From the highlighted spaces in the picture above, we can see that the ping was indeed successful, 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**.

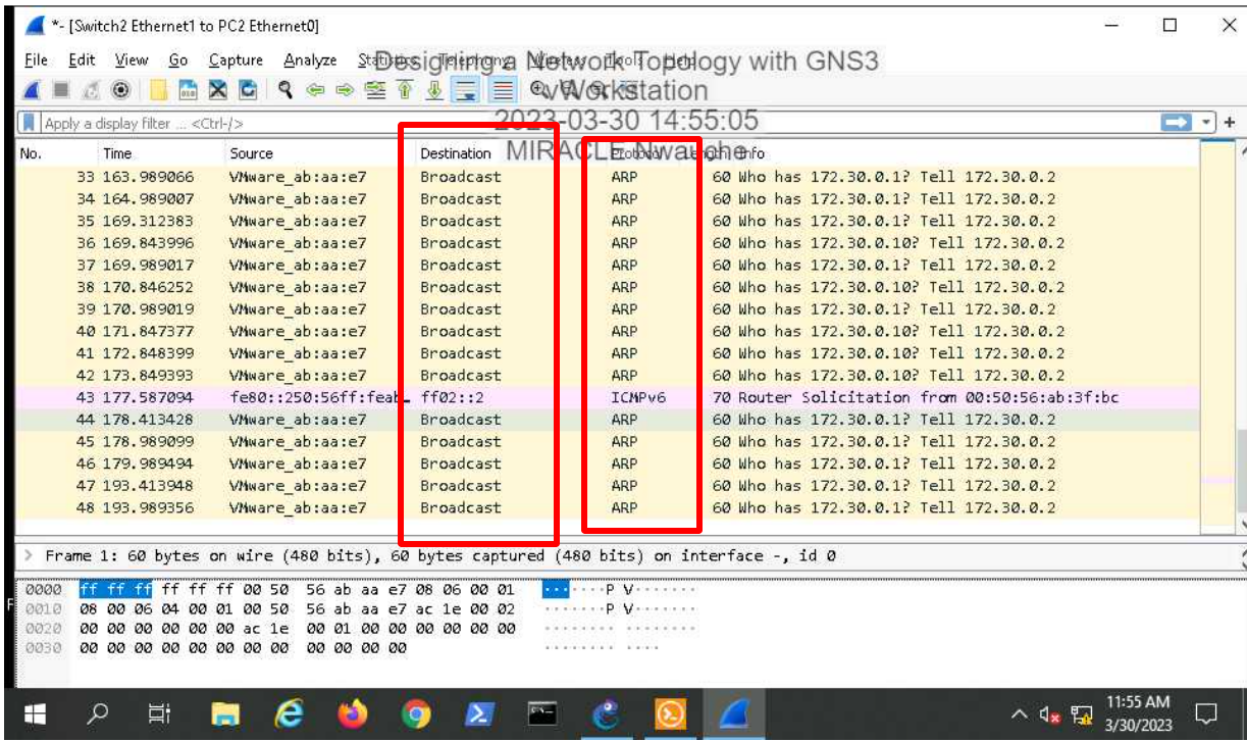

Figure 6 Section 1, Part 2: As prompted in the lab instructions, the user had to go through the command prompt to see the arp configuration between the Switch2 device and the PC2 device. Since this connection was the farthest away from the two devices connected to Switch1, it is good to see that the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

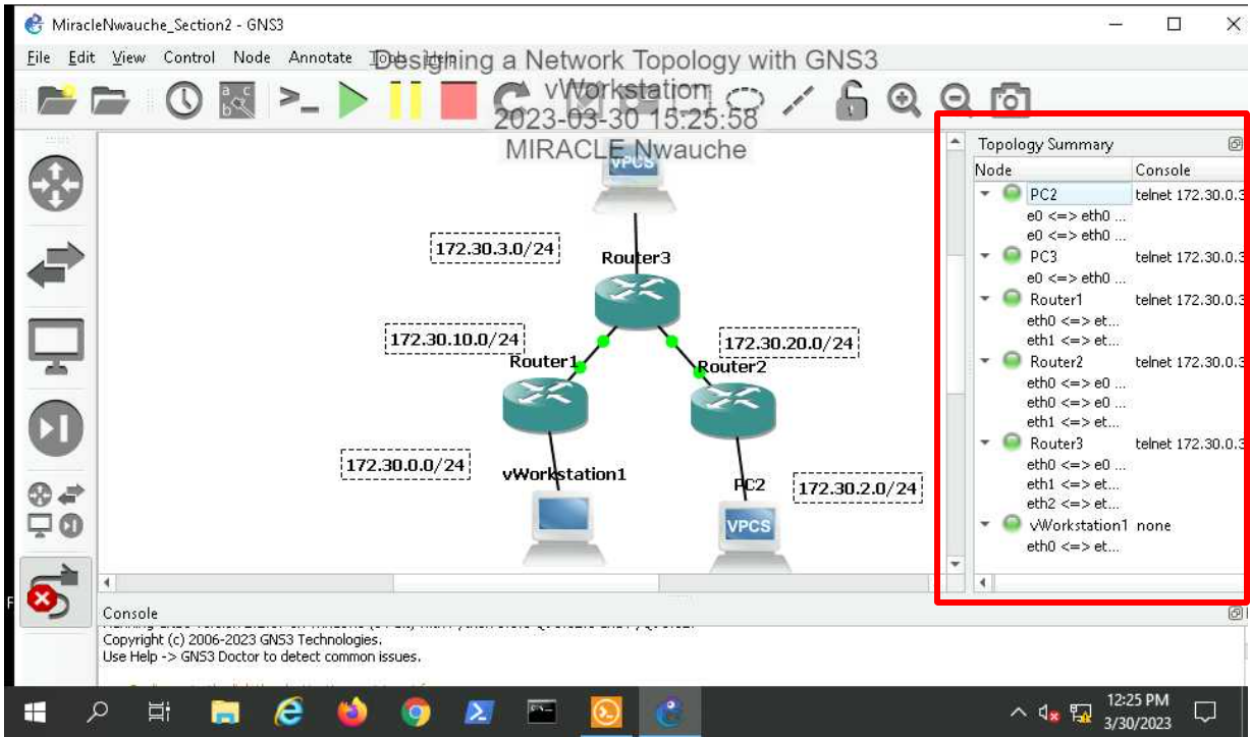

Figure 7 Section 2, Part 1: The network topology of the second network is shown above, this time with a little more information than the previous example. Instead of using the switches, the routers are substituted to show the connection of IP addresses between the devices shown. Once all of the system 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.**

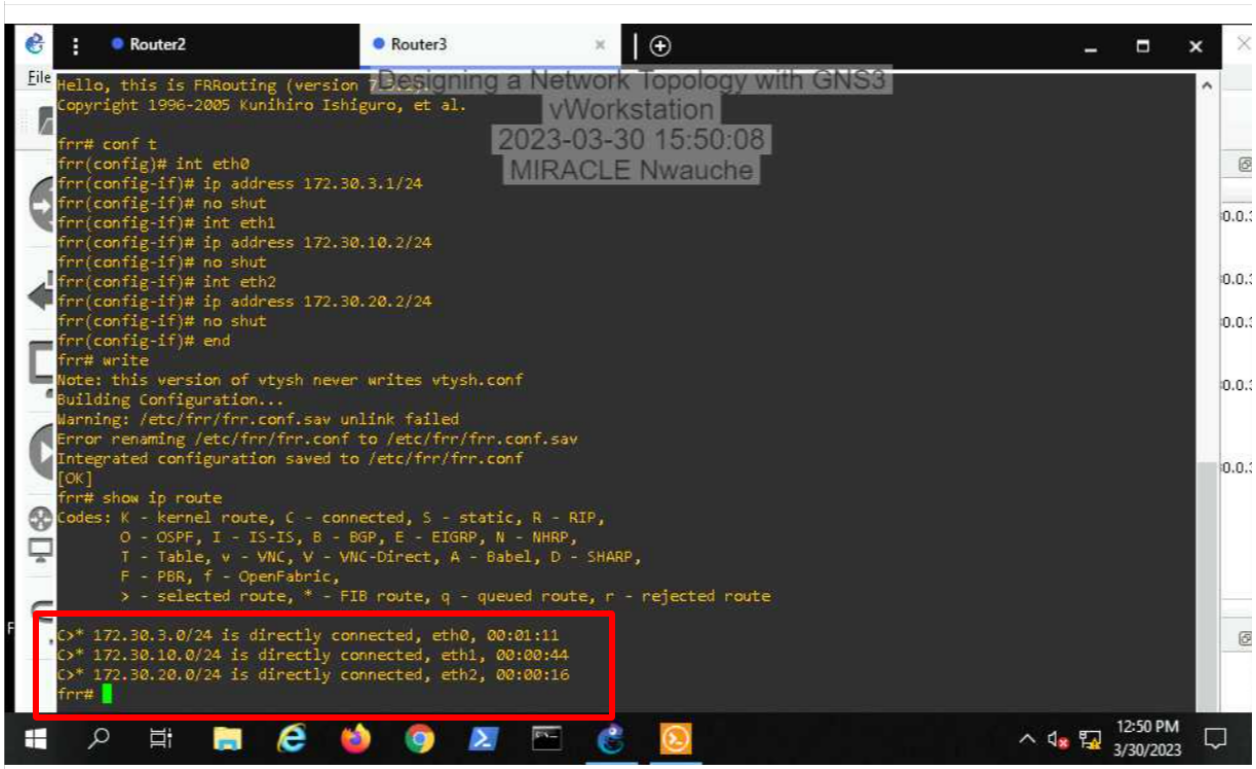

Figure 8 Section 2, Part 2: The IP route for the routers connected in the logically connected Layer 3 network are shown above. After configuring the IP addresses for each of the interfaces connected to all the routers, making sure they won't shut down (using the *no shut* command for that) as well. The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

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

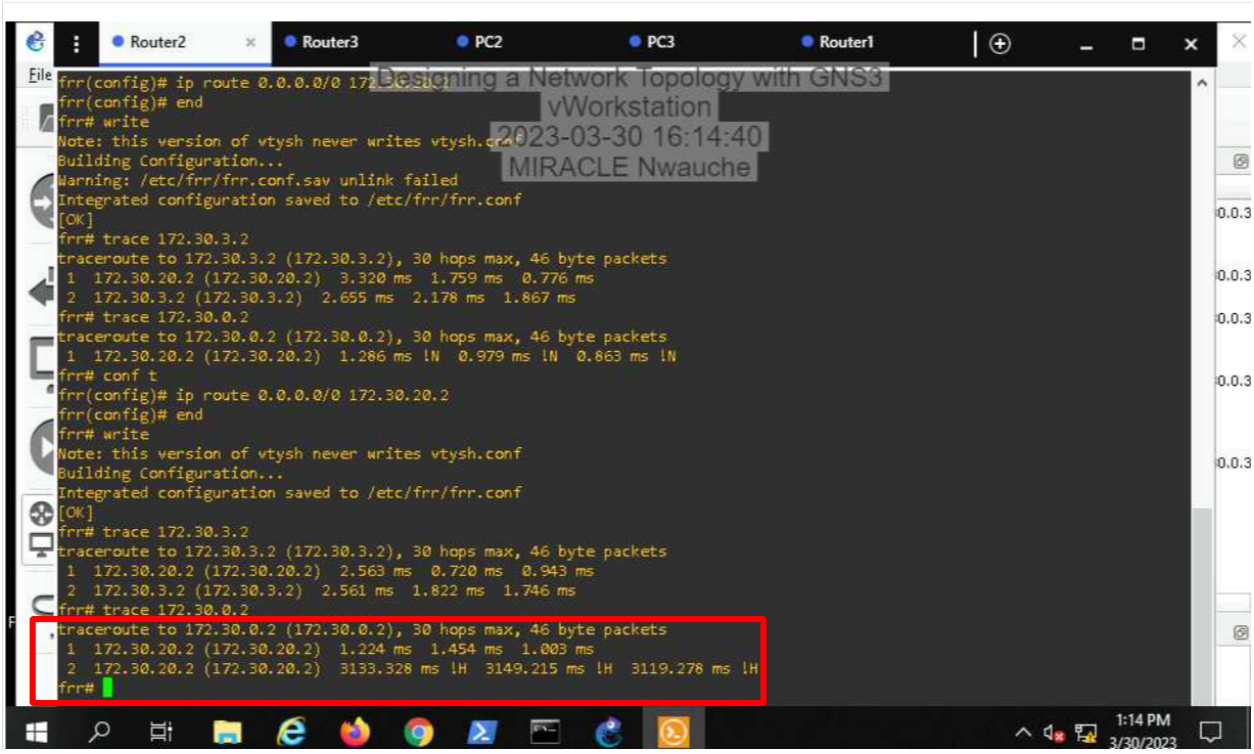

Figure 9 Section 2, Part 2: This capture is a little complex because of the way the traceroute worked with this specific command. Since the trace route for the second IP is only supposed to reach Router3 and then fail, there are exclamation points next to the time sensitive information after the command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

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

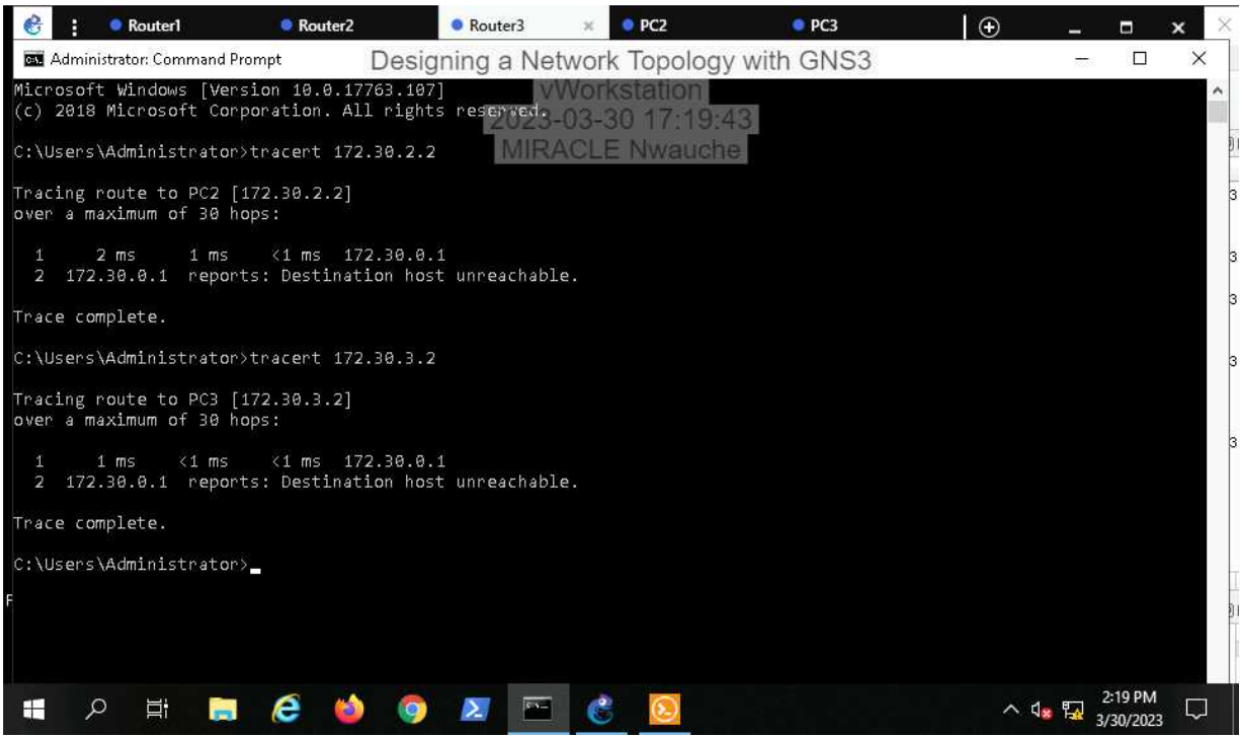

Figure 10 Section 2, Part 2: These were my results from the traceroute. I'm not sure why but whenever I tried to configure the IP address with the Gateway *172.30.10.2/24* it would give me a warning. I'm guessing that is why the second destination is unreachable for both of the routers.

SECTION 3: PART 3: Manage Switch Ports

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

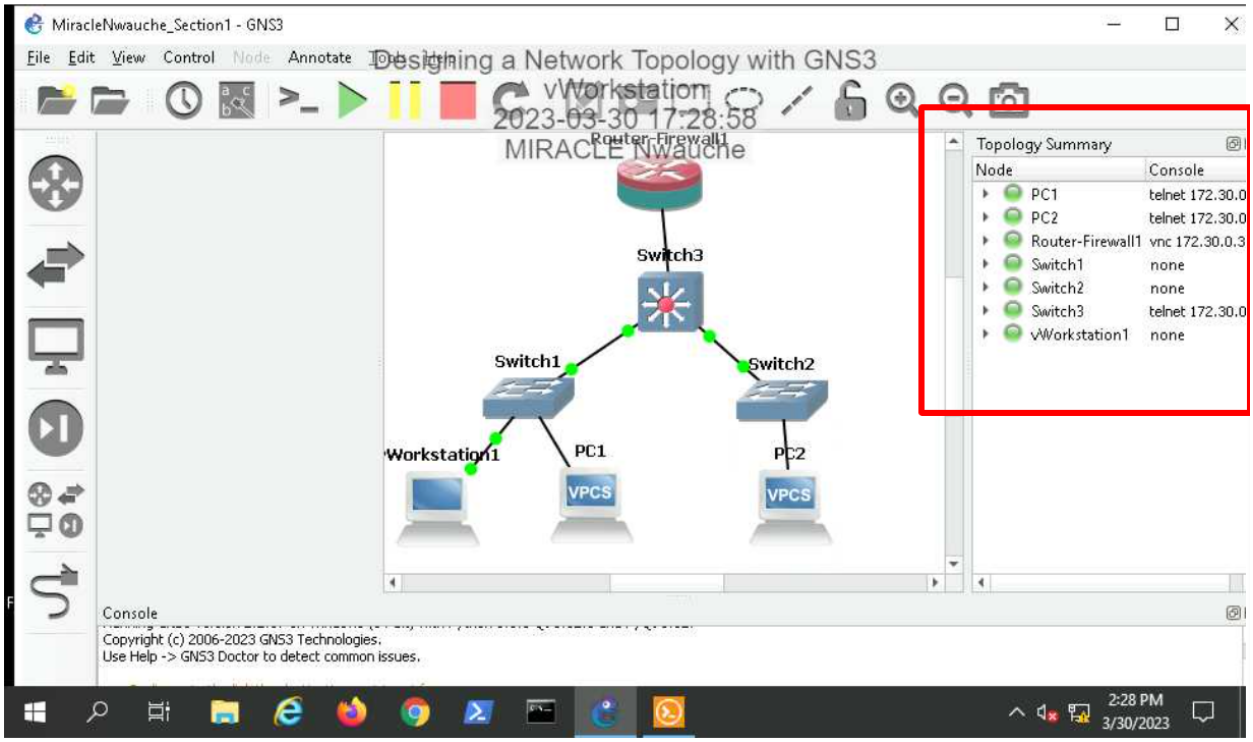

Figure 11 Section 3, Part 3: Here is the network topology for the connection in Section 1. Switch3 with an extra adapter. After everything is powered on the right side of the screen shows that all of the connections are active. This topology shows that switch 3 is a four way connection for switches 1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

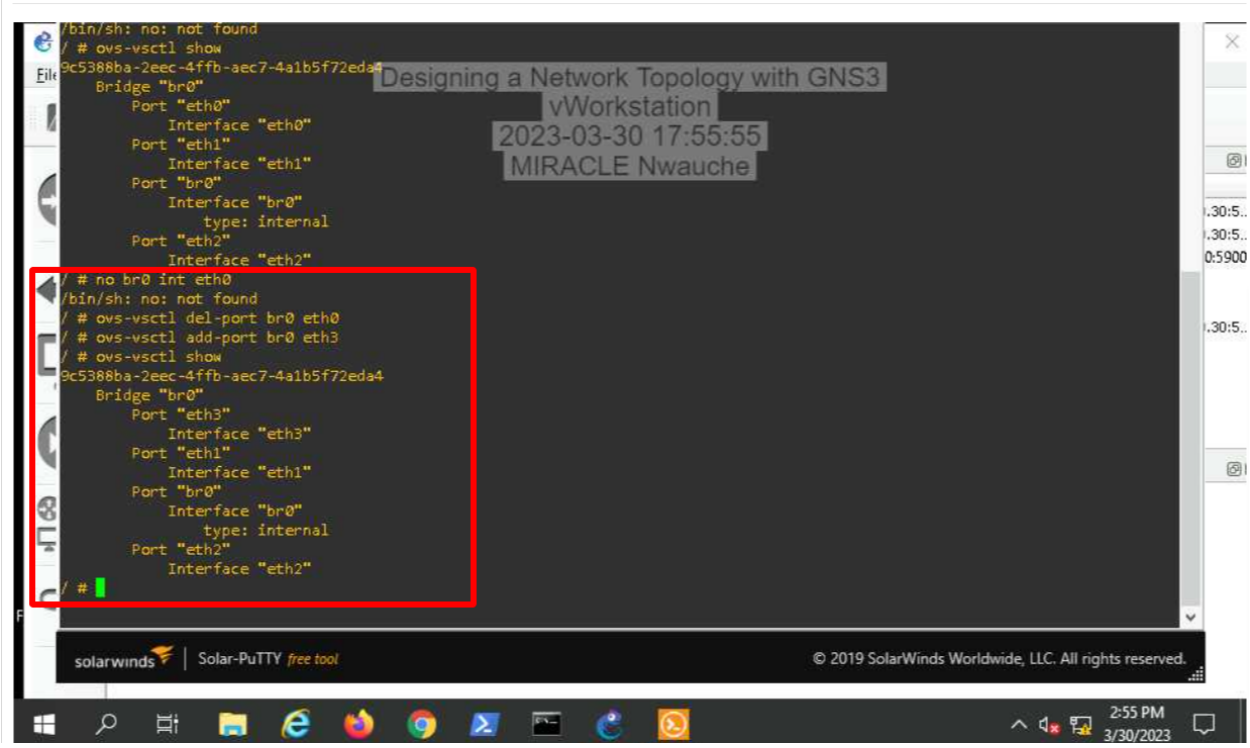

Figure 12 Section 3, Part 2: The rearranged ports on the bridge are shown in the highlighted box. Before the results, the user had to delete the interface *eth0* from the bridge. Using the command *ovs-vsctl del-port br0 eth0* made all of that possible in the first place. After that command was executed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

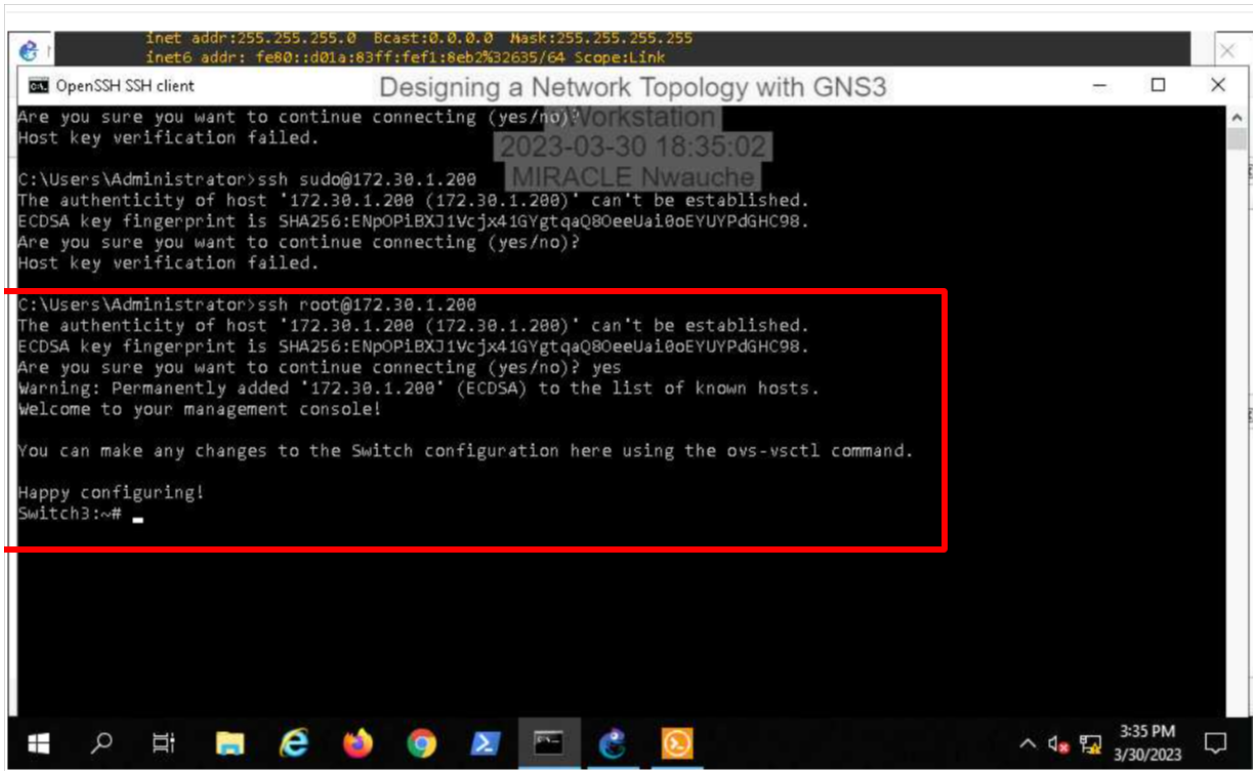

Figure 13 Section 3, Part 3: After a couple tries, I finally got into the root account via SSH using the command in the above highlighted box. The thing I forgot to do was type in the "yes" or "no," once I did that the configuration worked! It even left a nice message saying "Happy configuring!"

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

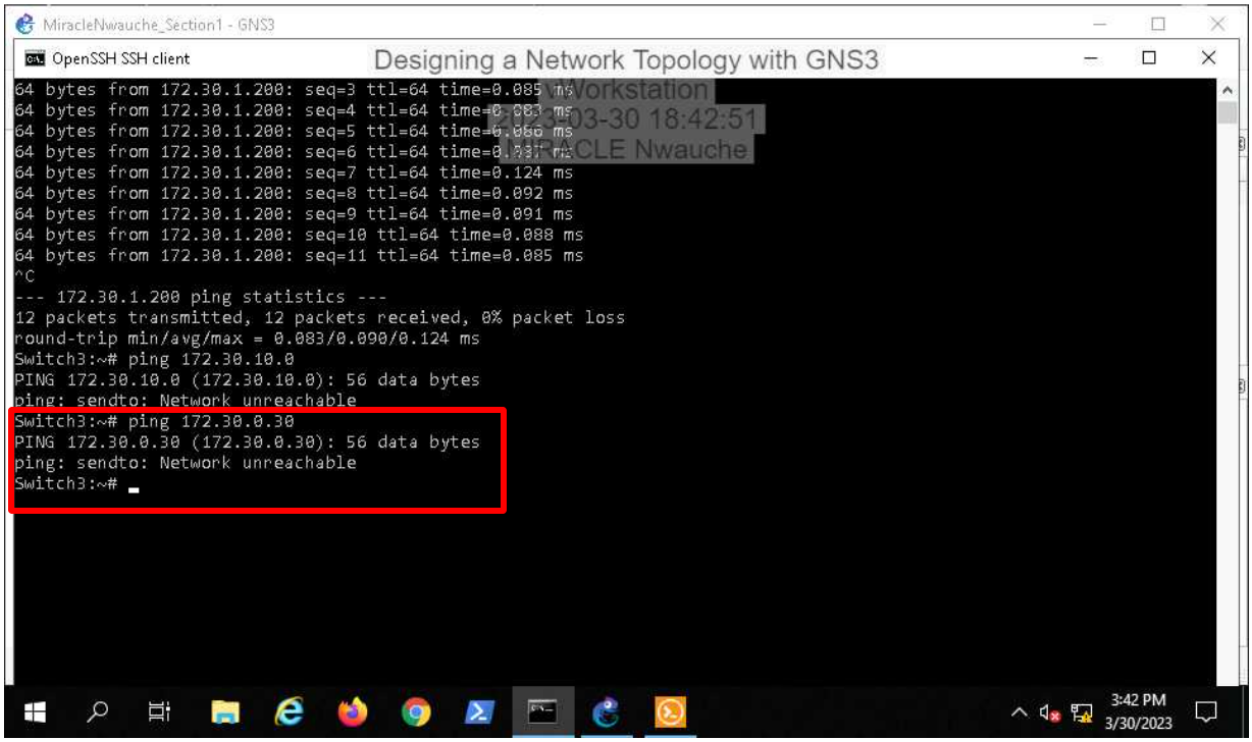

Figure 15, Section 3, Part 3: In the command prompt, the last command the user needs to execute is to confirm that the destination is unreachable because of the connection that was terminated (eth0). Once this is confirmed, we now know that the commands in the PuTTy application worked correctly and the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