PROJECT REPORT

Data Communication & Networking

Muhammad Ammar Hashmi
01-134201-048

Bahria University, Islamabad Campus

Department of Computer Science

Assignment 1

Class/Section: BS-CS (4 C & D)

(Spring 2024 Semester)

Course: Data Communication and Networking	Date: 3/6/2024
Course Code: CEN-222	
Faculty's Name: Dr. Moazam Ali	Max Marks: 5
Time Allowed:	Total Pages: 1 (including this)

Student Name: Muhammad Ammar Hashmi **Enrollment Number:** 01-134201-048

Table of Contents

1. Abstract:

This report presents a comprehensive study and implementation of various network topologies using GNS3, a network simulation tool. It explores the functionalities, advantages, and limitations of GNS3 and compares it with other network simulators. The report details the setup and configuration of star, mesh, bus, and ring topologies, providing step-by-step instructions and screenshots to illustrate the process. Additionally, the report includes an analysis of network protocols and the use of packet sniffing tools like Wireshark to monitor network traffic and diagnose connectivity issues. This study aims to provide a practical understanding of network design and simulation for educational and professional purposes.

2. Introduction

The study of network topologies and the implementation of various network configurations are essential for understanding data communication and networking. This report focuses on using GNS3 (Graphical Network Simulator-3), a powerful network simulation tool, to create and analyze different network topologies. GNS3 allows users to emulate real network devices and configurations, providing a practical learning environment for network engineering students and professionals. This report aims to explore the capabilities of GNS3, demonstrate the setup of different topologies, and evaluate the tool's performance in simulating realistic network scenarios.

3. Question No. 1:

Define what is GNS3 (introduction section)?

Answer:

3.1. Introduction to GNS3:

GNS3, or Graphical Network Simulator version 3, is a powerful tool used by networking professionals and students alike to design, configure, and test virtual networks. It's like having a virtual lab where we can experiment with different network setups without needing physical equipment.

3.2. The GNS Team:

The team behind GNS3 includes some highly skilled individuals, such as Christophe Fillot, Jeremy Grossmann, and Julien Duponchelle.

- **Christophe Fillot** is known for creating Dynamips, a program that emulates MIPS processors, which are used in Cisco routers. This emulation allows GNS3 users to run Cisco's router operating system (IOS) on their computers, simulating a real networking environment.
- **Jeremy Grossmann** is the brains behind GNS3 itself, integrating Dynamips and other open-source tools into an easy-to-use graphical interface.
- **Julien Duponchelle** contributes to the development of GNS3 by writing code, helping to improve its functionality and performance over time.

3.3. Advantages:

One of the main advantages of GNS3 is its versatility. we can create virtual networks using various devices and operating systems, including Cisco IOS, Juniper, MikroTik, Arista, and Vyatta. This means we can practice configuring and troubleshooting different types of networks, which is incredibly valuable for students preparing for networking certifications like the CCNA and CCNP.

3.4. Limitations:

However, it's important to understand that GNS3 has its limitations. While it's a fantastic tool for learning and testing, it's not a perfect substitute for real-world experience. Some features of physical networks, such as hardware limitations and environmental factors, cannot be accurately replicated in a virtual environment. Additionally, GNS3 may not support every feature or device that we would encounter in a production network.

GNS3 is a valuable tool for networking enthusiasts and professionals to gain hands-on experience with network design and configuration. It provides a safe and cost-effective way to experiment with different network setups and prepare for certification exams. However, it's essential to recognize its limitations and supplement virtual lab work with real-world experience where possible.

4. Question No. 2:

List reasons, why we have to use GNS3 (introduction section)?

Answer:

4.1. Why use GNS3:

4.1.1. Traditional Lab Setup Challenges:

Before virtualization became widely available, setting up network labs required physical hardware or renting time on specialized racks, both of which were costly and inconvenient. Additionally, these options limited the variety of network designs we could work with.

4.1.2. Limitations of Other Simulation Programs:

While there were software simulation programs like RouterSim and Boson NetSim, they were restricted to simulating only the commands of Cisco IOS, the operating system used in Cisco devices. Cisco Education did offer a more affordable option with virtualized rack rental based on Cisco IOS on Unix (IOU). However, this method had its drawbacks.

4.1.3. Advantages of GNS3:

GNS3, on the other hand, stands out for its flexibility and freedom. It allows us to customize our network labs precisely according to our needs. We can create unlimited projects using both Cisco and non-Cisco technologies and add as many devices as we want to our projects. Moreover, GNS3 doesn't require constant internet access; we can access our projects anytime, even offline.

4.1.4. Flexibility and Freedom:

The key to GNS3's flexibility lies in its ability to emulate real hardware devices running actual network operating systems like Cisco IOS and simulate operating systems like NX-OSv. Additionally, it enables you to distribute resources across multiple computers, further enhancing its capabilities.

5. Question No. 3:

What are the advantages and disadvantages of GNS3?

Answer:

5.1. Advantages of GNS3:

5.1.1. Emulated Hardware:

GNS3 has some cool advantages, and one big one is how it handles emulated hardware. Basically, it lets us set up virtual networks with different kinds of devices like routers, switches, and computers. But what makes GNS3 stand out is its ability to work with Cisco IOS, the operating system used in Cisco devices. Unlike other similar programs that just pretend to be using Cisco IOS, GNS3 actually runs the real IOS software on our computer. This means we get all the real commands and features of IOS, just like we would on a physical Cisco device. So, we can use any protocol or feature that the IOS version supports in our network designs. This sets GNS3 apart from programs like RouterSim, Boson NetSim, or VIRL, which only provide limited environments and features to work with.

5.1.2. Simulated Operating Systems:

In addition to emulating hardware, GNS3 also supports simulated operating systems. This means we can incorporate various operating systems into our virtual network setups. For example, we can use Linux or Windows virtual machines alongside our Cisco devices to simulate real-world network environments. This versatility allows us to test a wide range of configurations and scenarios within our virtual network.

5.1.3. Scalability with the GNS3 Server:

GNS3 leverages client-server technology; much like a web browser connects to a web server to access and display web pages, the GNS3 graphical user interface (GUI) program accesses a GNSS server, allowing it to start, stop, and otherwise control GNS3 devices. This allows our projects to scale because they're not restricted to running on a single computer. If we work with large or complex topologies, we can also run the GNS3 server program on a different PC than the GNSS GUI program. If we have access to a high-end server with a lot of memory and processing power, we can install the GNS3 server program on the server hardware but control all the devices from the GNS3 GUI program running on a more modest PC.

5.1.4. Virtual Connectivity:

One of the key features of GNS3 is its ability to simulate network connectivity between virtual devices. Using protocols like IPv4 and IPv6, we can create interconnected virtual networks that behave like real-world networks. This virtual connectivity allows us to test communication between devices, configure routing protocols, and troubleshoot network issues in a safe and controlled environment. Whether we're setting up a simple lab with a few components or designing a complex network topology, GNS3 provides the flexibility to simulate realistic network scenarios and validate our configurations.

5.2. Disadvantages of GNS3:

GNS3 does have its drawbacks. Dynamips, for example, is restricted in a way that makes it unsuitable for use in actual production environments, meaning it's primarily intended for educational purposes. Some limitations of GNS3 are:

5.2.1. Some Assembly Required:

GNS3 has a bit of a learning curve and requires some setup. One major requirement is obtaining Cisco IOS images to run on virtual Dynamips routers. GNS3 doesn't provide these images, so we need to source them from a router we own or through a Cisco

Connection Online (CCO) account if we have a contract with Cisco. This process can be a hassle for beginners and adds complexity to getting started with GNS3.

5.2.2. Limited Emulation:

GNS3, particularly through its Dynamips component, struggles to emulate the hardware of advanced Catalyst switches, especially their application-specific integrated circuits (ASICs). This limitation affects those aiming for advanced Cisco certifications like the Cisco Certified Internetwork Expert (CCIE), as they require access to these advanced switch features. While GNS3 can integrate with physical Catalyst switches to address this limitation, it adds complexity and cost to setting up labs. Additionally, virtual switch modules like the Cisco NM-16ESW can be used for basic layer 3 switching needs, but they lack the advanced features of physical switches. Although Cisco IOU images can emulate Cisco switches with more commands than Dynamips, they still have limitations compared to physical hardware.

5.2.3. Hamstrung Network Performance:

Another downside of GNS3 is its limited network performance. Since Dynamips is an emulator without hardware acceleration, network throughput is restricted, typically ranging from 1.5Mb to 800Mb per second, depending on the IOS version and configuration. While this limitation prevents users from virtualizing Cisco hardware for production environments, it also affects network testing and simulations. Users may experience slower performance than expected, which can impact the accuracy of test results and simulations. However, this limitation is necessary to prevent misuse of virtualized Cisco hardware in production environments and to avoid potential legal issues with Cisco.

6. Question No. 4:

Comparison of GNS3 with other networking simulation tools (related work section).

Answer:

6.1. GNS3 vs Packet Tracer

GNS3	Packet Tracer
Offers more advanced features and flexibility compared to Packet Tracer. It supports the emulation of real Cisco IOS software and integrates with various virtualization technologies.	Primarily designed for beginners and students, Packet Tracer provides a user-friendly interface for learning networking concepts. It's limited in terms of supported devices and features but is sufficient for basic networking exercises and scenarios.

6.2. GNS3 vs Cisco VIRL (Virtual Internet Routing Lab)

GNS3	VIRL
Provides greater flexibility and customization options compared to VIRL. It supports a wider range of devices and operating systems, including non-Cisco devices, and allows for scalability through its client-server architecture.	Offers a comprehensive set of Cisco networking devices and features, making it suitable for Cisco certification training and advanced network simulations. However, it requires a subscription fee and has limitations on the number of devices and topologies.

6.3. GNS3 vs Boson NetSim

GNS3	Boson NetSim
Offers more extensive device support and flexibility compared to Boson NetSim. GNS3 allows for the emulation of various networking devices and operating systems, including Cisco IOS, Juniper, and others.	Primarily focused on Cisco certification preparation, Boson NetSim provides a structured learning environment with pre-built labs and guided exercises. It's suitable for beginners and individuals seeking targeted practice for specific certifications.

6.4. GNS3 vs RouterSim

GNS3	RouterSim

Provides a more comprehensive and customizable network simulation environment compared to RouterSim. GNS3 supports a wider range of devices and allows for more extensive network configurations and scenarios.	Offers simplified network simulation tools aimed at beginners and individuals seeking basic networking practice. It's user-friendly but lacks the depth and flexibility of GNS3 for more advanced networking tasks.

7. Question No. 5:

a. Design Mesh, Ring, Star and Bus topology in GNS3?

b. Explain all these topologies that you have created in GNS3?

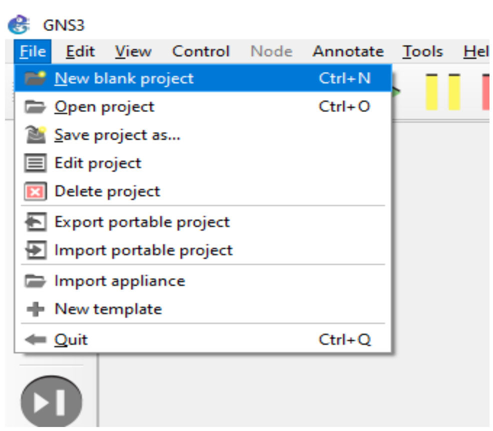
c. Take the screen shot of all the topologies.

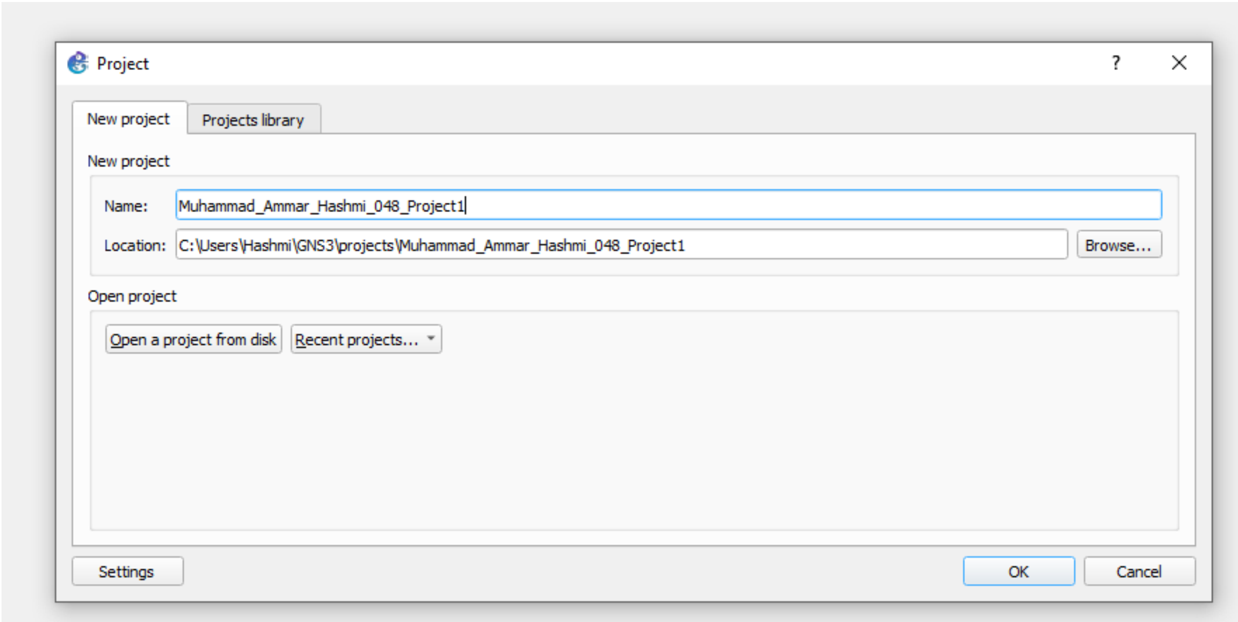
Answer:

Topologies design in GNS3

7.1. Setting up project:

This assignment begins by creating a new blank project in GNS3.Yhe project in GNS3 is created by clicking “file” button on top left corner of the screen and clicking ‘New blank project”. After that, the project is given its name.





7.2. Star Topology

7.2.1. What is Star Topology?

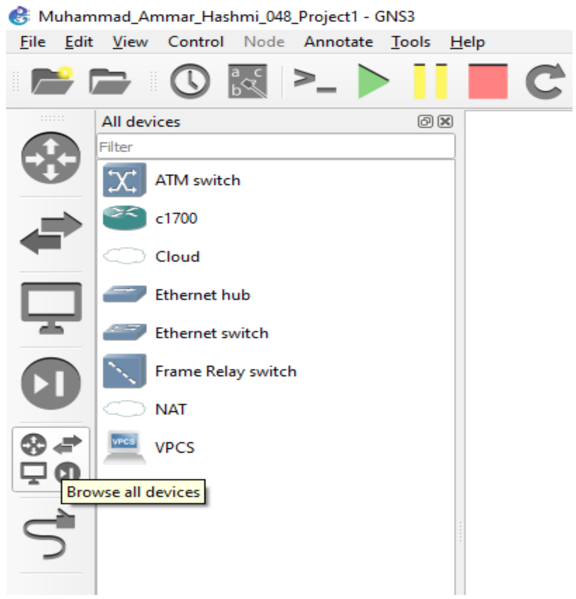
In a star topology, all devices are connected to a central device, such as a switch or a hub. Each device has a point-to-point connection only to the central hub.

7.2.2. Characteristics:

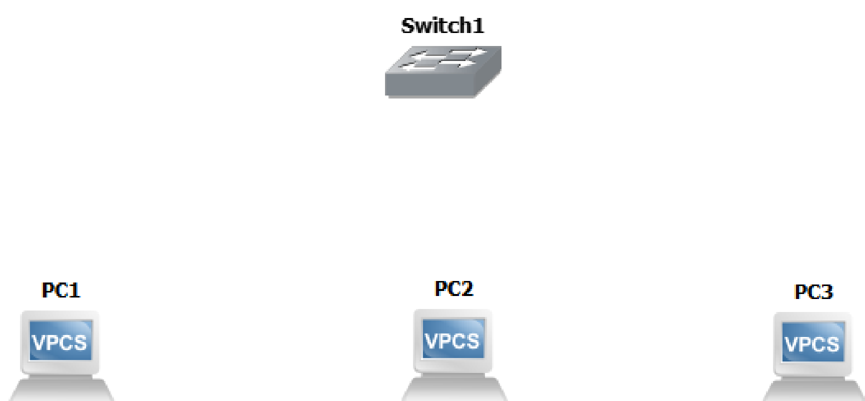
- It is easy to set up and manage.
- In stat topology, if one connection fails, it doesn't affect other devices.
- Central device can become a single point of failure.

7.2.3. How we implemented Star Topology in GNS3:

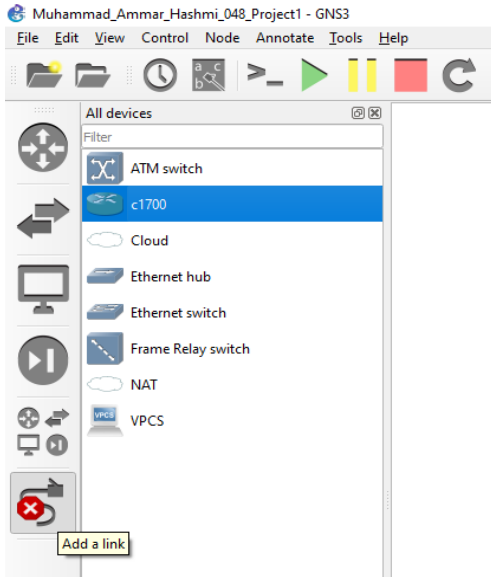
To implement start topology, we need some devices. We accessed these devices from the **“browse all devices”** section.



We then dragged and dropped 3 VPCs and 1 switch from **“Browse all devices”** section onto the project screen as shown below.



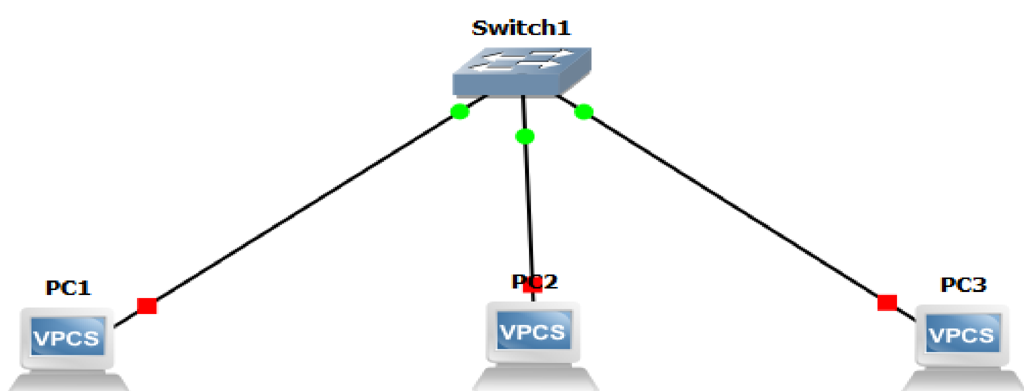
After that, to link these devices, we clicked on **“Add a link”** section and connected these devices through a link.



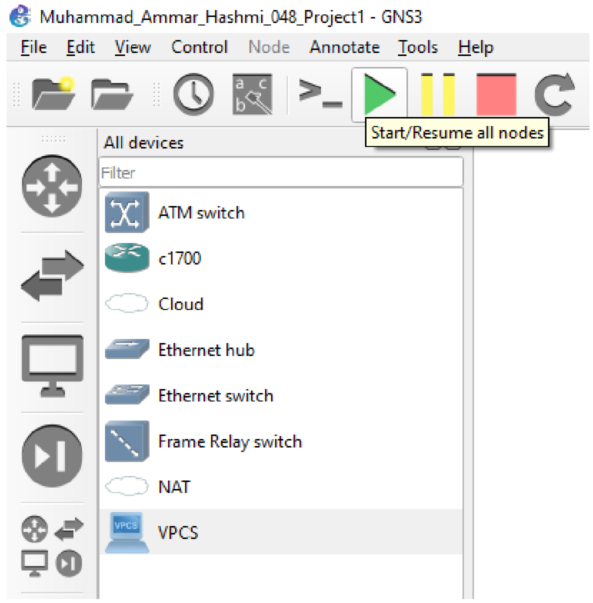
7.2.4. Connection information:

Connection between 3 VPCs and Switch 1:

- ethernet0 of pc1 is connected to ethernet0 of switch 1.
- ethernet0 of pc2 is connected to ethernet1 of switch 1.
- ethernet0 of pc3 is connected to ethernet2 of switch 1.

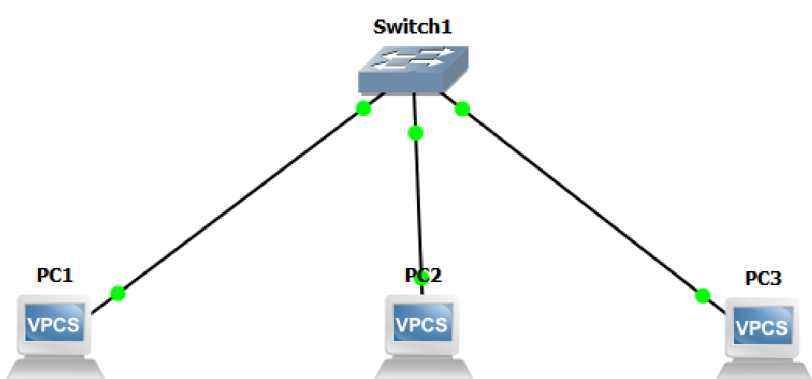


As we can see, the devices are not yet started. To start these devices, we clicked the start button on top of the screen to start all 3 devices.

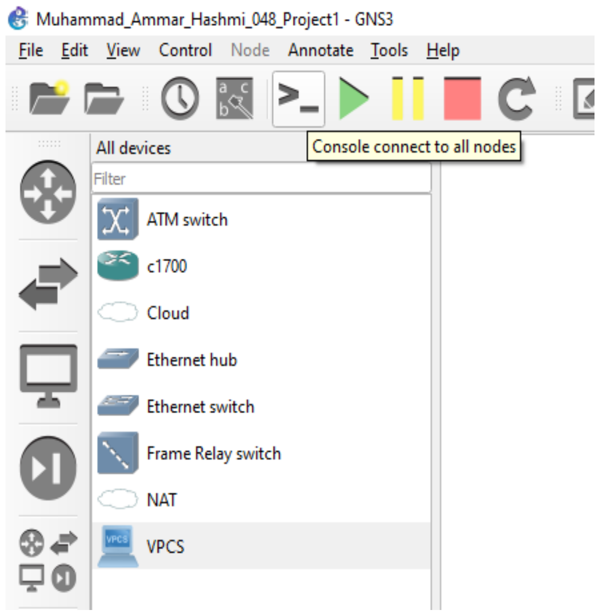


The devices were now started:

7.2.4.1. Screenshot of Star Topology



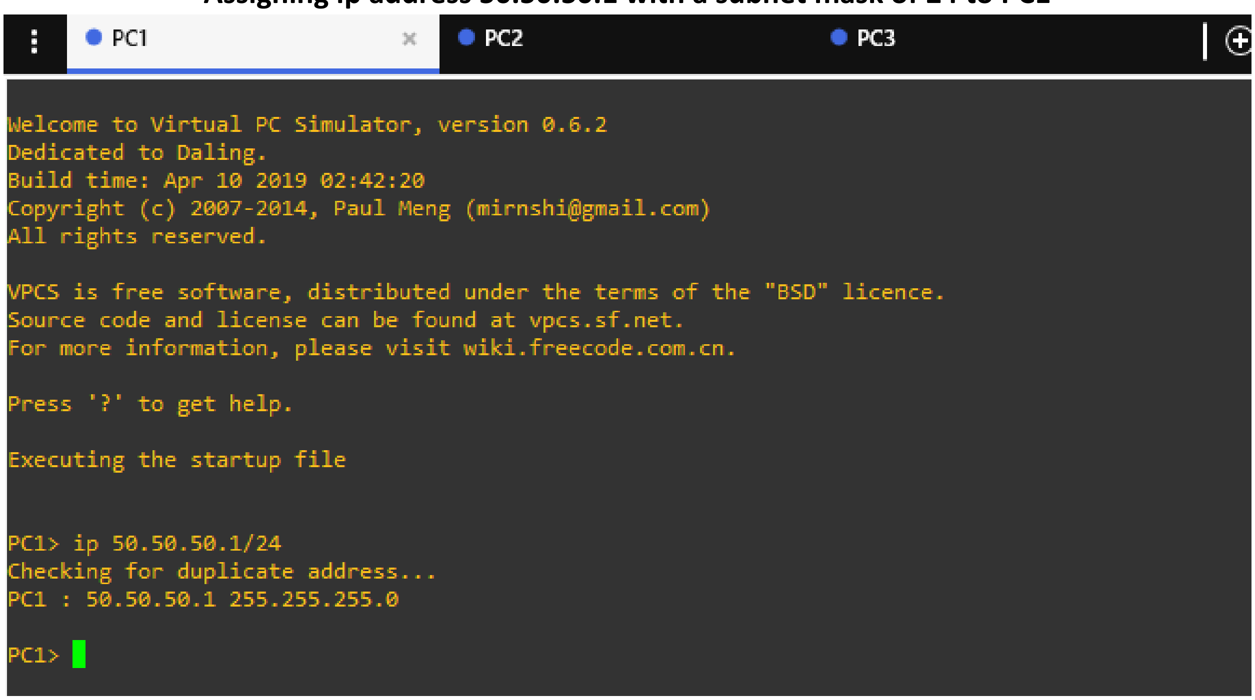
But we need to access consoles of these devices to find if they are connected. For this purpose, we clicked on the console button on top of the screen to start consoles for all 3 VPCs.



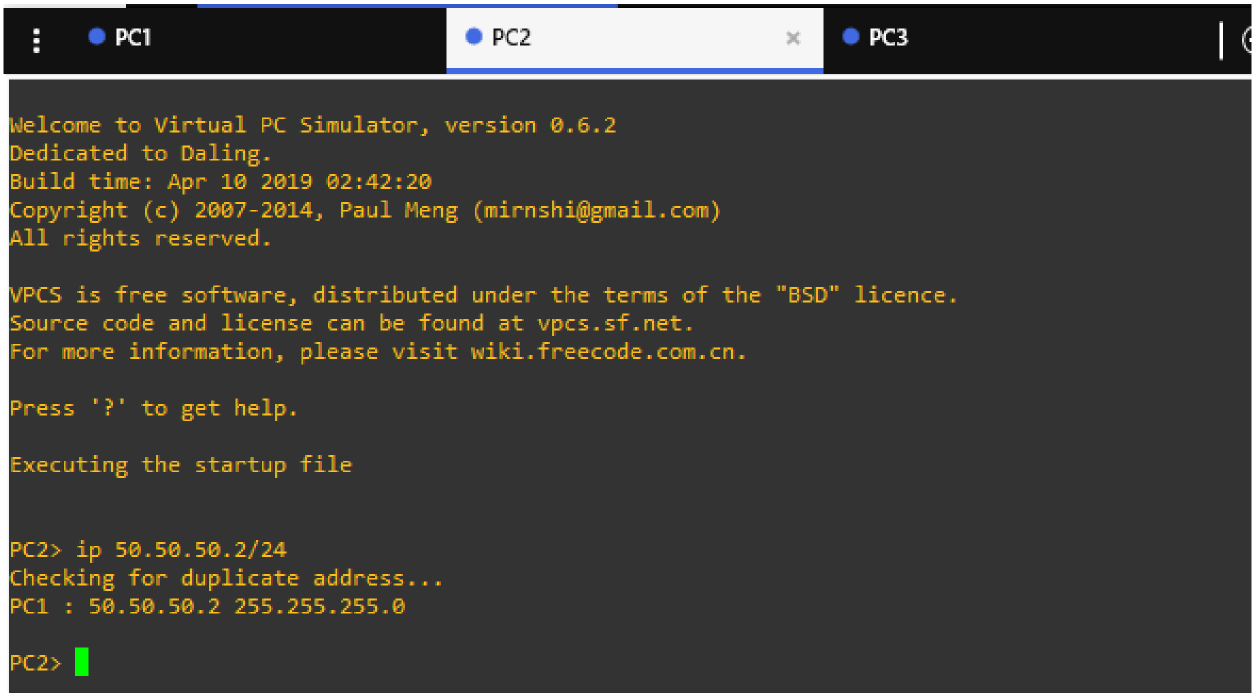
The consoles are now started:

7.2.5. Assigning IP Addresses and subnet masks to PCs:

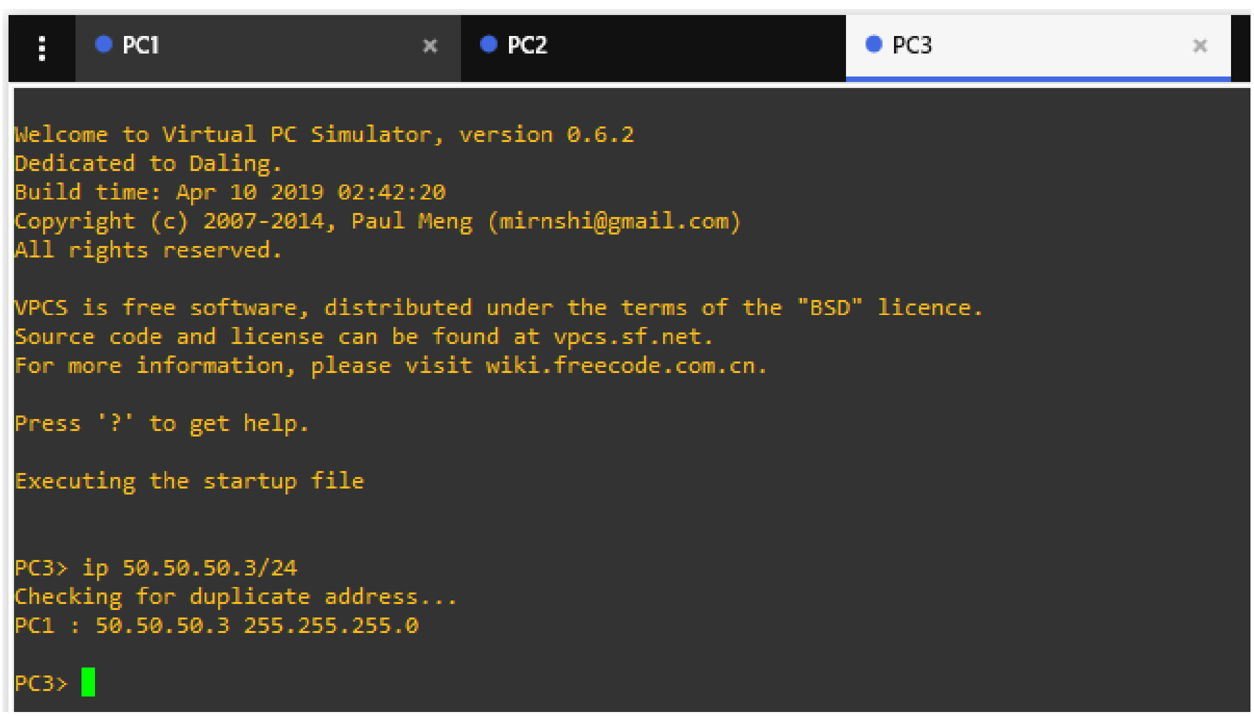
Assigning ip address 50.50.50.1 with a subnet mask of 24 to PC1



Assigning ip address 50.50.50.2 with a subnet mask of 24 to PC2



Assigning ip address 50.50.50.3 with a subnet mask of 24 to PC3



7.2.6. Testing Connectivity:

Testing connectivity from PC1 to PC2

```
PC1> ping 50.50.50.2
84 bytes from 50.50.50.2 icmp_seq=1 ttl=64 time=0.936 ms
84 bytes from 50.50.50.2 icmp_seq=2 ttl=64 time=0.762 ms
84 bytes from 50.50.50.2 icmp_seq=3 ttl=64 time=0.794 ms
84 bytes from 50.50.50.2 icmp_seq=4 ttl=64 time=0.558 ms
84 bytes from 50.50.50.2 icmp_seq=5 ttl=64 time=0.732 ms

PC1>
```

Testing connectivity from PC1 to PC3:

```
PC1> ping 50.50.50.3
84 bytes from 50.50.50.3 icmp_seq=1 ttl=64 time=0.750 ms
84 bytes from 50.50.50.3 icmp_seq=2 ttl=64 time=0.820 ms
84 bytes from 50.50.50.3 icmp_seq=3 ttl=64 time=0.741 ms
84 bytes from 50.50.50.3 icmp_seq=4 ttl=64 time=0.691 ms
84 bytes from 50.50.50.3 icmp_seq=5 ttl=64 time=0.739 ms

PC1>
```

Testing connectivity from PC2 to PC1:

```
PC2> ping 50.50.50.1
84 bytes from 50.50.50.1 icmp_seq=1 ttl=64 time=1.117 ms
84 bytes from 50.50.50.1 icmp_seq=2 ttl=64 time=0.859 ms
84 bytes from 50.50.50.1 icmp_seq=3 ttl=64 time=0.707 ms
84 bytes from 50.50.50.1 icmp_seq=4 ttl=64 time=0.987 ms
84 bytes from 50.50.50.1 icmp_seq=5 ttl=64 time=0.996 ms

PC2>
```

Testing connectivity from PC2 to PC3:

```
PC2> ping 50.50.50.3
84 bytes from 50.50.50.3 icmp_seq=1 ttl=64 time=0.672 ms
84 bytes from 50.50.50.3 icmp_seq=2 ttl=64 time=0.708 ms
84 bytes from 50.50.50.3 icmp_seq=3 ttl=64 time=0.764 ms
84 bytes from 50.50.50.3 icmp_seq=4 ttl=64 time=0.783 ms
84 bytes from 50.50.50.3 icmp_seq=5 ttl=64 time=0.761 ms

PC2>
```

Testing connectivity from PC3 to PC1:

```
PC3> ping 50.50.50.1
84 bytes from 50.50.50.1 icmp_seq=1 ttl=64 time=0.760 ms
84 bytes from 50.50.50.1 icmp_seq=2 ttl=64 time=0.428 ms
84 bytes from 50.50.50.1 icmp_seq=3 ttl=64 time=1.096 ms
84 bytes from 50.50.50.1 icmp_seq=4 ttl=64 time=0.673 ms
84 bytes from 50.50.50.1 icmp_seq=5 ttl=64 time=0.937 ms

PC3>
```

Testing connectivity from PC3 to PC2:

```
PC3> ping 50.50.50.2
84 bytes from 50.50.50.2 icmp_seq=1 ttl=64 time=1.019 ms
84 bytes from 50.50.50.2 icmp_seq=2 ttl=64 time=0.718 ms
84 bytes from 50.50.50.2 icmp_seq=3 ttl=64 time=0.821 ms
84 bytes from 50.50.50.2 icmp_seq=4 ttl=64 time=0.689 ms
84 bytes from 50.50.50.2 icmp_seq=5 ttl=64 time=0.779 ms

PC3>
```

This implies that all the connections are working correctly and start topology is implemented successfully.

7.3. Mesh Topology Using Switches

7.3.1. What is Mesh Topology?

In a mesh topology, every device is connected to every other device in the network.

7.3.2. Characteristics:

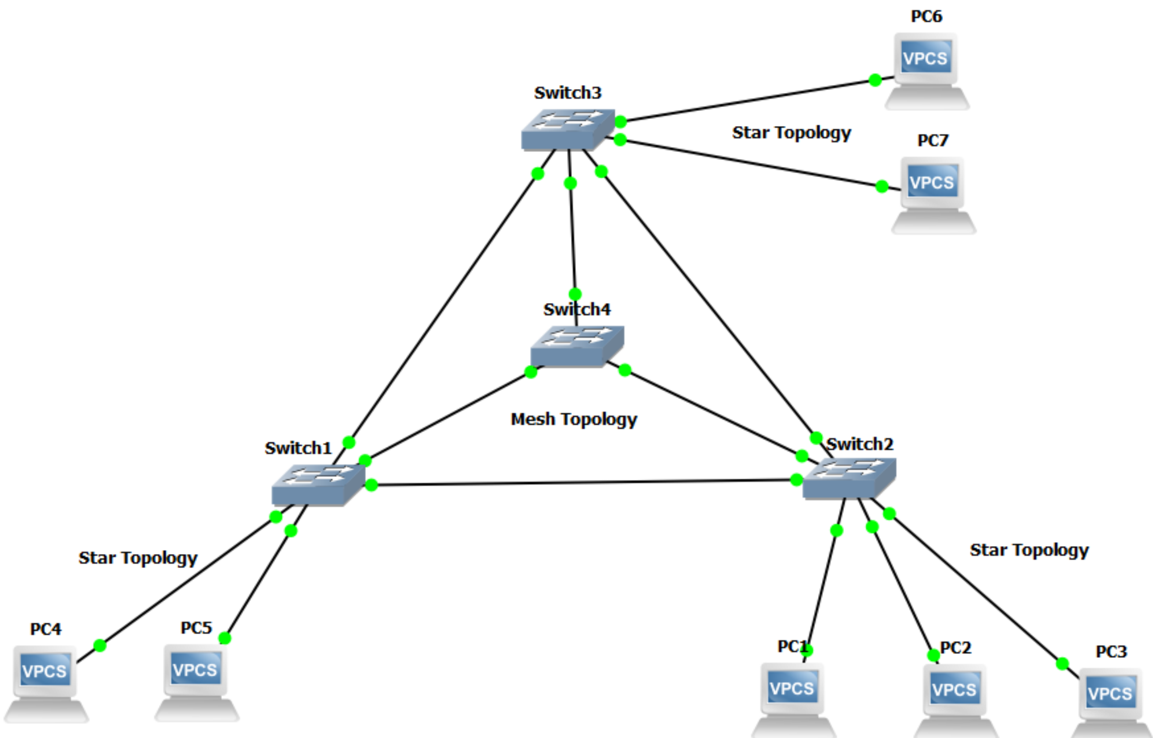
- Mesh topology is more fault tolerant.
- It has high implementation cost because it requires a large number of connections.
- It is mostly used In LANs and its implementation is close to impossible in MANs and WANs.
- Mesh topology is often used in critical applications where reliability is paramount.

7.3.3. How we implemented Mesh Topology in GNS3:

To implement mesh topology, we used 4 ethernet switches and connected each switch to all other ethernet switches, forming a mesh like structure.

We then connected several PCs with some of those switches, so it became like a hybrid topology (using both star and mesh topology), but switches are connected in mesh.

7.3.3.1. Screenshot of Mesh Topology



7.4. Bus Topology using Routers and Switches

7.4.1. What is Bus Topology?

In a bus topology, all devices are connected to a single backbone cable (bus) through interface connectors. The backbone cable has terminators at both ends to prevent signal reflection.

7.4.2. Characteristics?

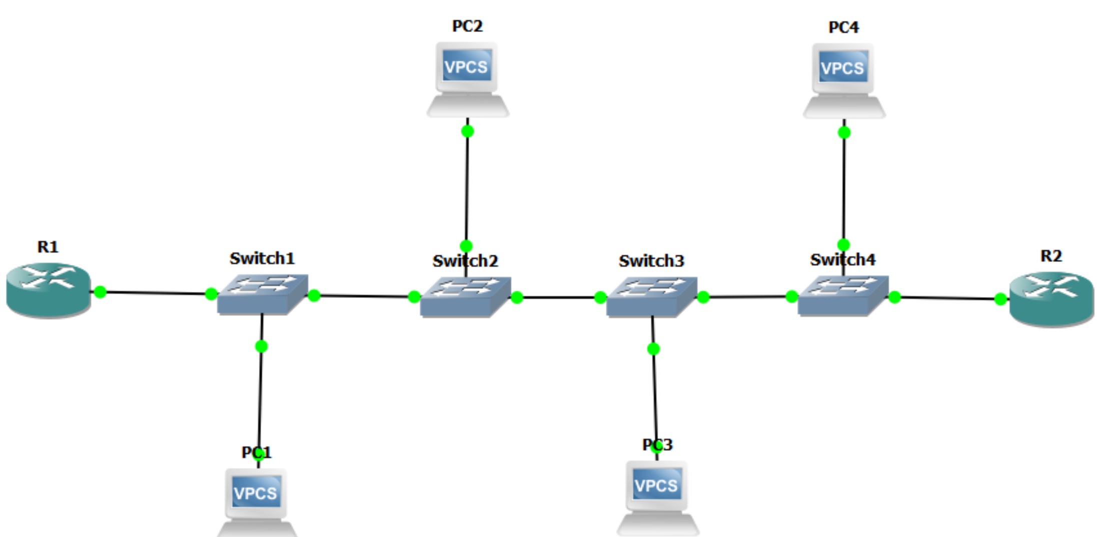
Bus topology is simple and inexpensive to set up.

It is highly affected by cable faults or failures, which can bring down the entire network.

7.4.3. How we implemented Bus Topology in GNS3:

- We first downloaded router image from online resources.
- In this scenario we have used the R1700 router which has 1 FastEthernet fixed port (C1700-MB-1ETH) on its motherboard, 2 subslots for WICs (maximum of 2 Ethernet ports or 4 serial ports), and no Network Module slots.
- We then placed 4 ethernet switches in between those routers which act as “taps”.
- Finally, we connected VPCs to those switches forming a bus topology as shown below:

7.4.3.1. Screenshot of Bus Topology



7.5. Ring Topology using Ethernet Switches

7.5.1. What is Ring Topology?

In a ring topology, each device is connected to two other devices, forming a closed loop.

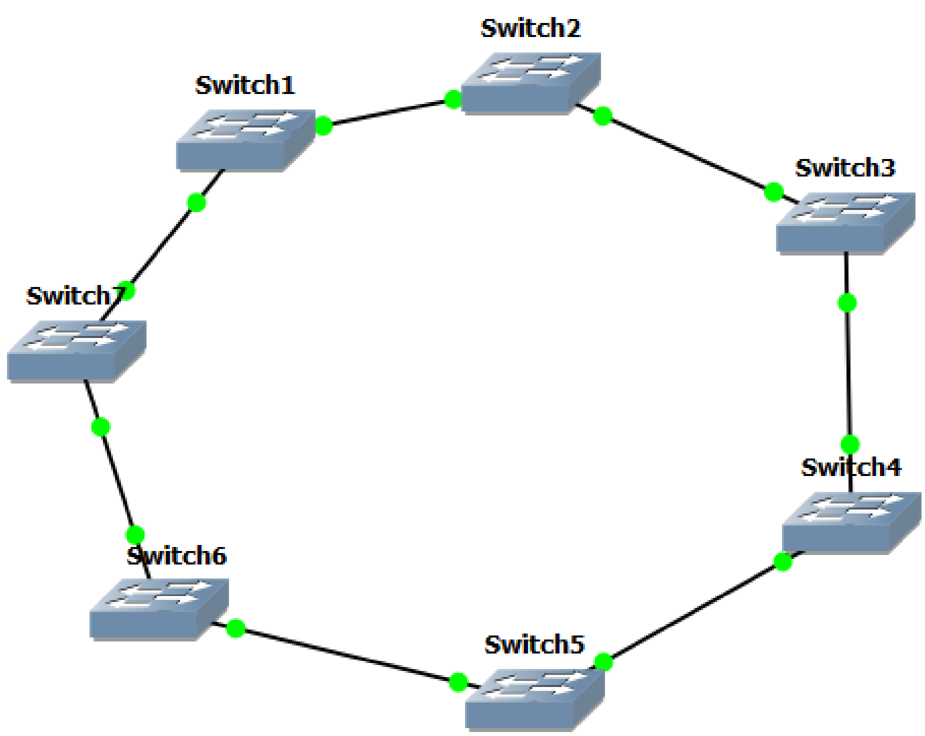
7.5.2. Characteristics:

- Ring topology is simple and easy to install.
- In ring topology, each device has equal access to the network.
- If one device or connection fails, the entire network can be affected.

7.5.3. How we implemented Ring Topology in GNS3:

- We first dragged and dropped 7 ethernet switches from the devices list.
- We then placed them in a circular manner.
- After that, we connected them together forming a circular ring.
- The connection is a closed loop as shown below:

7.5.3.1. Screenshot of Ring Topology



8. Question No. 6:

Design a complex networking architecture for an organization.

8.1. Explain in detail the Network address, Protocols that are used.

Solution:

8.1.1. Network Address:

In the given network setup, each PC has its own unique IP address within the same subnet. Details are given below:

PC1: IP Address - 50.50.50.1, Subnet Mask - 255.255.255.0 (/24)

PC2: IP Address - 50.50.50.2, Subnet Mask - 255.255.255.0 (/24)

PC3: IP Address - 50.50.50.3, Subnet Mask - 255.255.255.0 (/24)

Network Address: 50.50.50.0 with a subnet mask of 255.255.255.0 (/24). This means the first 24 bits of the IP address are for the network part, and the last 8 bits are for the host part. The IP range in this network is from 50.50.50.1 to 50.50.50.254.

Subnet Mask: 255.255.255.0 (/24).

8.1.2. Protocols Used:

These are the protocols used in this network:

8.1.2.1. Internet Protocol (IP):

Version: IPv4.

Purpose: It handles the addressing and routing of data packets so they can move from one computer to another across networks.

8.1.2.2. Address Resolution Protocol (ARP):

Purpose: ARP matches IP addresses to physical MAC (Media Access Control) addresses. When a PC needs to communicate with another PC on the same network, it uses ARP to find out the MAC address of the destination IP address.

8.1.2.3. Internet Control Message Protocol (ICMP):

Purpose: ICMP is used for sending error messages and operational information. Tools like ping use ICMP to check if another computer is reachable on the network.

8.1.2.4. Transmission Control Protocol (TCP):

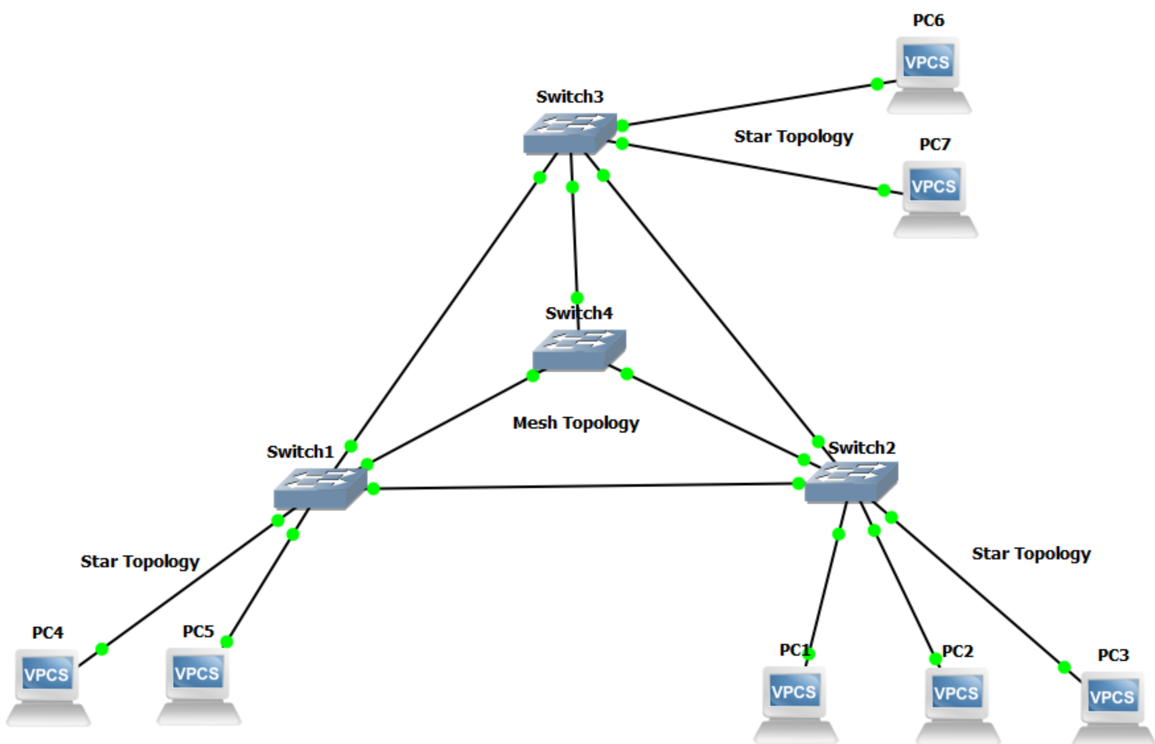
Purpose: TCP ensures that data is reliably transmitted between computers. It's used for applications where it's important that data arrives correctly, like web browsing or email.

8.1.2.5. User Datagram Protocol (UDP):

Purpose: UDP allows for faster data transmission without checking for errors. It's used for activities where speed is more important than reliability, like streaming video or online gaming.

8.2. Use the packet sniffing tool (wire-shark) to analyse the packet and flow of plackets in the network.

Network Design



8.2.1. Assigning IP addresses to nodes:

PC1

```
PC1> ip 50.50.50.1/24
Checking for duplicate address...
PC1 : 50.50.50.1 255.255.255.0
```

PC2

```
PC2> ip 50.50.50.2/24
Checking for duplicate address...
PC1 : 50.50.50.2 255.255.255.0
```

PC3

```
PC3> ip 50.50.50.3/24
Checking for duplicate address...
PC1 : 50.50.50.3 255.255.255.0
```

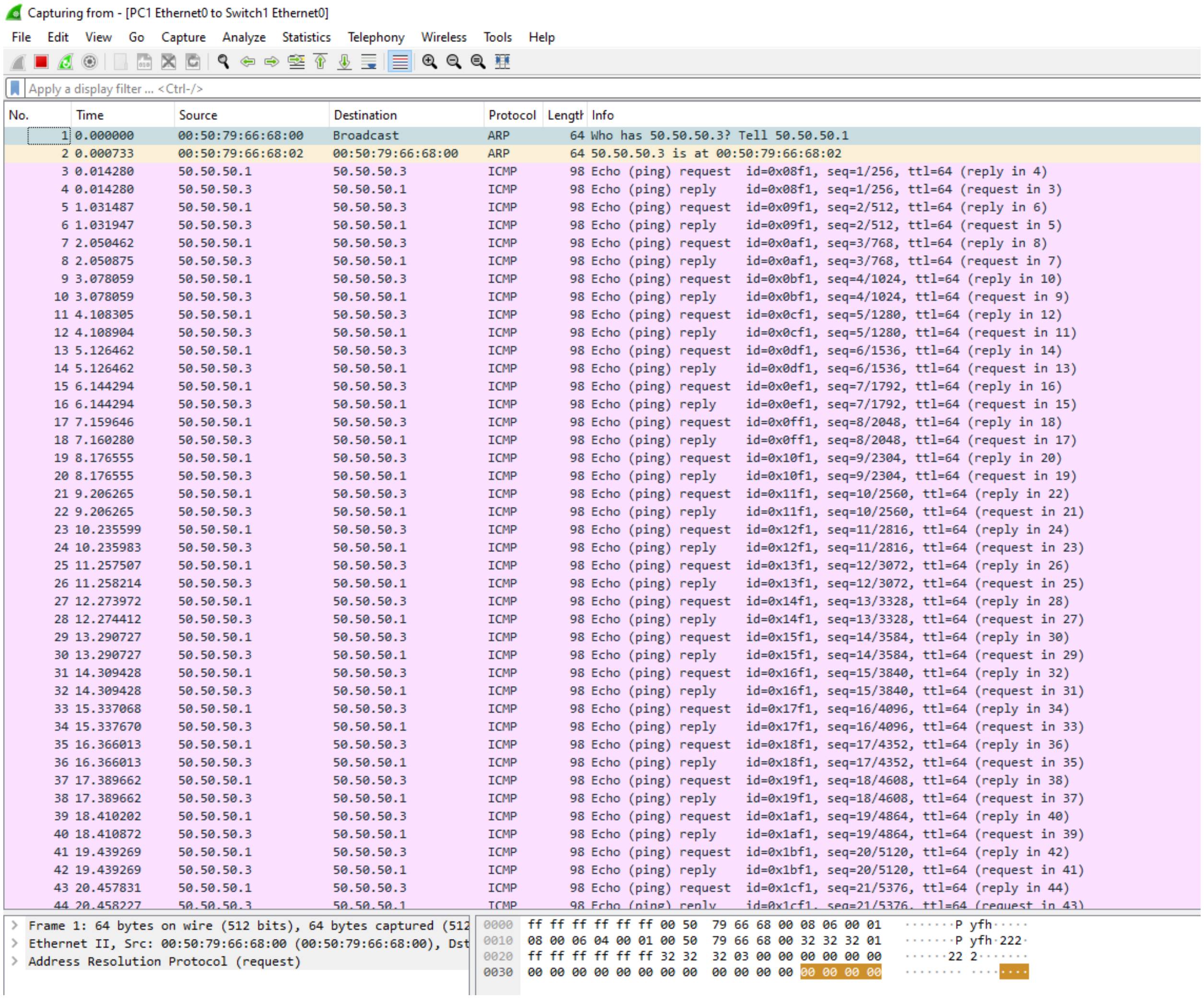
8.2.2. Pinging:

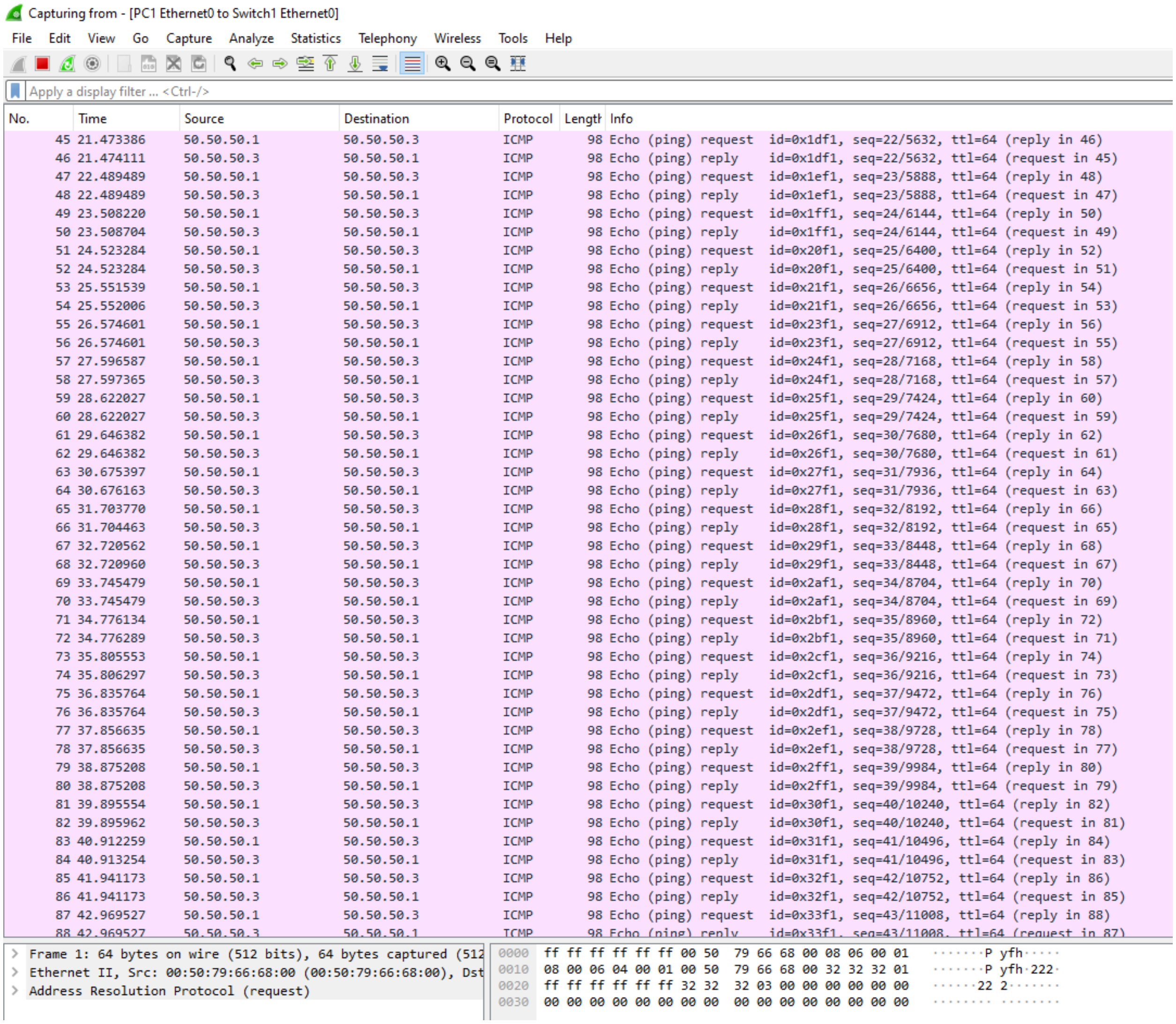
Pinging from PC1 to PC3

```
PC1> ip 50.50.50.1/24
Checking for duplicate address...
PC1 : 50.50.50.1 255.255.255.0

PC1> ping 50.50.50.3 -t
84 bytes from 50.50.50.3 icmp_seq=1 ttl=64 time=0.757 ms
84 bytes from 50.50.50.3 icmp_seq=2 ttl=64 time=0.840 ms
84 bytes from 50.50.50.3 icmp_seq=3 ttl=64 time=0.664 ms
84 bytes from 50.50.50.3 icmp_seq=4 ttl=64 time=0.471 ms
84 bytes from 50.50.50.3 icmp_seq=5 ttl=64 time=1.771 ms
84 bytes from 50.50.50.3 icmp_seq=6 ttl=64 time=0.894 ms
84 bytes from 50.50.50.3 icmp_seq=7 ttl=64 time=0.756 ms
84 bytes from 50.50.50.3 icmp_seq=8 ttl=64 time=0.378 ms
84 bytes from 50.50.50.3 icmp_seq=9 ttl=64 time=0.720 ms
84 bytes from 50.50.50.3 icmp_seq=10 ttl=64 time=0.978 ms
84 bytes from 50.50.50.3 icmp_seq=11 ttl=64 time=0.779 ms
84 bytes from 50.50.50.3 icmp_seq=12 ttl=64 time=0.580 ms
84 bytes from 50.50.50.3 icmp_seq=13 ttl=64 time=0.621 ms
84 bytes from 50.50.50.3 icmp_seq=14 ttl=64 time=0.685 ms
84 bytes from 50.50.50.3 icmp_seq=15 ttl=64 time=0.691 ms
84 bytes from 50.50.50.3 icmp_seq=16 ttl=64 time=0.723 ms
84 bytes from 50.50.50.3 icmp_seq=17 ttl=64 time=0.663 ms
84 bytes from 50.50.50.3 icmp_seq=18 ttl=64 time=0.721 ms
```

8.2.3. Capturing Packets using Wireshark





No.	Time	Source	Destination	Protocol	Length	Info
45	21.473386	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x1df1, seq=22/5632, ttl=64 (reply in 46)
46	21.474111	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x1df1, seq=22/5632, ttl=64 (request in 45)
47	22.489489	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x1ef1, seq=23/5888, ttl=64 (reply in 48)
48	22.489489	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x1ef1, seq=23/5888, ttl=64 (request in 47)
49	23.508220	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x1ff1, seq=24/6144, ttl=64 (reply in 50)
50	23.508704	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x1ff1, seq=24/6144, ttl=64 (request in 49)
51	24.523284	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x20f1, seq=25/6400, ttl=64 (reply in 52)
52	24.523284	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x20f1, seq=25/6400, ttl=64 (request in 51)
53	25.551539	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x21f1, seq=26/6656, ttl=64 (reply in 54)
54	25.552006	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x21f1, seq=26/6656, ttl=64 (request in 53)
55	26.574601	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x23f1, seq=27/6912, ttl=64 (reply in 56)
56	26.574601	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x23f1, seq=27/6912, ttl=64 (request in 55)
57	27.596587	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x24f1, seq=28/7168, ttl=64 (reply in 58)
58	27.597365	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x24f1, seq=28/7168, ttl=64 (request in 57)
59	28.622027	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x25f1, seq=29/7424, ttl=64 (reply in 60)
60	28.622027	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x25f1, seq=29/7424, ttl=64 (request in 59)
61	29.646382	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x26f1, seq=30/7680, ttl=64 (reply in 62)
62	29.646382	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x26f1, seq=30/7680, ttl=64 (request in 61)
63	30.675397	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x27f1, seq=31/7936, ttl=64 (reply in 64)
64	30.676163	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x27f1, seq=31/7936, ttl=64 (request in 63)
65	31.703770	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x28f1, seq=32/8192, ttl=64 (reply in 66)
66	31.704463	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x28f1, seq=32/8192, ttl=64 (request in 65)
67	32.720562	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x29f1, seq=33/8448, ttl=64 (reply in 68)
68	32.720960	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x29f1, seq=33/8448, ttl=64 (request in 67)
69	33.745479	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x2af1, seq=34/8704, ttl=64 (reply in 70)
70	33.745479	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x2af1, seq=34/8704, ttl=64 (request in 69)
71	34.776134	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x2bf1, seq=35/8960, ttl=64 (reply in 72)
72	34.776289	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x2bf1, seq=35/8960, ttl=64 (request in 71)
73	35.805553	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x2cf1, seq=36/9216, ttl=64 (reply in 74)
74	35.806297	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x2cf1, seq=36/9216, ttl=64 (request in 73)
75	36.835764	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x2df1, seq=37/9472, ttl=64 (reply in 76)
76	36.835764	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x2df1, seq=37/9472, ttl=64 (request in 75)
77	37.856635	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x2ef1, seq=38/9728, ttl=64 (reply in 78)
78	37.856635	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x2ef1, seq=38/9728, ttl=64 (request in 77)
79	38.875208	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x2ff1, seq=39/9984, ttl=64 (reply in 80)
80	38.875208	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x2ff1, seq=39/9984, ttl=64 (request in 79)
81	39.895554	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x30f1, seq=40/10240, ttl=64 (reply in 82)
82	39.895962	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x30f1, seq=40/10240, ttl=64 (request in 81)
83	40.912259	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x31f1, seq=41/10496, ttl=64 (reply in 84)
84	40.913254	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x31f1, seq=41/10496, ttl=64 (request in 83)
85	41.941173	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x32f1, seq=42/10752, ttl=64 (reply in 86)
86	41.941173	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x32f1, seq=42/10752, ttl=64 (request in 85)
87	42.969527	50.50.50.1	50.50.50.3	ICMP	98	Echo (ping) request id=0x33f1, seq=43/11008, ttl=64 (reply in 88)
88	42.969527	50.50.50.3	50.50.50.1	ICMP	98	Echo (ping) reply id=0x33f1, seq=43/11008, ttl=64 (request in 87)

> Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512
> Ethernet II, Src: 00:50:79:66:68:00 (00:50:79:66:68:00), Dst
> Address Resolution Protocol (request)

```
0000  ff ff ff ff ff ff 00 50  79 66 68 00 08 06 00 01   ·······P yfh·····
0010  08 00 06 04 00 01 00 50  79 66 68 00 32 32 32 01   ·······P yfh·222·
0020  ff ff ff ff ff ff 32 32  32 03 00 00 00 00 00 00   ······22 2·······
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

9. Evaluation:

To evaluate the effectiveness of GNS3 in network simulation, several topologies were implemented and tested. The topologies included star, mesh, bus, and ring configurations, each chosen for their unique characteristics and common use cases in network design.

9.1. Star Topology:

- Setup: Involved connecting multiple PCs to a single switch.
- Characteristics: Easy to set up and manage, but with a single point of failure.

- Results: Successfully demonstrated with all devices able to communicate after assigning IP addresses and testing connectivity.

9.2. Mesh Topology:

- Setup: Required connecting every device to every other device, providing high redundancy.
- Characteristics: High fault tolerance but complex to set up.
- Results: Verified through multiple connectivity tests, showing robust network performance.

9.3. Bus Topology:

- Setup: Involved connecting devices along a single communication line.
- Characteristics: Simple and cost-effective, but prone to collisions.
- Results: Implemented using routers and switches, with successful data transmission observed.

9.4. Ring Topology:

- Setup: Devices connected in a circular manner, each linked to two other devices.
- Characteristics: Equal access to the network but the entire network can be affected by a single failure.
- Results: Effectively demonstrated, with all nodes maintaining connectivity.

10. Conclusion:

This report demonstrates the practical application of GNS3 in designing and testing various network topologies. The successful implementation of star, mesh, bus, and ring topologies highlights GNS3's capability as a versatile and powerful network simulation tool. Through this study, it is evident that GNS3 provides a valuable platform for learning and experimenting with network configurations, offering a realistic and flexible environment for both educational and professional purposes. Future work could involve more complex network setups and the exploration of additional features and integrations available in GNS3.

11. References:

- Neumann, Jason C. *The book of GNS3: build virtual network labs using Cisco, Juniper, and more*. No Starch Press, 2015.

- *Dynamips. (2024, March 5). In Wikipedia. Retrieved March 5, 2024, from https://en.wikipedia.org/wiki/Dynamips*

- *GNS3 Documentation. GNS3. Retrieved March 5, 2024, from https://docs.gns3.com/*

- NiKESHALA, JB. (2023, August 18). *Cisco Packet Tracer vs GNS3: Which Network Simulator is Right for You? LinkedIn. Retrieved March 5, 2024, from https://www.linkedin.com/pulse/cisco-packet-tracer-vs-gns3-which-network-simulator-right-nikeshala#:~:text=As%20a%20beginner%2C%20Cisco%20Packet,also%20more%20complex%20to%20use.*

- Molenaar, Rene. (2011, May 23). *GNS3 Introduction. GNS3 Vault. Retrieved March 5, 2024, from https://gns3vault.com/faq/gns3-introduction#:~:text=GNS3%20is%20a%20Graphical%20frontend,what%20it%20is%20meant%20for.*