LAB: 1

AIM: Study about Network Cabling

Theory:

In this exercise, we used Cat6 cables, which are a type of twisted-pair cable designed to support higher bandwidths than Cat5e cables. The cable has eight individual wires twisted into four pairs. Color coding is important to ensure proper connectivity and data transmission. There are two main wiring standards for Ethernet cables:

- T568A
- T568B

Each of these standards determines the order in which the eight wires are arranged. Consistency in wiring standards is crucial for proper network functioning.

Color Coding for T568B Standard:

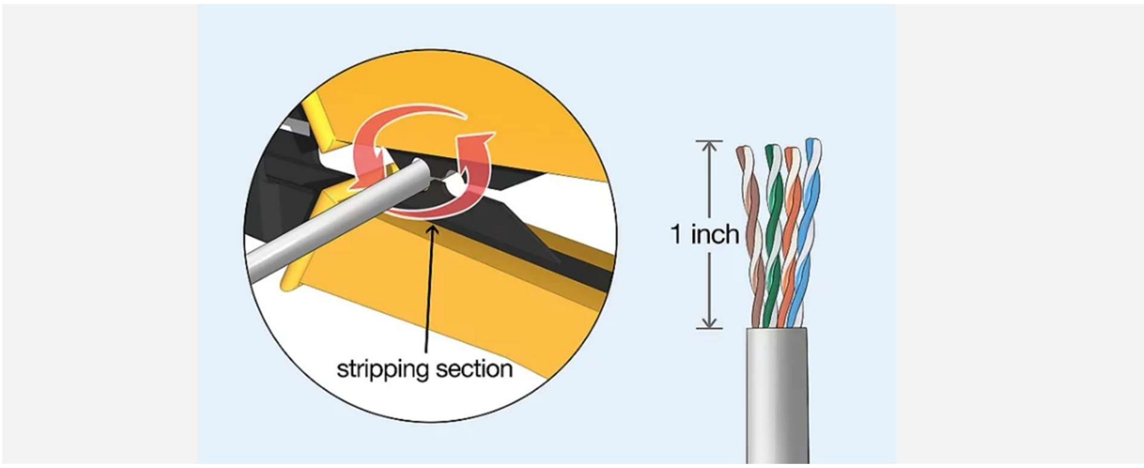
1. White/Orange
2. Orange
3. White/Green
4. Blue
5. White/Blue
6. Green
7. White/Brown
8. Brown

Color Coding for T568A Standard:

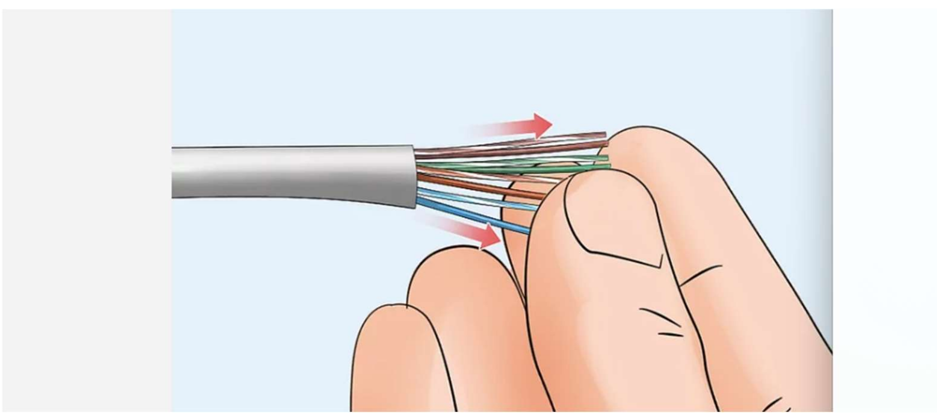
1. White/Green
2. Green
3. White/Orange
4. Blue
5. White/Blue
6. Orange
7. White/Brown
8. Brown

Procedure:

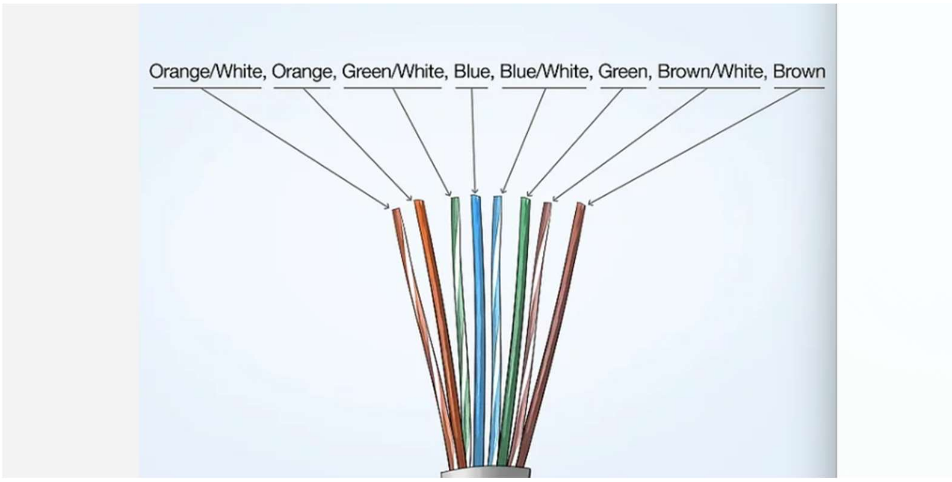
1. Uncovering the Cable:
   - Strip about 1-2 inches of the outer insulation from the Cat6 cable using a cable stripper or cutter, revealing the four twisted pairs of wires inside.



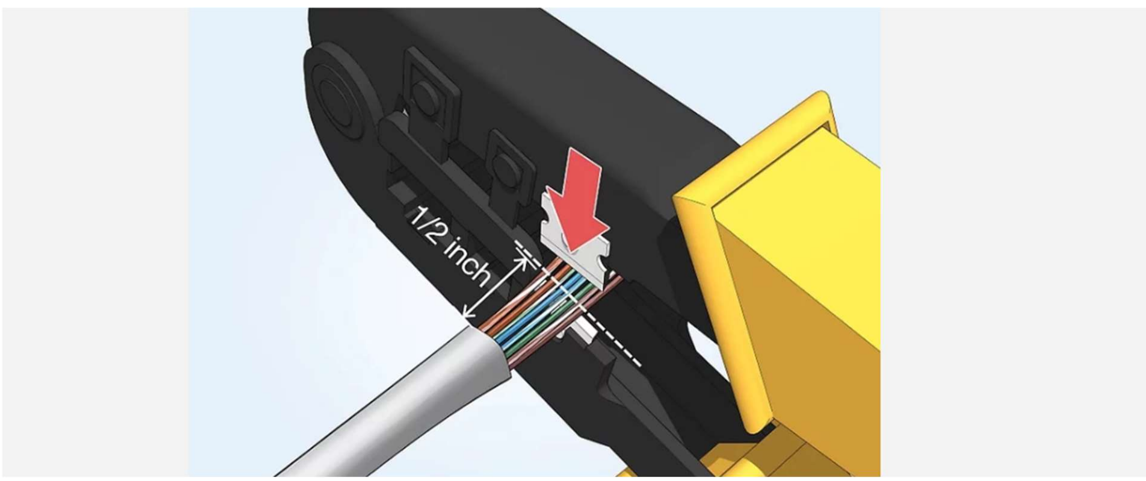
2. Sorting and Straightening:
   - Untwist the pairs of wires and straighten them as much as possible for easy insertion into the RJ45 connector.



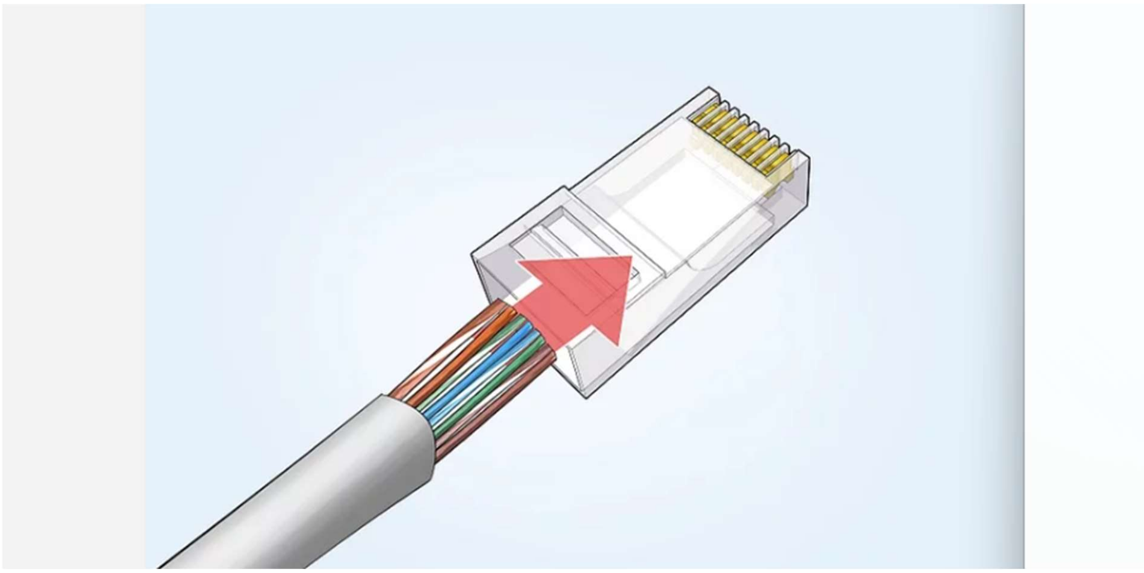
3. Color Coding:
   - Arrange the wires according to the T568B color code (or T568A if chosen).
   - Make sure to align the wires in the correct order before insertion.



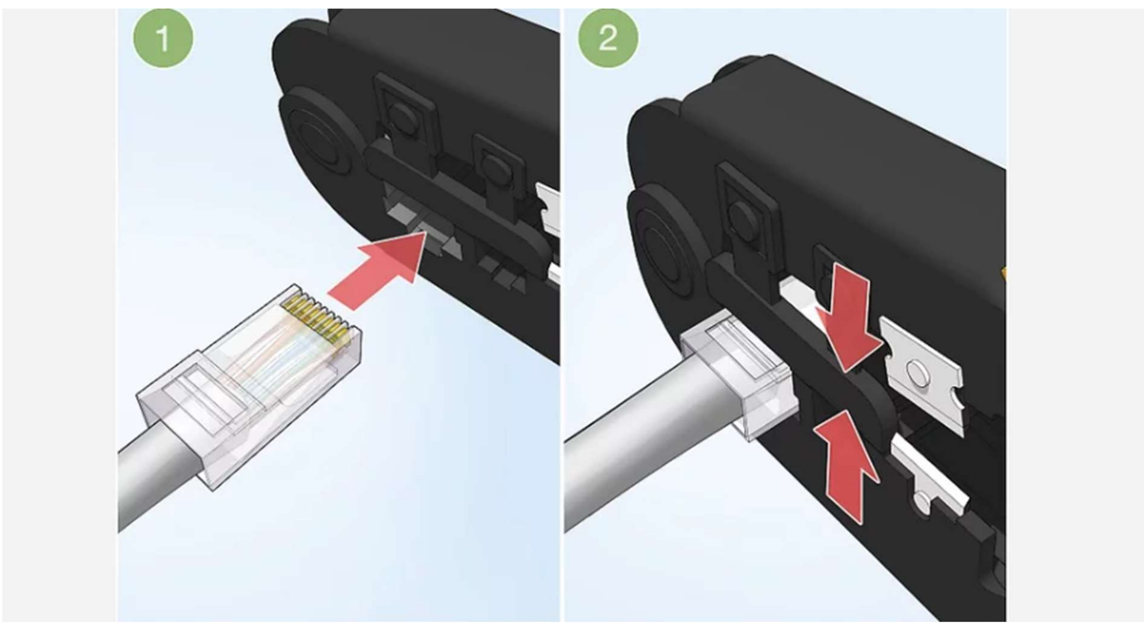
4. Trimming the Wires:
    - Trim the ends of the wires evenly to ensure they fit neatly into the RJ45 connector.



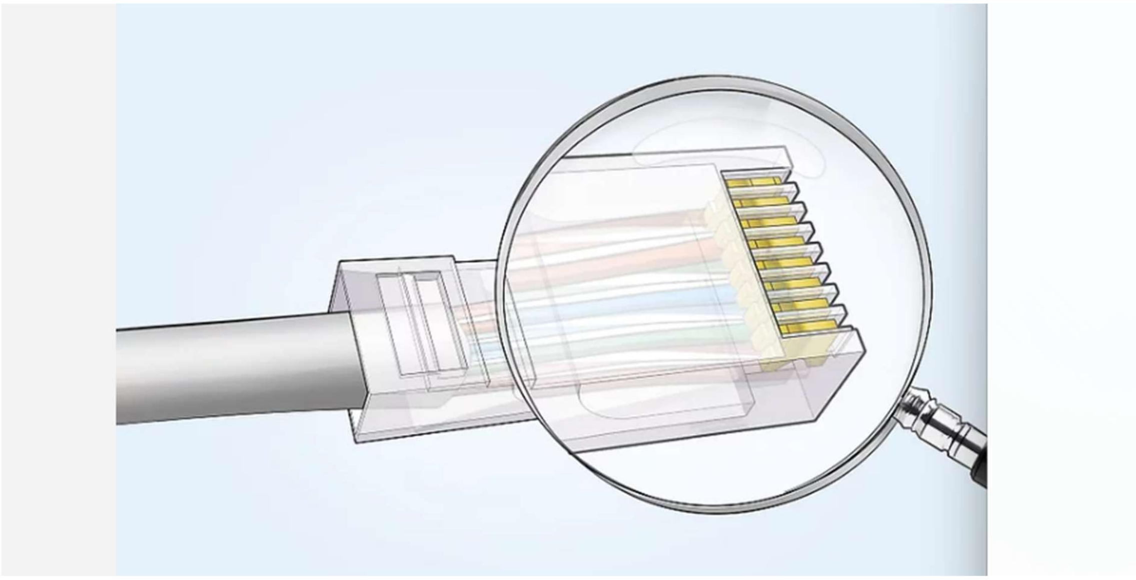
5. Inserting into the RJ45 Connector:
    - Insert the sorted wires into the RJ45 connector, ensuring that each wire goes into the correct channel and reaches the metal contacts.



6. Clamping (Crimping):
   - Place the RJ45 connector into the crimping tool and apply firm pressure to clamp the connector onto the cable, securing the wires and making electrical contact.



7. Testing the Cable:
   - Use a cable tester to verify that the cable has been properly wired and that all connections are working correctly.



Conclusion:

In this exercise, we learned how to correctly identify the color coding for Cat6 cables and apply the T568B wiring standard. We successfully built a functional Ethernet cable by following the correct procedure, from uncovering the cable to crimping the RJ45 connector.

Discussion:

During the process, one of the main difficulties we faced was untwisting and straightening the wires, as they tend to tangle or become difficult to align. Sorting the wires according to the color coding standard also required careful attention to avoid mistakes. Another challenge was clamping the connector; ensuring all the wires were fully inserted and maintaining alignment while using the crimping tool was crucial. We overcame these difficulties by taking extra care to straighten the wires properly and double-checking the color arrangement before clamping. This careful approach helped us complete the task successfully.

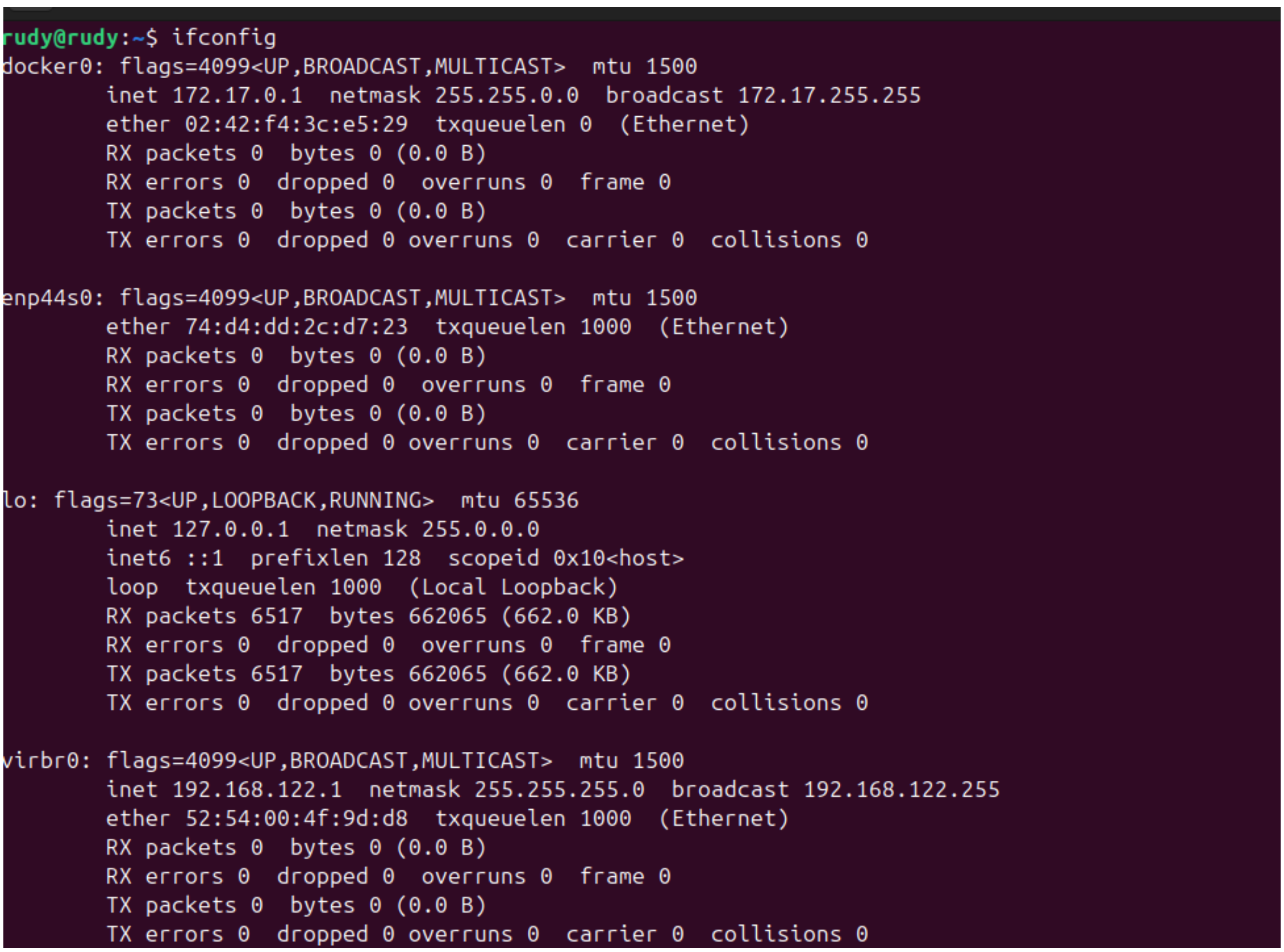
LAB: 2

Theory:

Networking commands are essential tools used by system administrators and network engineers to troubleshoot, configure, and monitor network-related issues. These commands help in viewing the network configuration of a machine, testing network connectivity, identifying issues, and more. Whether it's determining the IP address, checking connectivity to a remote server, or resolving domain names, these commands are invaluable for managing both local and wide-area networks.

Basic Networking Commands:

1. ipconfig (Windows) / ifconfig (Linux/macOS)
   - Description: Displays the network configuration of the system, including IP address, subnet mask, and gateway.
   - Syntax:
     - Windows: ipconfig
     - Linux/macOS: ifconfig



```
rudy@rudy:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:f4:3c:e5:29  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp44s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 74:d4:dd:2c:d7:23  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6517  bytes 662065 (662.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6517  bytes 662065 (662.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:4f:9d:d8  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. ping
    - Description: Sends ICMP echo requests to a specific host to test network connectivity and measure the round-trip time.
    - Syntax: ping [hostname/IP address]
        - Example: ping theodinproject.com

```
rudy@rudy:~$ ping theodinproject.com
PING theodinproject.com (2606:4700:20::ac43:494a) 56 data bytes
64 bytes from 2606:4700:20::ac43:494a: icmp_seq=1 ttl=59 time=9.27 ms
64 bytes from 2606:4700:20::ac43:494a: icmp_seq=2 ttl=59 time=5.64 ms
64 bytes from 2606:4700:20::ac43:494a: icmp_seq=3 ttl=59 time=10.7 ms
^C
--- theodinproject.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 5.639/8.526/10.670/2.120 ms
```

3. tracert (Windows) / traceroute (Linux/macOS)
    - Description: Traces the route packets take from your computer to a destination. It shows each hop along the way and helps identify where delays or failures occur.
    - Syntax:
        - Windows: tracert [hostname/IP address]
        - Linux/macOS: traceroute [hostname/IP address]
        - Example: tracert theodinproject.com

```
rudy@rudy:~$ tracepath theodinproject.com
 1?: [LOCALHOST]                      0.013ms pmtu 1500
 1:  2400-1A00-B020.ip6.firenet.com.np                     2.798ms
 1:  2400-1A00-B020.ip6.firenet.com.np                     5.716ms
 2:  2400-1A00-B1A2.ip6.firenet.com.np                     6.720ms
 3:  ???                                                  12.104ms
 4:  ???                                                   6.676ms
^C
rudy@rudy:~$
```

4. nslookup
   - Description: Queries DNS to find the IP address of a domain or the domain associated with an IP address.
   - Syntax: nslookup [hostname/IP address]
     - Example: nslookup : theodinproject.com

```
rudy@rudy:~$ nslookup theodinproject.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   theodinproject.com
Address: 172.67.73.74
Name:   theodinproject.com
Address: 104.26.8.228
Name:   theodinproject.com
Address: 104.26.9.228
Name:   theodinproject.com
Address: 2606:4700:20::ac43:494a
Name:   theodinproject.com
Address: 2606:4700:20::681a:9e4
Name:   theodinproject.com
Address: 2606:4700:20::681a:8e4
```

5. netstat
   - Description: Displays active network connections, listening ports, and network statistics. Useful for identifying open connections and troubleshooting.
   - Syntax: netstat

```
rudy@rudy:~$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 REDACTED_IPREDACTED_PORT         REDACTED_IP:*             LISTEN
tcp        0      0 REDACTED_IPREDACTED_PORT        REDACTED_IP:*             LISTEN
tcp        0      0 REDACTED_IPREDACTED_PORT          REDACTED_IP:*             LISTEN
tcp        0      0 REDACTED_IPREDACTED_PORT         REDACTED_IP:*             LISTEN
tcp        0      0 REDACTED_IPREDACTED_PORT          REDACTED_IP:*             LISTEN
tcp        0      0 REDACTED_IPREDACTED_PORT          REDACTED_IP:*             LISTEN
tcp        0      0 REDACTED_IPREDACTED_PORT         REDACTED_IP:*             LISTEN
tcp6       0      0 :REDACTED_PORTREDACTED_PORT                :::*                  LISTEN
udp        0      0 REDACTED_IPREDACTED_PORT        REDACTED_IP:*
udp        0      0 REDACTED_IPREDACTED_PORT          REDACTED_IP:*
udp        0      0 REDACTED_IPREDACTED_PORT          REDACTED_IP:*
udp        0      0 REDACTED_IPREDACTED_PORT            REDACTED_IP:*
udp        0      0 REDACTED_IPREDACTED_PORT           REDACTED_IP:*
udp        0      0 REDACTED_IPREDACTED_PORT          REDACTED_IP:*
udp        0      0 REDACTED_IPREDACTED_PORT           REDACTED_IP:*
udp6       0      0 ::REDACTED_PORT              :::*
udp6       0      0 ::REDACTED_PORT               :::*
```

```
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     16969    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     17836
unix  2      [ ]         DGRAM      CONNECTED     10910
unix  3      [ ]         STREAM     CONNECTED     107188   /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     27756
unix  3      [ ]         STREAM     CONNECTED     25770    /home/rudy/.cache/ibus/dbus-6ewYxiW0
unix  3      [ ]         STREAM     CONNECTED     102219
unix  3      [ ]         STREAM     CONNECTED     22723    /run/user/1000/pipewire-0
unix  3      [ ]         STREAM     CONNECTED     14881
unix  3      [ ]         STREAM     CONNECTED     11687    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     11683    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     19037
unix  3      [ ]         STREAM     CONNECTED     117355
unix  3      [ ]         STREAM     CONNECTED     17927    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     104743
unix  3      [ ]         STREAM     CONNECTED     19049    /run/user/1000/pipewire-0
unix  3      [ ]         STREAM     CONNECTED     25758    /run/user/1000/at-spi/bus_0
unix  3      [ ]         STREAM     CONNECTED     27737    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     16913    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     17247
unix  3      [ ]         STREAM     CONNECTED     19033    /run/user/1000/at-spi/bus_0
unix  2      [ ]         STREAM     CONNECTED     14934
unix  3      [ ]         STREAM     CONNECTED     123072
unix  3      [ ]         STREAM     CONNECTED     77141
unix  3      [ ]         STREAM     CONNECTED     13755
unix  3      [ ]         STREAM     CONNECTED     17156
unix  3      [ ]         STREAM     CONNECTED     8614     /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     71929
unix  3      [ ]         STREAM     CONNECTED     25749    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     22682    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     27687
unix  3      [ ]         STREAM     CONNECTED     15261    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     64716    /run/user/1000/pipewire-0
unix  3      [ ]         STREAM     CONNECTED     23894    /run/systemd/journal/stdout
```

6. arp
   - Description: Displays or modifies the ARP (Address Resolution Protocol) table, which maps IP addresses to physical MAC addresses.
   - Syntax: arp -a (to view the ARP table)

```
rudy@rudy:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   5c:8c:30:5b:4e:2c   C                     wlp0s
20f3
192.168.1.74             ether   f2:b2:9c:08:61:e6   C                     wlp0s
20f3
```

7. route
   - Description: Displays and manipulates the IP routing table, which determines the path that network traffic takes.
   - Syntax: route

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlp0s20f
3
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 wlp0s20f
3
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

8. hostname
    - Description: Displays the current hostname of the computer or allows the hostname to be set.
    - Syntax: hostname

```
rudy@rudy:~$ hostname
rudy
```

9. telnet
    - Description: Establishes a Telnet connection to a remote machine. Telnet is a protocol for remote command-line access.
    - Syntax: telnet [hostname/IP address] [port]
        - Example: telnet 192.168.1.1 80

```
rudy@rudy: ~                                   rudy@rudy: ~
rudy@rudy:~$ telnet 127.0.0.1 23
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```

10. curl

- Description: Transfers data from or to a server using various protocols. It's often used for testing APIs and web servers.
- Syntax: curl [URL]
    - Example: curl -[flag] [method] [address]

```
rudy@rudy: ~                                   rudy@rudy: ~
rudy@rudy:~$ curl https://jsonplaceholder.typicode.com/posts/1
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehe
nderit",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nr
eprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem evenie
t architecto"
}rudy@rudy:~$
```

Conclusion:

By understanding and using these basic networking commands, we can perform crucial tasks such as diagnosing connectivity issues, analyzing network performance, and resolving DNS queries. These tools are fundamental in managing and troubleshooting both simple and complex network environments.

8. hostname
    - Description: Displays the current hostname of the computer or allows the hostname to be set.
    - Syntax: hostname

```
rudy@rudy:~$ hostname
rudy
```

9. telnet
    - Description: Establishes a Telnet connection to a remote machine. Telnet is a protocol for remote command-line access.
    - Syntax: telnet [hostname/IP address] [port]
        - Example: telnet 192.168.1.1 80

```
rudy@rudy: ~                                  rudy@rudy: ~
rudy@rudy:~$ telnet 127.0.0.1 23
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```

10. curl

- Description: Transfers data from or to a server using various protocols. It's often used for testing APIs and web servers.
- Syntax: curl [URL]
    - Example: curl -[flag] [method] [address]

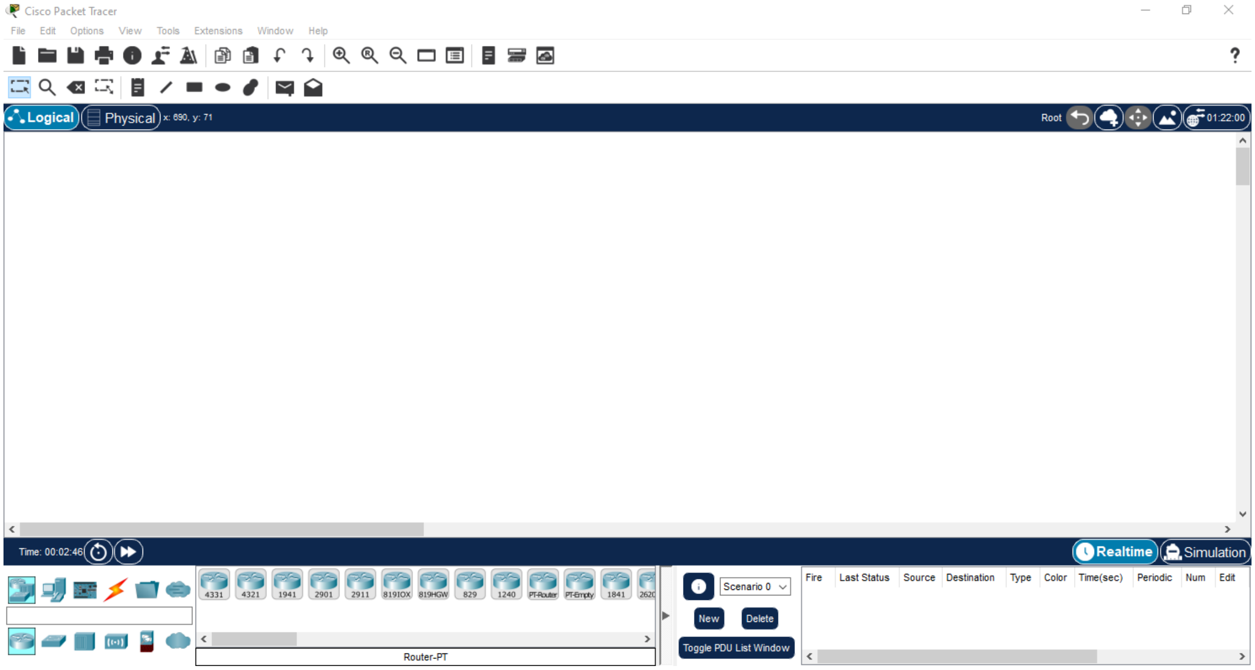
```
rudy@rudy: ~                                  rudy@rudy: ~
rudy@rudy:~$ curl https://jsonplaceholder.typicode.com/posts/1
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehe
nderit",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nr
eprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem evenie
t architecto"
}rudy@rudy:~$
```

Conclusion:

By understanding and using these basic networking commands, we can perform crucial tasks such as diagnosing connectivity issues, analyzing network performance, and resolving DNS queries. These tools are fundamental in managing and troubleshooting both simple and complex network environments.

Lab 3: Introduction to Packet Tracer and Basic LAN setup.

Introduction to Packet Tracer



Cisco Packet Tracer is a powerful network simulation tool that allows users to design, configure, and troubleshoot networks in a virtual environment. It provides a hands-on platform for both students and professionals to gain practical experience with network protocols, devices, and configurations without the need for physical hardware.

Workspace for CISCO Packet Tracer

Logical: The logical workspace shows the logical network topology that is built by the user. It displays the connecting, placing, and clustering of virtual network devices.

Physical: In the physical workspace, we can see the physical implementation of the logical network. It also shows how the network devices such as switches, routers, and hosts are connected in a real network topology.

Features of CISCO Packet Tracker

There are some features that are provided by the CISCO packet tracker:

1. CISCO packet tracker supports the multi-user system that allows any user to connect in different topologies across different computer networks. By using this feature, the teacher assigns different tasks to different students.
2. We can also remove the capabilities of the CISCO packet tracker with the help of an API. This feature is also provided by the CISCO packet tracker.
3. We can also remove the special features like accessibility, gaming, assessment delivery, and interaction with the real world from the CISCO packet tracker.
4. We can also simulate the configuration related to routers, and this can be accessed anywhere.
5. We can access this configuration with unlimited devices.
6. It also provides a self-placed and interactive environment.
7. The Enhanced Physical Mode transports you to a virtual lab where you can simulate cabling devices on a rack. Refresh key skills such as device placement (Rack & Stack), on-device power switching, device port-to-port cabling (including cable selection and management), troubleshooting, and more.
8. The Network Controller allows you a centralized dashboard to see the network's state, instantly discover and diagnose issues, and push configuration changes to all managed devices at once, whether you use its Web GUI or its APIs. You may also use real-world programs on your computer to access the Network Controller and run your infrastructure automation scripts.

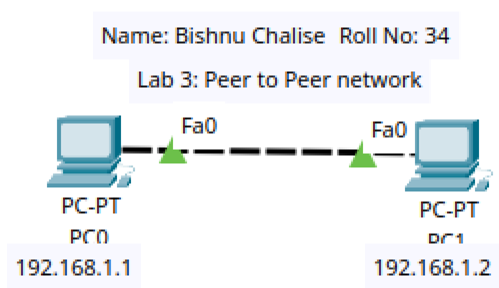
Peer to Peer Network

Objective: To understand the Peer-to-Peer network

Background: A peer-to-peer (P2P) network is a decentralized network architecture where each participant, or "peer," has equal status and can directly share resources, such as files, without relying on a central server. This model contrasts with traditional client-server networks, where clients depend on centralized servers for resources and services.

Key Characteristics of Peer-to-Peer Networks:

1. **Decentralization**: Each peer can act as both a client and a server, facilitating direct communication and resource sharing.
2. **Resource Sharing**: Peers can share various resources, including files, bandwidth, and processing power, making the network highly efficient.
3. **Scalability**: P2P networks can easily scale as more peers join, distributing the load across the network without the need for additional servers.
4. **Robustness**: Since there is no single point of failure, P2P networks can be more resilient to outages or attacks.
5. **Dynamic Participation**: Peers can join or leave the network at any time, allowing for flexible and adaptive network



2. Copper cross-over wire has been selected from connections, and the wire has been connected with PC0 and PC1.
3. IP address has been set in PC0 and PC 1.

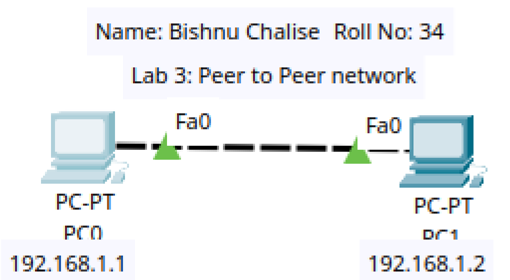
   PC0:

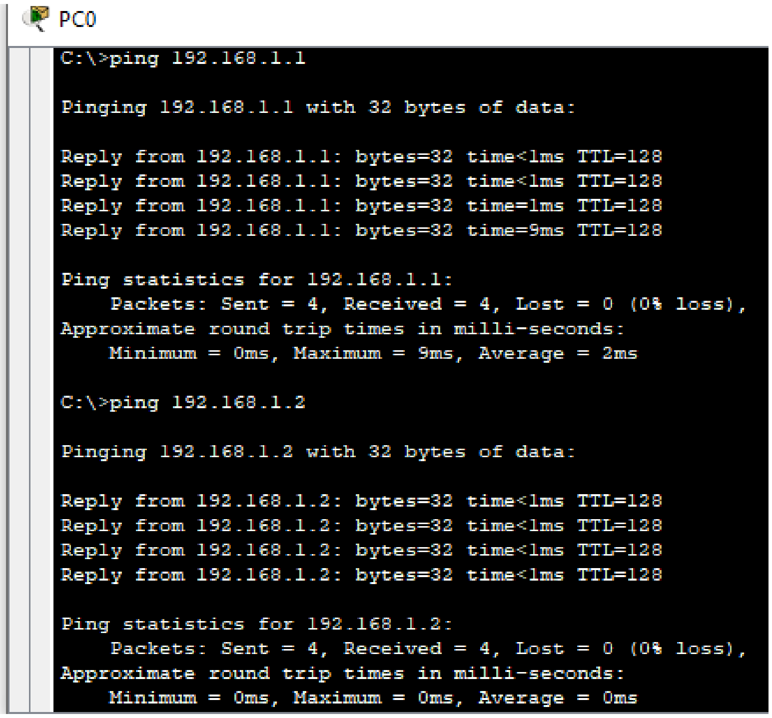
   3.1. To set IP address we clicked on PC0.
   3.2. Then, desktop option has been selected.
   3.3. IP configuration option has been selected.
   3.4. IP address has been entered in IPV4 Address and Subnet Mask.

   PC1:

   3.5. To set IP address we clicked on PC1.
   3.6. Then, desktop option has been selected.
   3.7. IP configuration option has been selected.
   3.8. IP address has been entered in IPV4 Address and Subnet Mask.

Validation:






```
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time=1ms TTL=128
Reply from 192.168.1.1: bytes=32 time=9ms TTL=128

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 9ms, Average = 2ms

C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Conclusion: In this experiment, we successfully implemented a peer-to-peer (P2P) network, demonstrating its fundamental principles.

LAN

Background: A Local Area Network (LAN) is a network that connects computers and devices within a limited geographic area, such as a home, office, or campus. LANs are characterized by high data transfer rates, low latency, and the ability to share resources such as printers, files, and internet connections among connected devices.

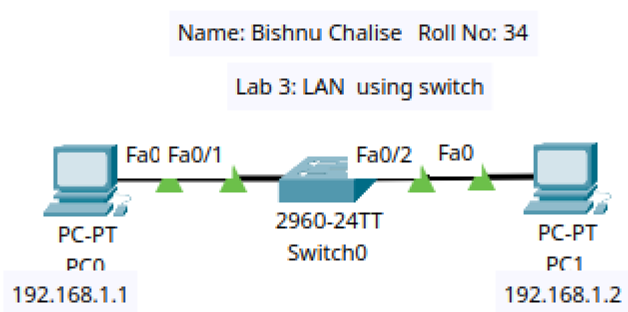
The primary purpose of a LAN is to facilitate communication and resource sharing among devices, making it an essential component in both personal and organizational settings. LANs can be implemented using various technologies, including Ethernet cables, Wi-Fi, and fiber optics, and can be configured in different topologies, such as star, bus, or ring.

LAN using switch

Objective: To understand the LAN using switch

Background: A switch is a critical networking device that connects multiple devices within a Local Area Network (LAN). Unlike a hub, which broadcasts data to all connected devices, a switch intelligently directs data only to the intended recipient by using MAC addresses. This not only improves network efficiency by reducing collisions but also enhances security, as data is not indiscriminately sent to all devices.

Topology:



Procedure:

1. Two PCs and a switch has been selected from the end devices and Network devices respectively.
2. Copper straight through wire has been selected from connections, then PC0 and switch and switch and PC1 has been connected respectively.
3. IP address has been set in PC0 and PC 1.

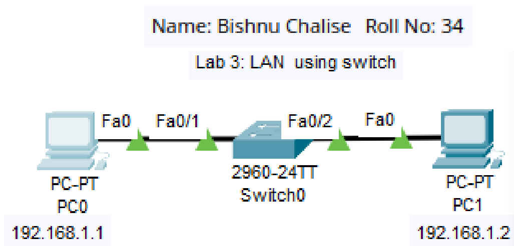
PC0:

3.1.1.To set IP address we clicked on PC0.
3.1.2.Then, desktop option has been selected.
3.1.3.IP configuration option has been selected.
3.1.4.IP address has been entered in IPV4 Address and Subnet Mask.

PC1:

3.1.5.To set IP address we clicked on PC1.
3.1.6.Then, desktop option has been selected.
3.1.7.IP configuration option has been selected.
3.1.8.IP address has been entered in IPV4 Address and Subnet Mask.

Validation:



PC0

```
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time<1ms TTL=128
Reply from 192.168.1.1: bytes=32 time=21ms TTL=128
Reply from 192.168.1.1: bytes=32 time=21ms TTL=128

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 21ms, Average = 10ms

C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=36ms TTL=128
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 36ms, Average = 9ms
```

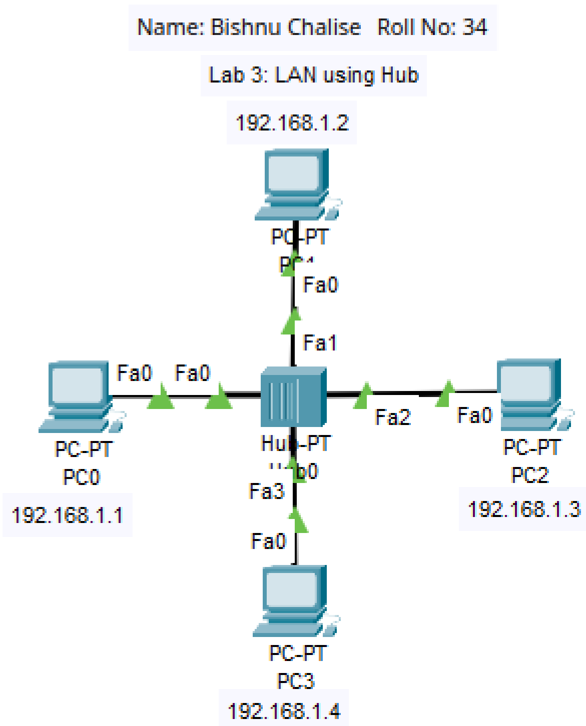
Conclusion: In this experiment, we successfully implemented a Local Area Network (LAN) using a network switch, demonstrating the fundamental principles of networking and the advantages of using a switch for data transmission.

LAN using hub

Objective: To understand the LAN using hub

Background: A hub is a basic networking device used to connect multiple Ethernet devices, forming a Local Area Network (LAN). Operating at the physical layer of the OSI model, a hub facilitates communication by receiving data packets from one device and broadcasting them to all other connected devices.

Topology:



Procedure:

1. Four PCs and a hub has been selected from the end devices and Network devices respectively.
2. Copper straight through wire has been selected from connections, then PC0, PC1, PC2 and PC3 are connected to hub respectively.
3. IP address has been set in PC0, PC1, PC2 and PC3.

PC0:

3.1.1. To set IP address we clicked on PC0.
3.1.2. Then, desktop option has been selected.
3.1.3. IP configuration option has been selected.
3.1.4. IP address has been entered in IPV4 Address and Subnet Mask.

PC1:

3.1.5. To set IP address we clicked on PC1.
3.1.6. Then, desktop option has been selected.
3.1.7. IP configuration option has been selected.
3.1.8. IP address has been entered in IPV4 Address and Subnet Mask.

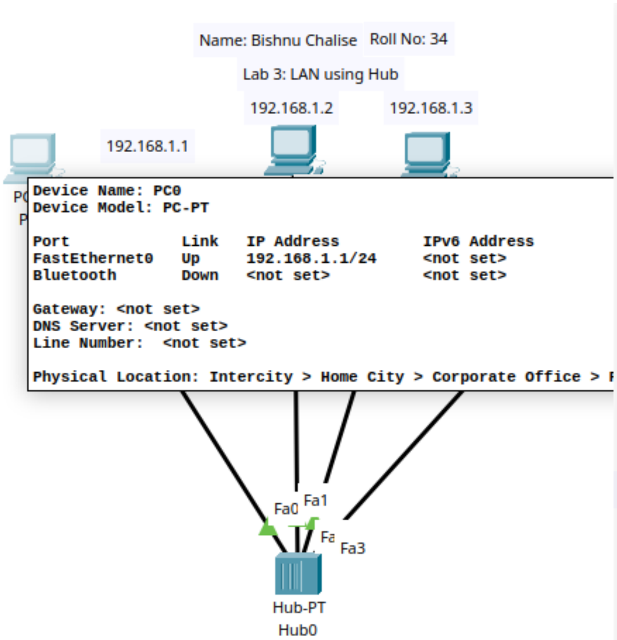
PC2:

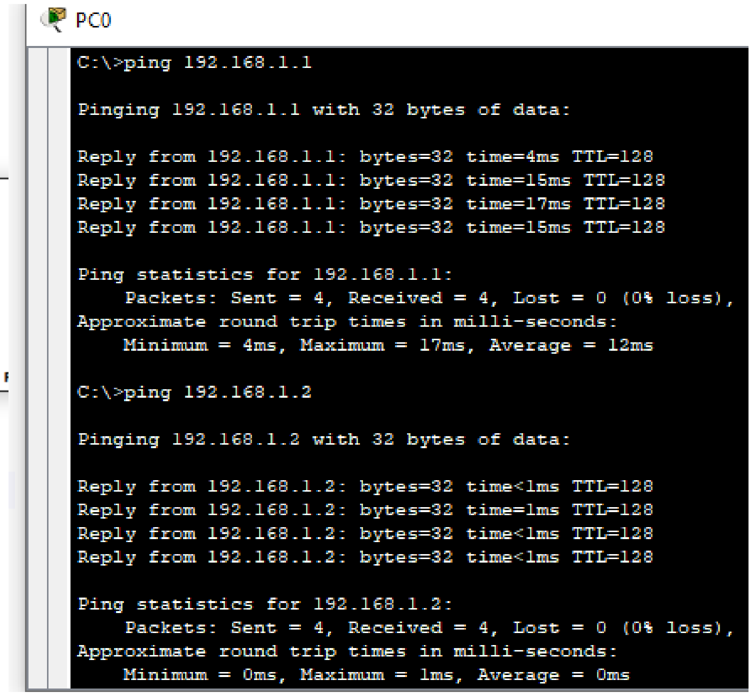
3.1.9. To set IP address we clicked on PC2.
3.1.10. Then, desktop option has been selected.
3.1.11. IP configuration option has been selected.
3.1.12. IP address has been entered in IPV4 Address and Subnet Mask.

PC3:

3.1.13. To set IP address we clicked on PC3.
3.1.14. Then, desktop option has been selected.
3.1.15. IP configuration option has been selected.

3.1.16. IP address has been entered in IPV4 Address and Subnet Mask.





Conclusion: In this experiment, we successfully implemented a Local Area Network (LAN) using a hub, demonstrating the fundamental principles of network connectivity and data transmission.

Lab 4: HTTP, DNS and DHCP

Objective: To configure and understand the HTTP, DNS and DHCP service.

Background:

DHCP Server: A **DHCP server** (Dynamic Host Configuration Protocol server) is a network server that automatically assigns IP addresses and other network configuration parameters (like subnet masks, gateways, and DNS servers) to devices (clients) on a network. It ensures that devices can communicate with each other on an IP network without the need for manual configuration.

DNS Server: A **DNS server** (Domain Name System server) is a system that translates domain names, such as `www.lokesh.com`, into IP addresses like `192.168.2.4`, which computers use to locate and communicate with each other over a network, such as the internet. DNS servers are essential for making the internet user-friendly, as they allow us to use human-readable domain names instead of numerical IP addresses.

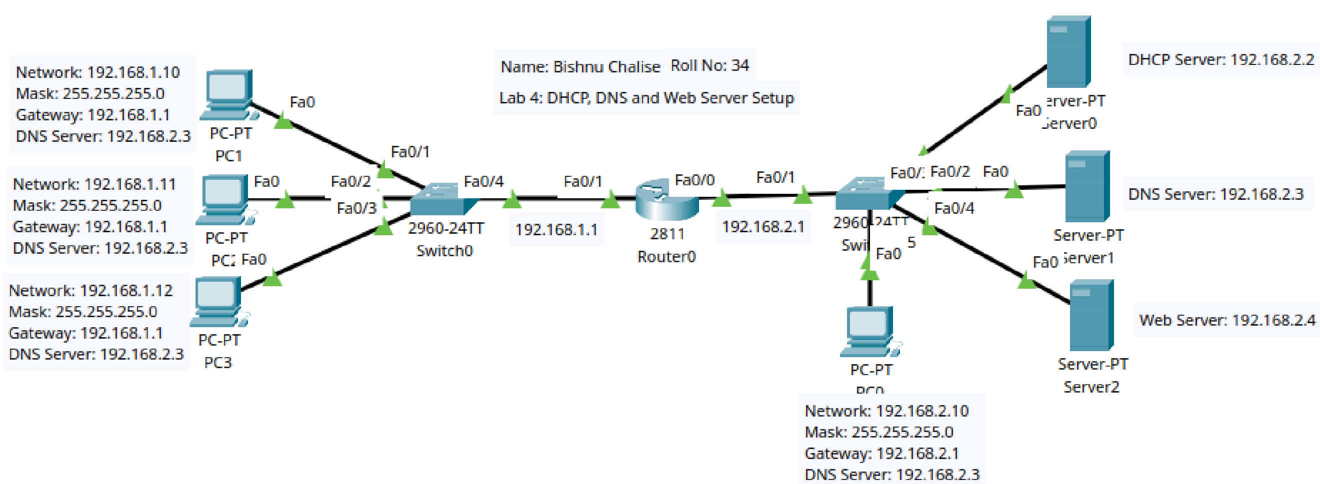
Web Server: A **web server** is a computer system or software that delivers web content (like websites or web applications) to users over the internet. It receives requests from clients (typically web browsers) and responds by serving web pages, files, or other data.

IP Addressing

Device	Interface	IP	Subnet Mask	DNS Server	Default gateway
Router 0	Fa0/4	192.168.1.1	255.255.255.0	-	-
Router 0	Fa0/0	192.168.2.1	255.255.255.0	-	-
Server 0	NIC	192.168.2.2	255.255.255.0	192.168.2.3	192.168.2.1
Server 1	NIC	192.168.2.3	255.255.255.0	192.168.2.3	192.168.2.1
Server 2	NIC	192.168.2.4	255.255.255.0	192.168.2.3	192.168.2.1
PC0	NIC	192.168.2.10	255.255.255.0	192.168.2.3	192.168.2.1
PC1	NIC	192.168.1.11	255.255.255.0	192.168.2.3	192.168.1.1
PC2	NIC	192.168.1.12	255.255.255.0	192.168.2.3	192.168.1.1
PC3	NIC	192.168.1.13	255.255.255.0	192.168.2.3	192.168.1.1

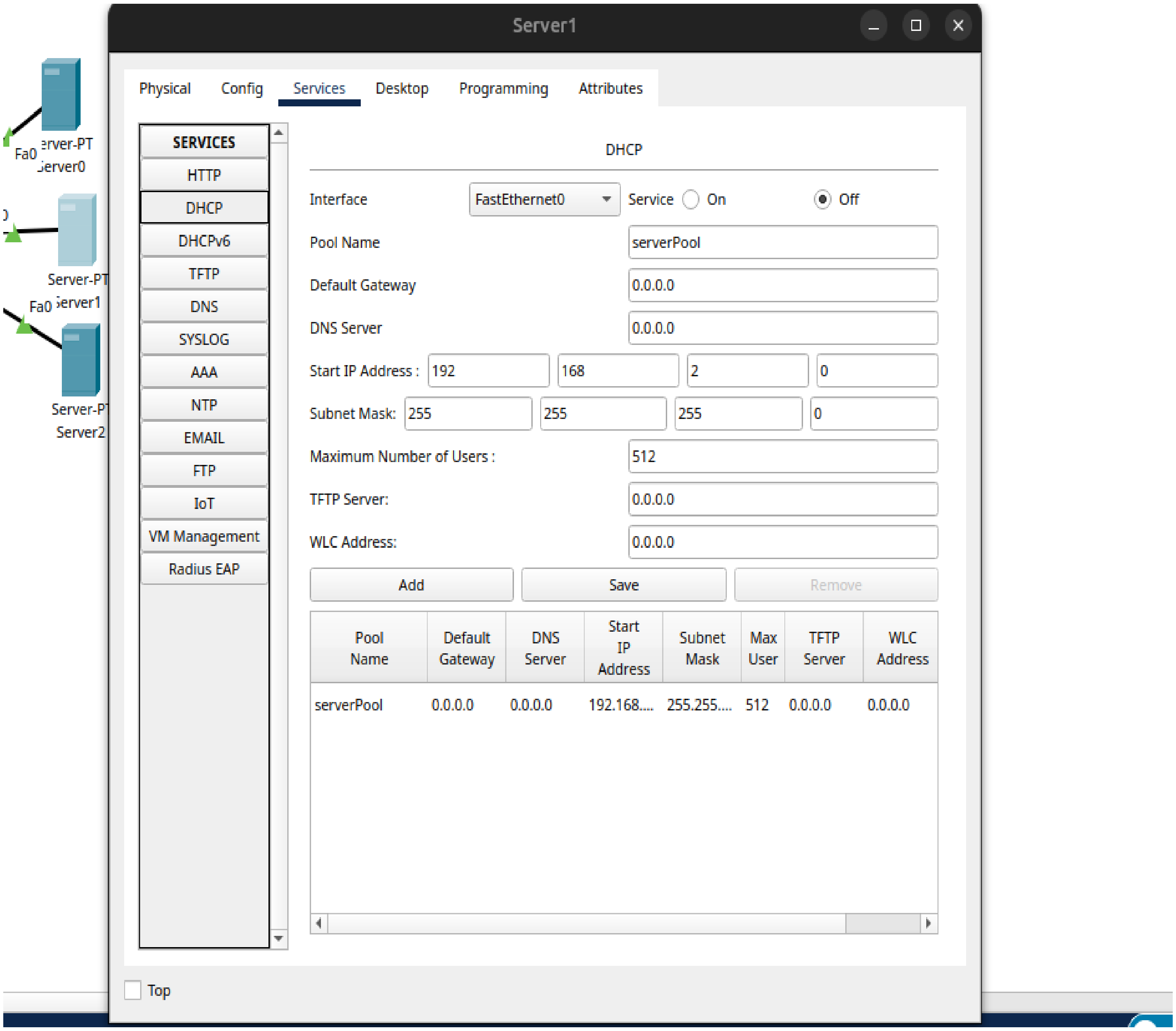
IP address to PCs has been assigned through DHCP server dynamically.

Topology

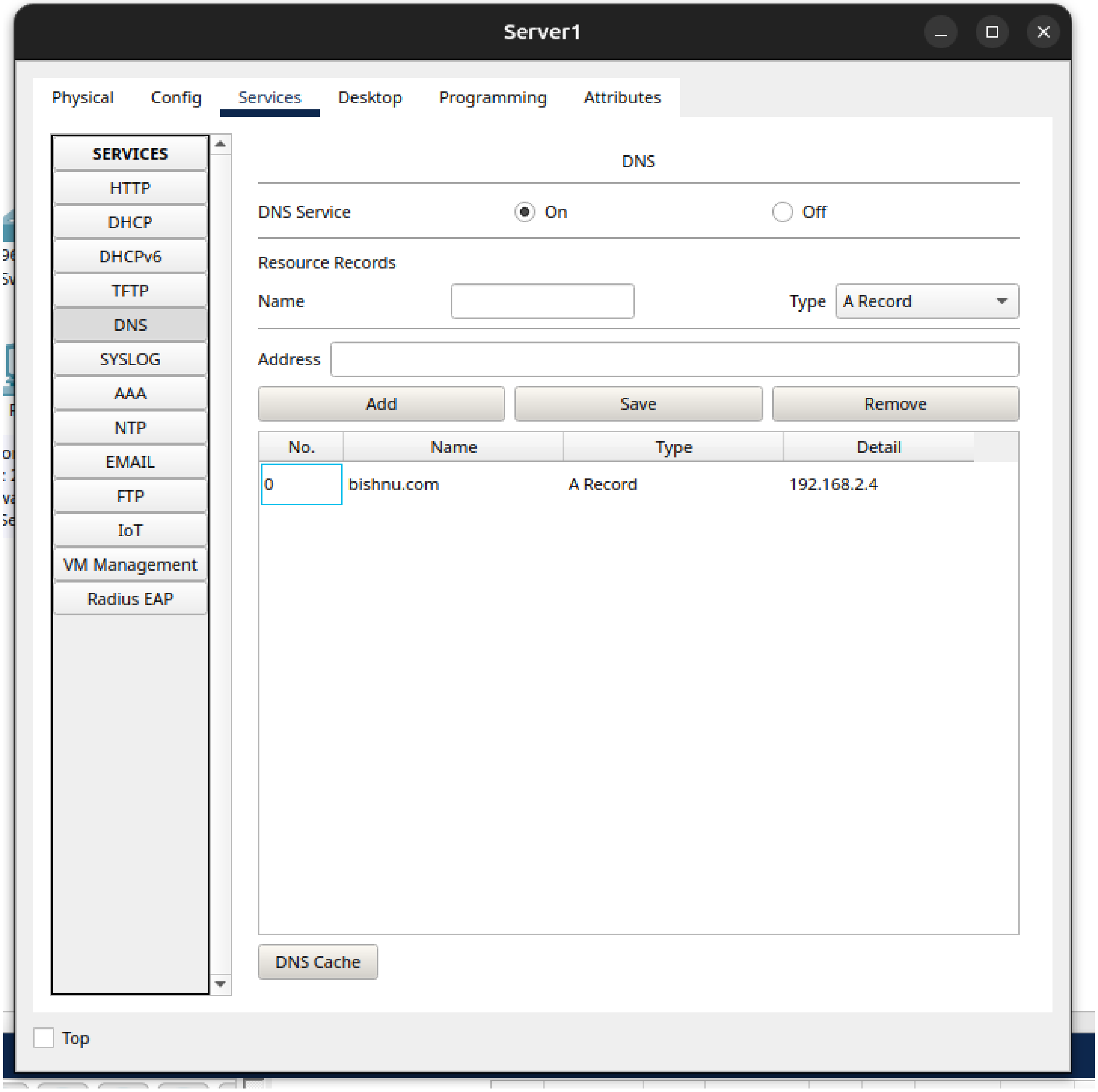


Procedure

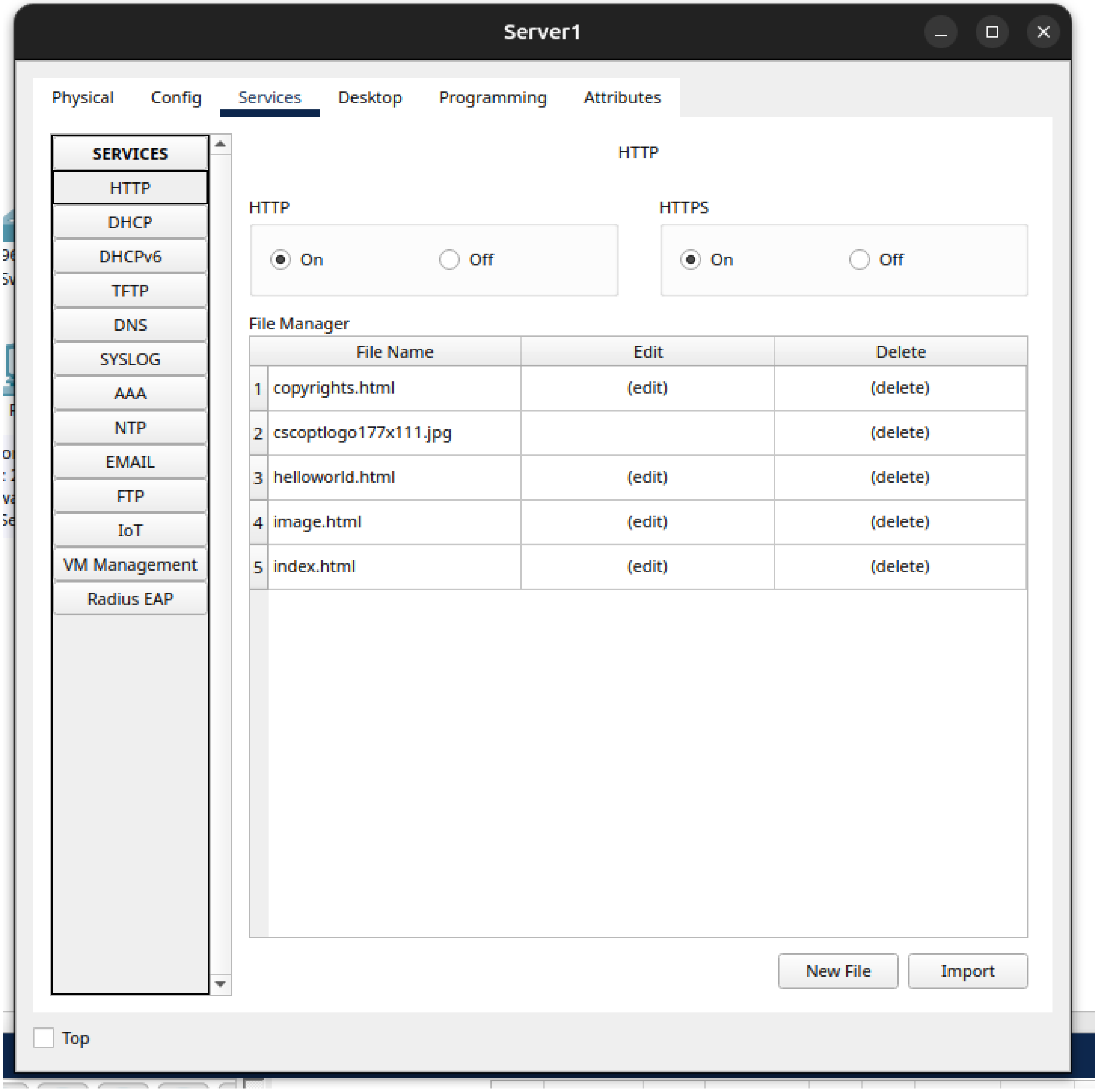
1. The topology has been created as shown above.
2. The IP address to each server has been assigned as shown in IP table.
3. The DHCP Server has been set as shown below



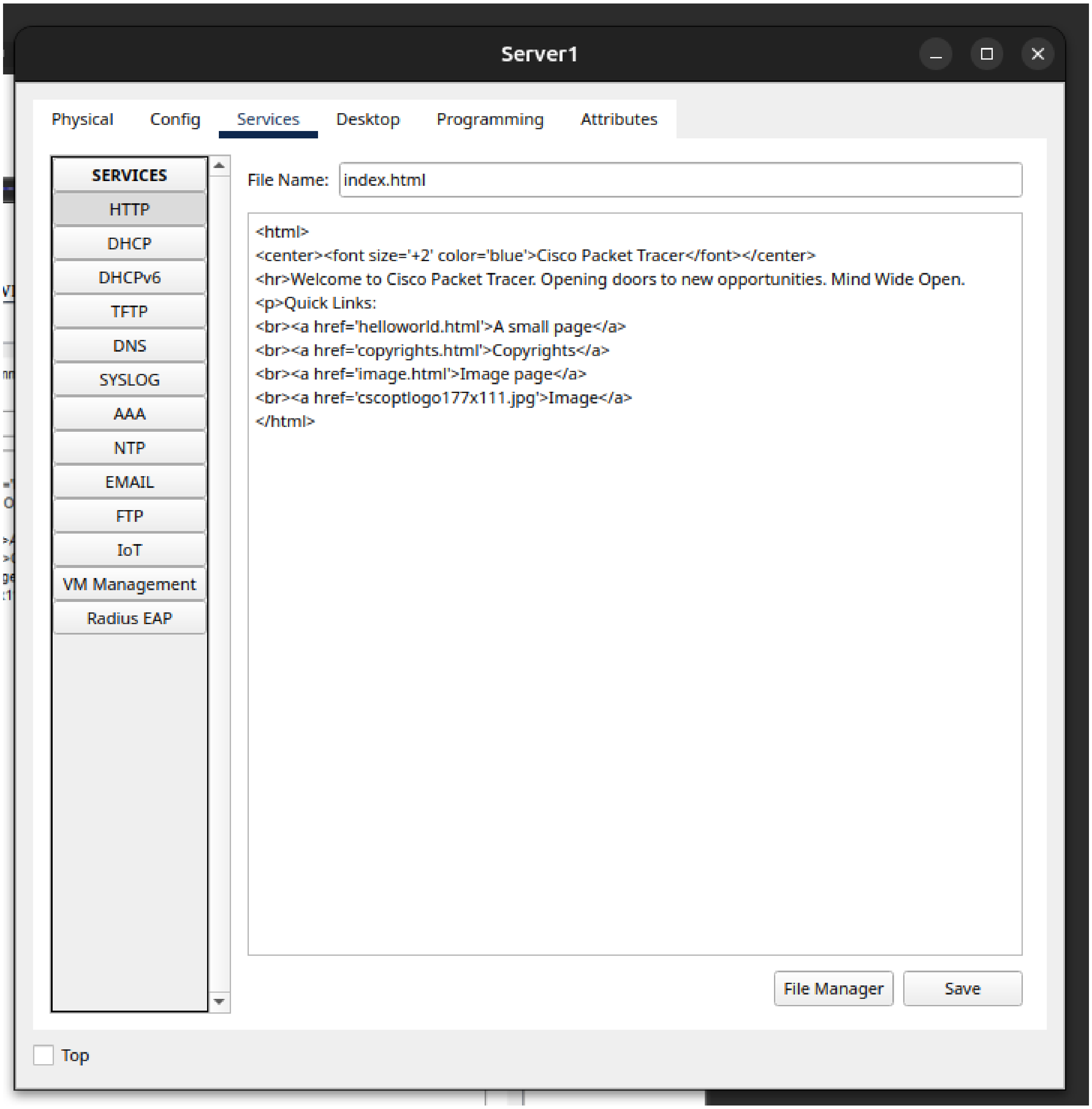
4. The DNS server has been set as shown below



5. The Web server has been set as shown below



6. The index.html file has been edited as shown below.



7. The router interface has been setup as:

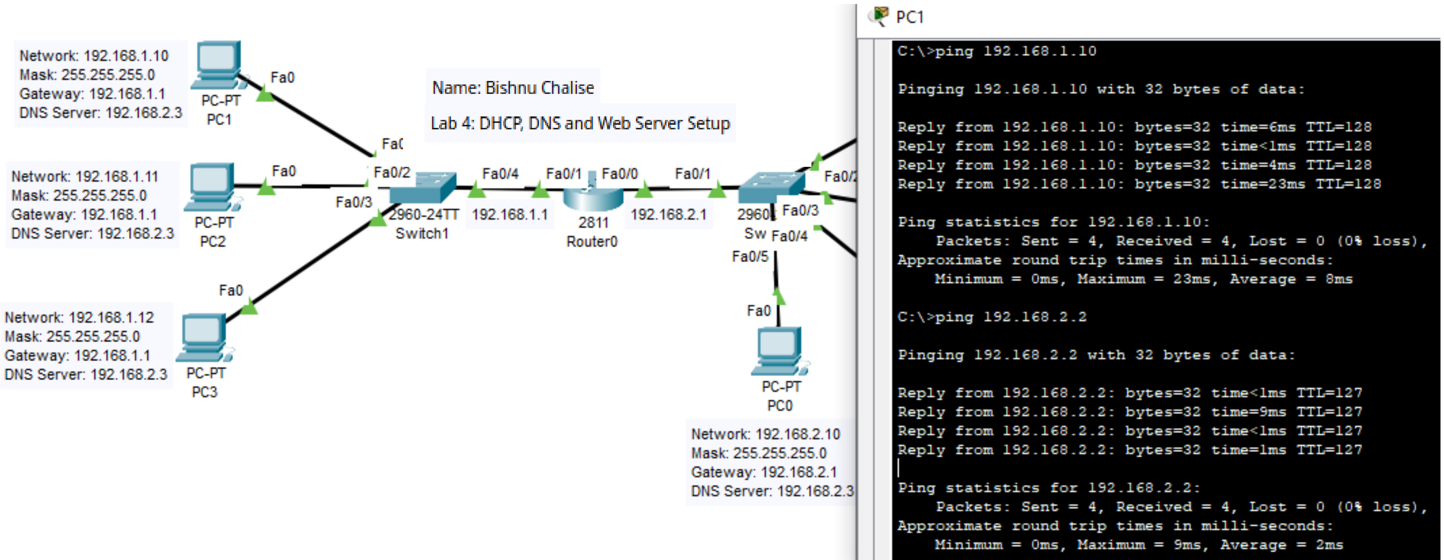
```
Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#ip add 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#int fa0/1
Router(config-if)#ip add 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
```

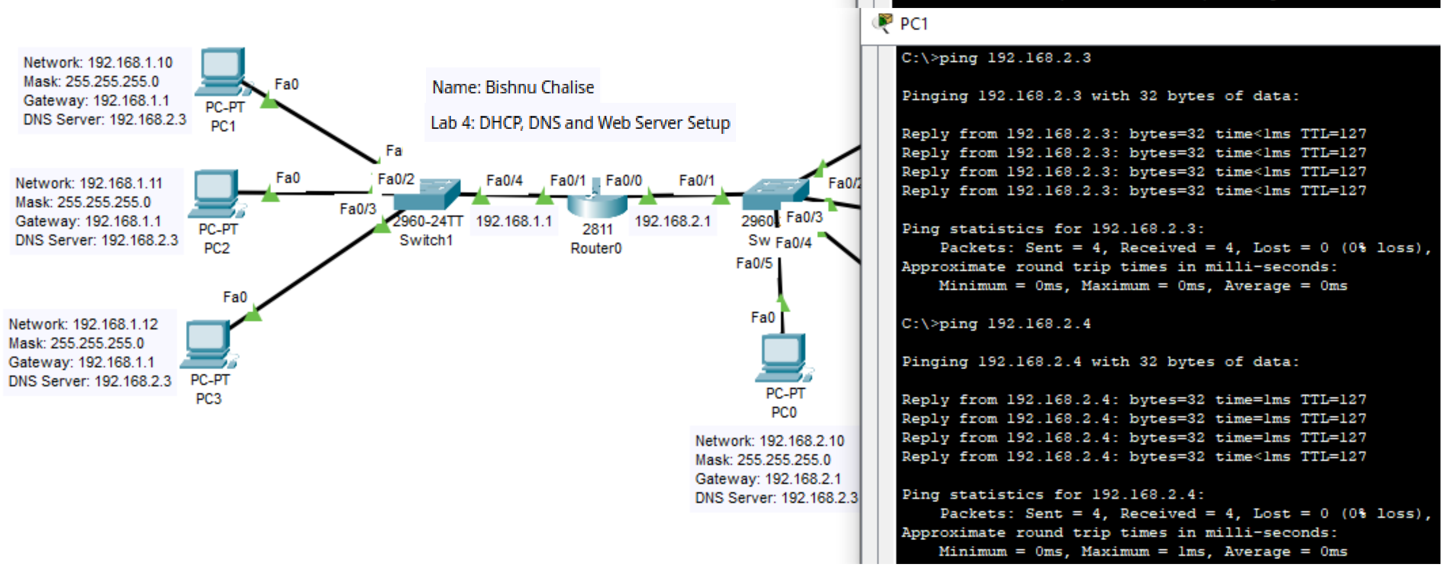
To forward broadcast message to DHCP server:

```
Router(config-if)#int fa0/1
Router(config-if)#ip helper-address 192.168.2.2
```

8. The IP in each PC has been provided by selecting DHCP option as show in IP table.

Validation:





Conclusion: In this way we can setup DHCP, DNS and Web server in our network and these services works in close association as shown in this experiment.

Lab 5: Basic Router Configuration with DHCP

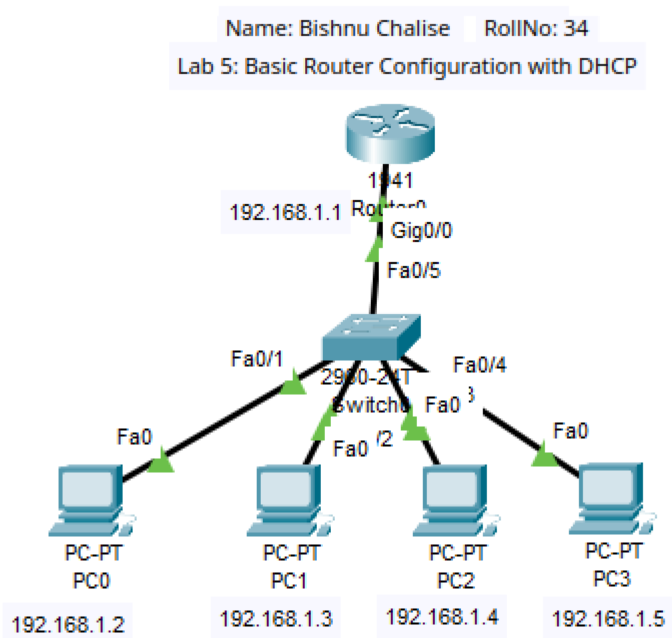
Objectives: Configure Basic Router with DHCP using packet tracer.

Background: Dynamic Host Configuration Protocol (DHCP) is a network management protocol used to automate the process of configuring devices on IP networks. DHCP allows servers to dynamically assign IP addresses and other network configuration parameters to clients, enabling them to communicate on the network.

IP Addressing Plan:

Device	Interface	IP Address	Subnet Mask	Gateway
PC0	NIC	192.168.1.2	255.255.255.0	192.168.1.1
PC1	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC2	NIC	192.168.1.4	255.255.255.0	192.168.1.1
PC3	NIC	192.168.1.5	255.255.255.0	192.168.1.1
Router0	GigabitEthernet0/0.100	192.168.1.1	255.255.255.0	-

Topology:



Procedure:

1. The topology has been created as shown above.
2. The following configurations has been performed in the router:
   2.1. The password has been set in the router.

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname ascol
```

```
ascol(config)#line console 0
ascol(config-line)#password 12345
ascol(config-line)#login
ascol(config-line)#exit
ascol(config)#exit
ascol#
%SYS-5-CONFIG_I: Configured from console by console

ascol#exit

User Access Verification

Password:

ascol>enable
ascol#show running-config
Building configuration...

Current configuration : 633 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname ascol
!
!
!
!
!
!
!
!
no ip cef
no ipv6 cef
```

```
!
!
!
!
license udi pid CISCO1941/K9 sn FTX1524WF11-
!
!
!
!
!
!
!
!
!
!
!
spanning-tree mode pvst
!
!
!
!
!
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!

interface Vlan1
 no ip address
 shutdown
!
ip classless
!
ip flow-export version 9
!
!
!
!
!
!
!
line con 0
 password 12345
 login
!
line aux 0
!
line vty 0 4
 login
!
!
!
end
```

```
ascol#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ascol(config)#enable secret 56789
ascol(config)#exit
ascol#
%SYS-5-CONFIG_I: Configured from console by console

ascol#exit

User Access Verification

Password:

ascol>enable
Password:
ascol#show running-config
Building configuration...

Current configuration : 682 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname ascol
!
!
!
enable secret 5 $1$mERr$HjE7DYKTpouXLdIC8YnwK0
!
!
!
!
!
!
no ip cef
no ipv6 cef
!
```

```
!
!
!
license udi pid CISCO1941/K9 sn FTX1524WF11-
!
!
!
!
!
!
!
!
!
!
!
spanning-tree mode pvst
!
!
!
!
!
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!

ip classless
!
ip flow-export version 9
!
!
!
!
!
!
!
line con 0
 password 12345
 login
!
line aux 0
!
line vty 0 4
 login
!
!
!
end
```

2.2. The IP address to each PC has been assigned.

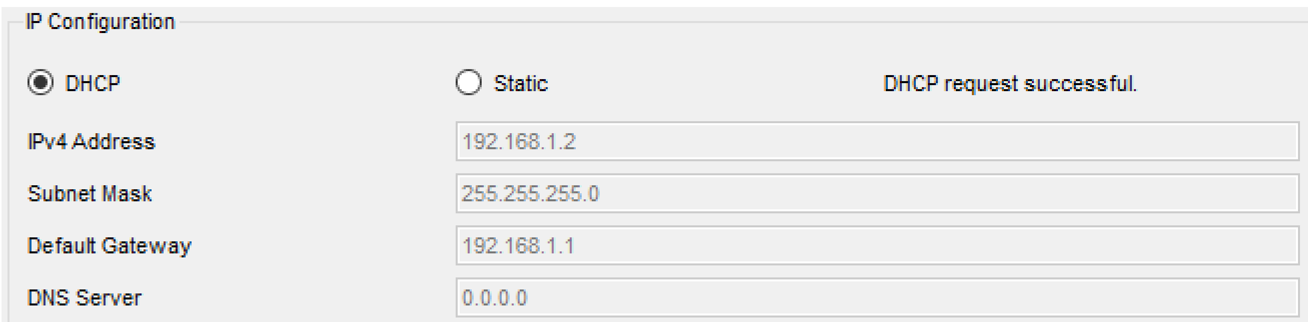
```
ascol#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ascol(config)#int gig0/0
ascol(config-if)#ip address 192.168.1.1 255.255.255.0
ascol(config-if)#no shut

ascol(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

ascol(config-if)#exit
```

2.3. DHCP has been set.

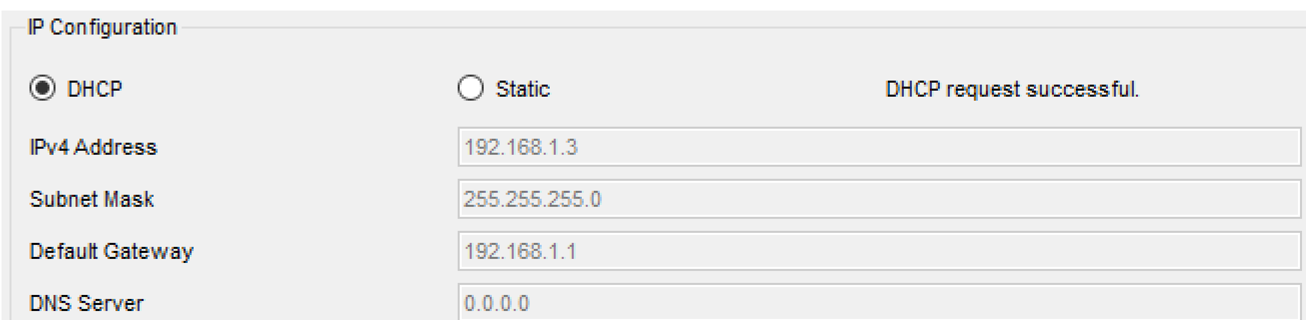
```
ascol(config)#ip dhcp pool lan1
ascol(dhcp-config)#network 192.168.1.0 255.255.255.0
ascol(dhcp-config)#default-router 192.168.1.1
ascol(dhcp-config)#exit
```

3. In each PC the DCHP option has been enabled.

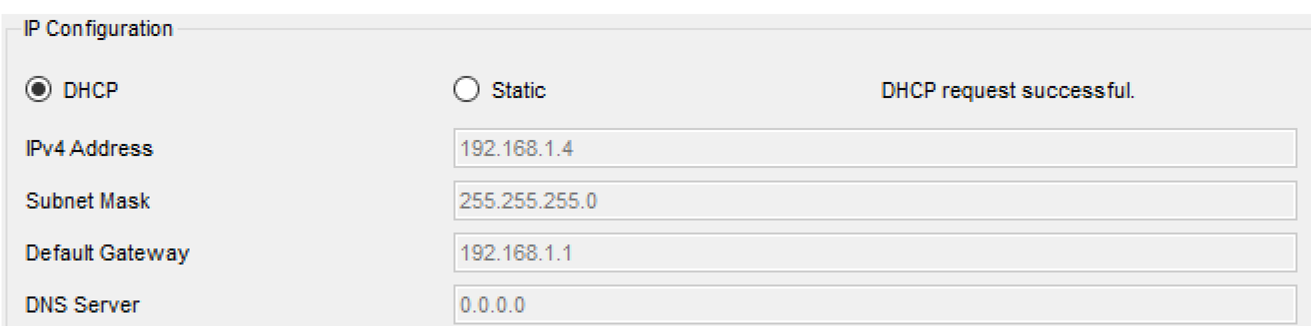
PC0:



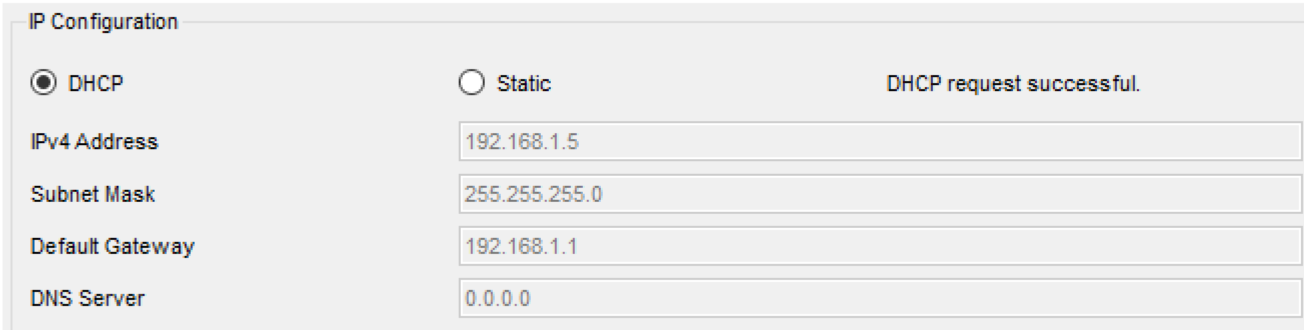
PC1:



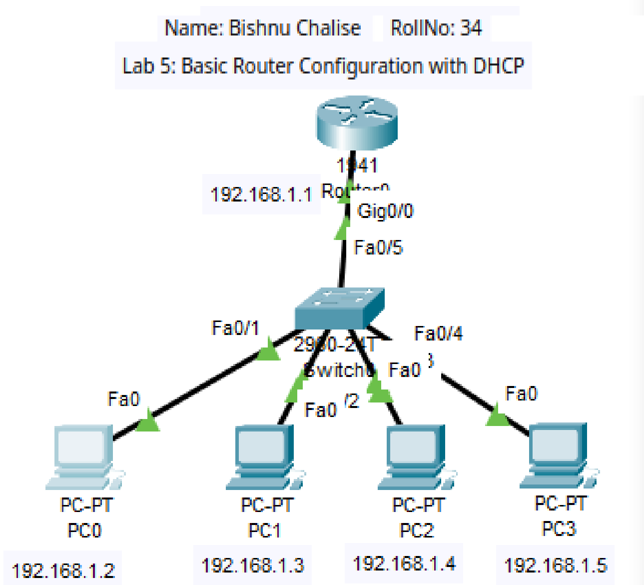
PC2:

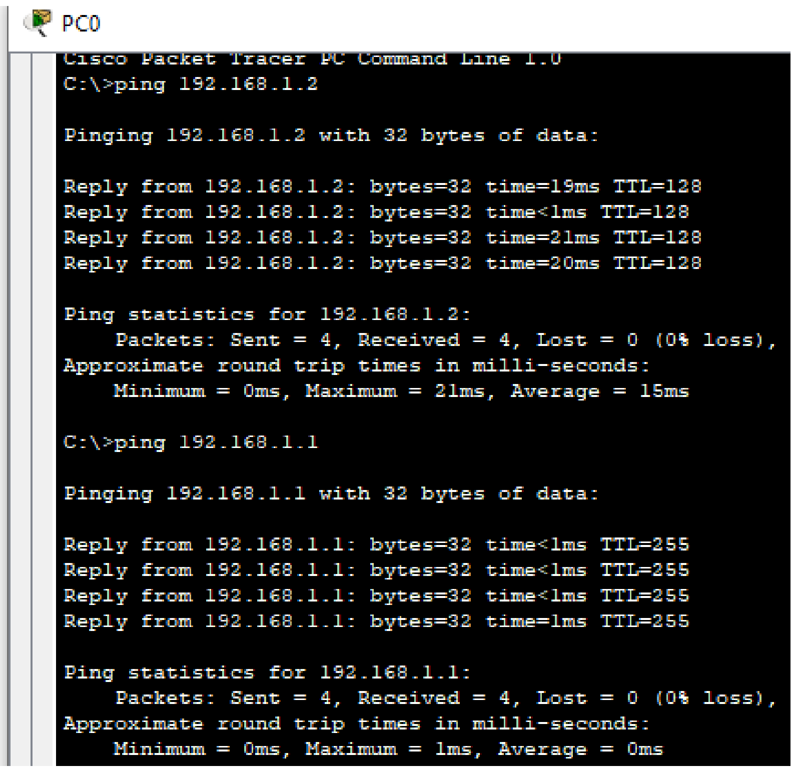


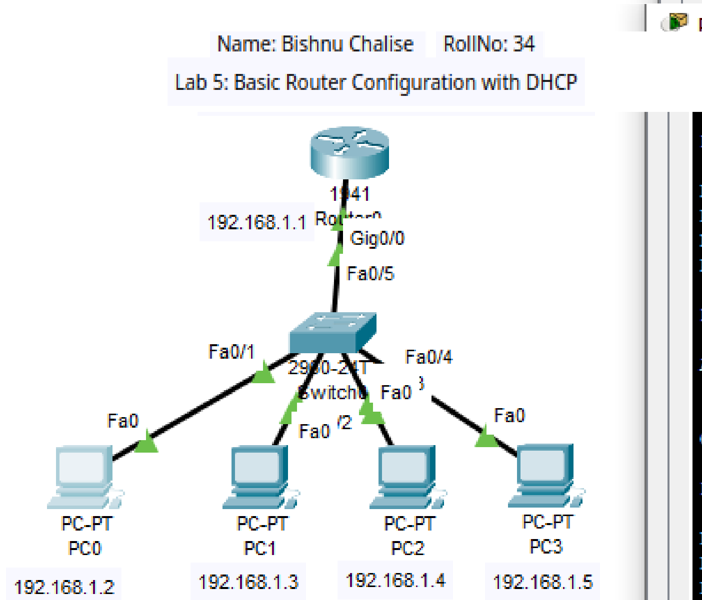
PC3:

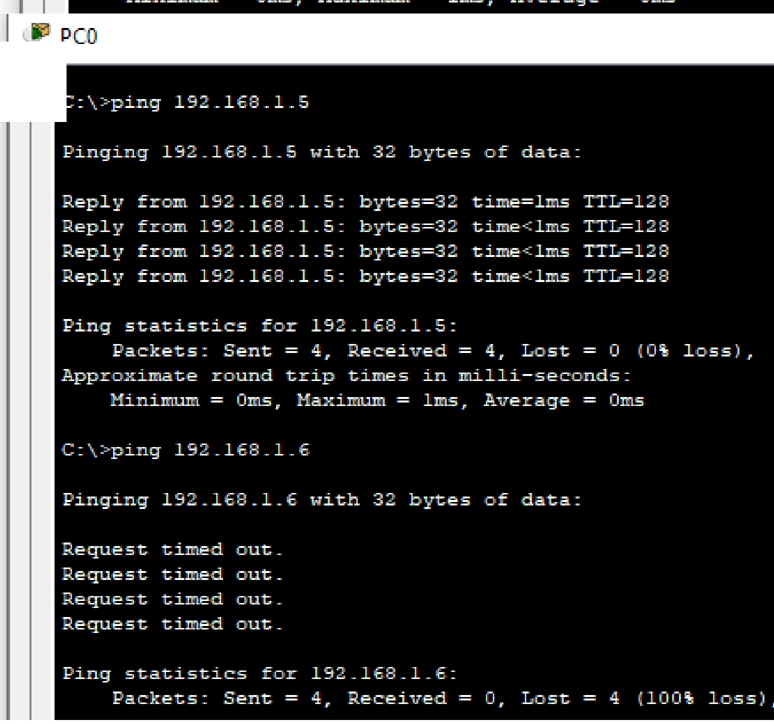

IP Configuration
DHCP
Static
DHCP request successful.
IPv4 Address
192.168.1.5
Subnet Mask
255.255.255.0
Default Gateway
192.168.1.1
DNS Server
0.0.0.0

Validation:


Name: Bishnu Chalise
RollNo: 34
Lab 5: Basic Router Configuration with DHCP
192.168.1.1
Gig0/0
Fa0/5
Fa0/1
Fa0/4
Fa0
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5


PC0
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=19ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=21ms TTL=128
Reply from 192.168.1.2: bytes=32 time=20ms TTL=128
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 21ms, Average = 15ms
C:\>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms


Name: Bishnu Chalise
RollNo: 34
Lab 5: Basic Router Configuration with DHCP
192.168.1.1
Gig0/0
Fa0/5
Fa0/1
Fa0/4
Fa0
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5


PC0
C:\>ping 192.168.1.5
Pinging 192.168.1.5 with 32 bytes of data:
Reply from 192.168.1.5: bytes=32 time=1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>ping 192.168.1.6
Pinging 192.168.1.6 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.6:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

Conclusion: In this experiment, we successfully configured a basic router with DHCP (Dynamic Host Configuration Protocol) to streamline IP address allocation within a network. The process involved accessing the router's interface, enabling DHCP, and defining the necessary address pool, subnet mask, and gateway settings.

Lab 6: Implementation of Static Routing

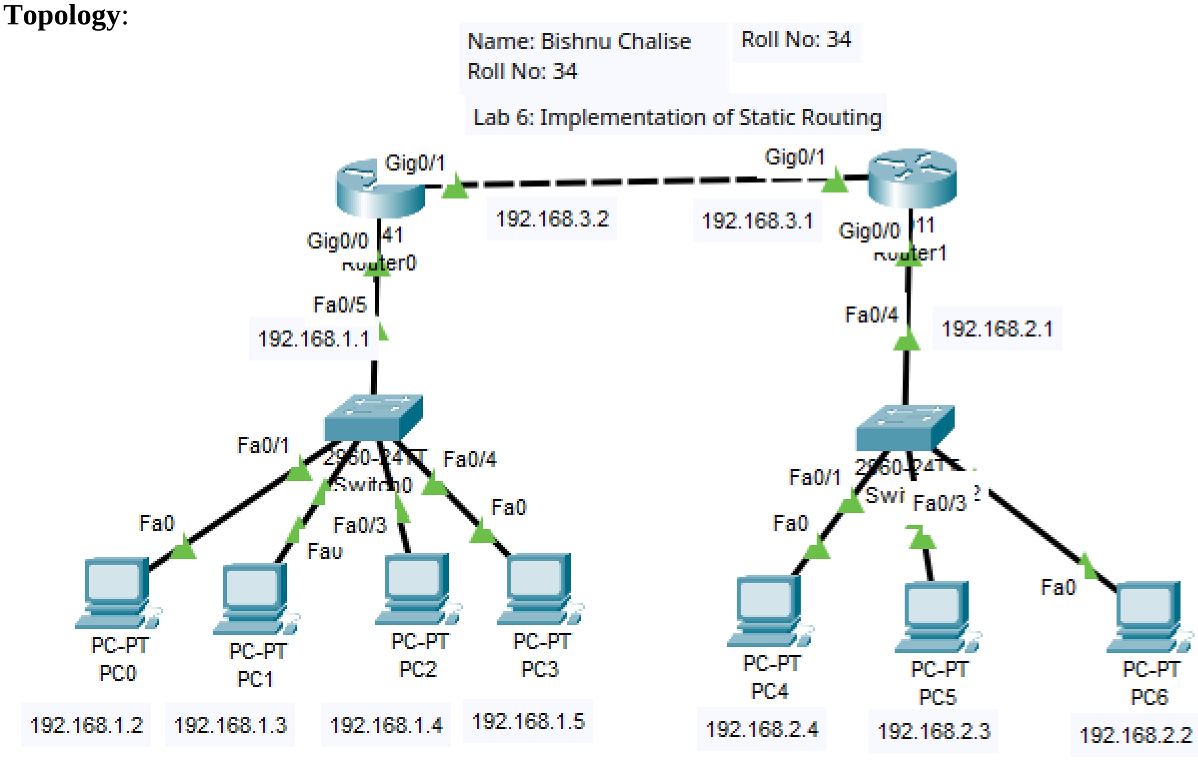
Objective: To configure and understand static routing.

Background: Static routing is a method of network routing where routes are manually configured by the network administrator. In static routing, fixed paths are set for data packets to follow through the network, as opposed to dynamic routing protocols, which automatically adjust routes based on network conditions. Static routes are predefined and remain constant unless manually changed, giving administrators full control over the routing decisions.

IP Addressing Plan:

Device	Interface	IP	Subnet Mask	VLAN	Default Gateway
PC0	NIC	192.168.1.2	255.255.255.0	Default	192.168.1.1
PC1	NIC	192.168.1.3	255.255.255.0	Default	192.168.1.1
PC2	NIC	192.168.1.4	255.255.255.0	Default	192.168.1.1
PC3	NIC	192.168.1.5	255.255.255.0	Default	192.168.1.1
PC4	NIC	192.168.2.4	255.255.255.0	Default	192.168.2.1
PC5	NIC	192.168.2.3	255.255.255.0	Default	192.168.2.1
PC6	NIC	192.168.2.2	255.255.255.0	Default	192.168.2.1
Router0	Gig0/0	192.168.1.1	255.255.255.0	Default	-
Router0	Gig0/1	192.168.3.2	255.255.255.0	Default	-
Router1	Gig0/0	192.168.2.1	255.255.255.0	Default	-
Router1	Gig0/1	192.168.3.1	255.255.255.0	Default	-

Topology:



Procedure:

1. IP and Default gateway has been set in each PC.

Router0:

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.1.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#ip dhcp pool lan1
Router(dhcp-config)#network 192.168.1.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.1.1
Router(dhcp-config)#exit
```

Router1:

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.2.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#ip dhcp pool lan2
Router(dhcp-config)#network 192.168.2.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.2.1
Router(dhcp-config)#exit
```

2. IP address in the routers interfaces has been set.

Router0:

```
Router(config)#int gig0/1
Router(config-if)#ip add 192.168.3.1 255.255.255.0
Router(config-if)#no shut
```

Router1:

```
Router(config)#int gig0/1
Router(config-if)#ip add 192.168.3.2 255.255.255.0
Router(config-if)#no shut
```

3. Static route has been set.

Router0:

```
Router(config)#ip route 192.168.2.0 255.255.255.0 192.168.3.2
```

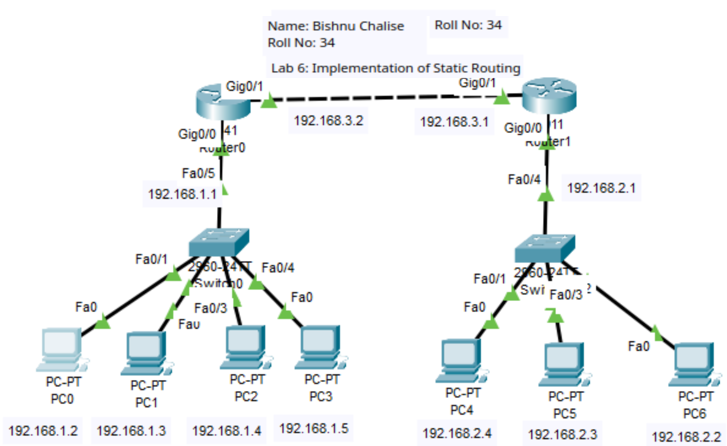
```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
S    192.168.2.0/24 [1/0] via 192.168.3.2
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/1
L       192.168.3.1/32 is directly connected, GigabitEthernet0/1
```

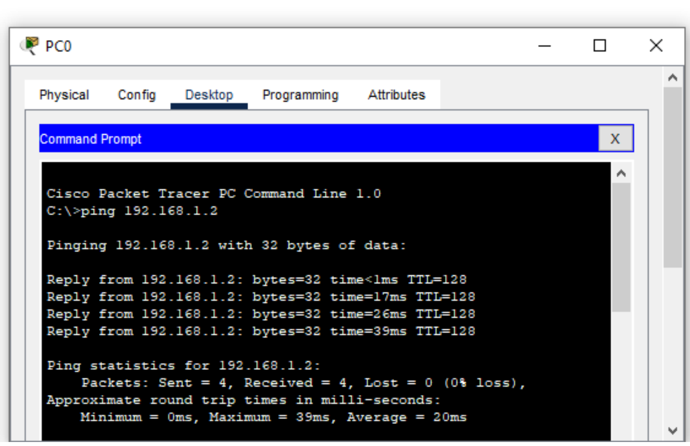
Router1:

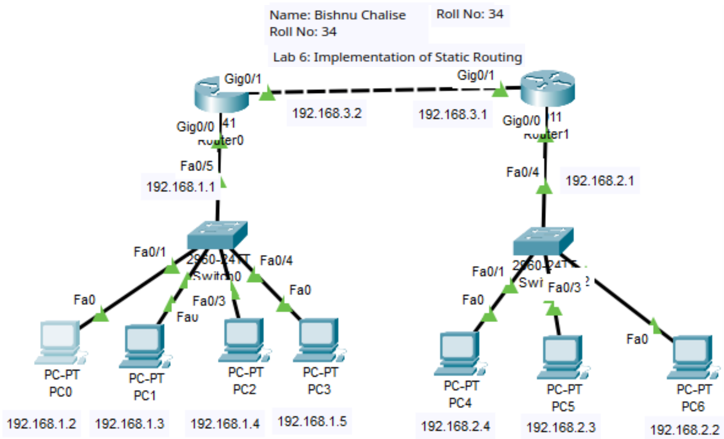
```
Router(config-if)#ip route 192.168.1.0 255.255.255.0 192.168.3.1
```

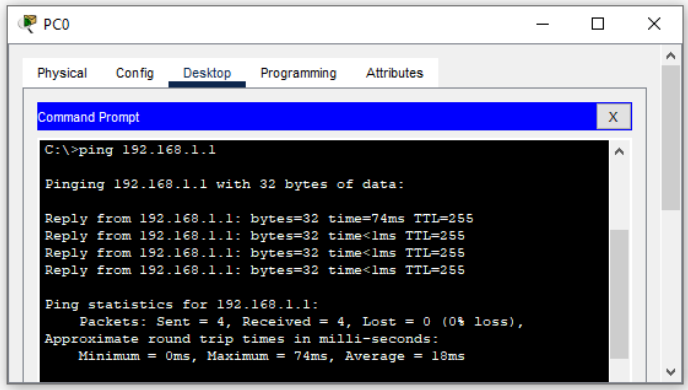
```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

S    192.168.1.0/24 [1/0] via 192.168.3.1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/0
L       192.168.2.1/32 is directly connected, GigabitEthernet0/0
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/1
L       192.168.3.2/32 is directly connected, GigabitEthernet0/1
```

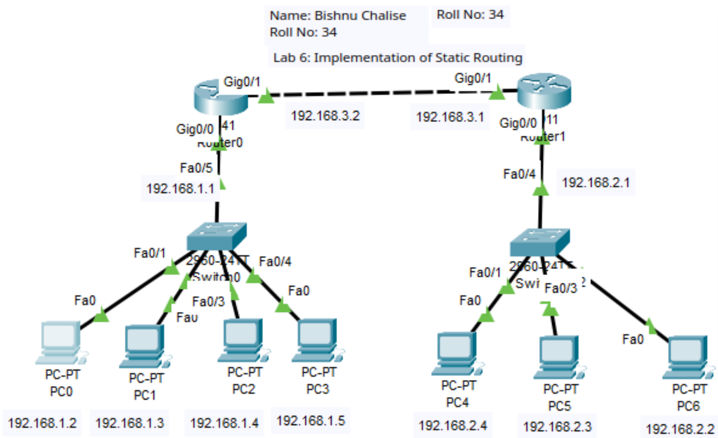
Verification:

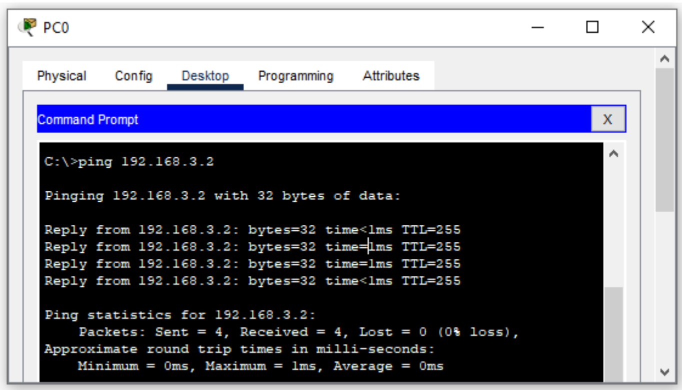


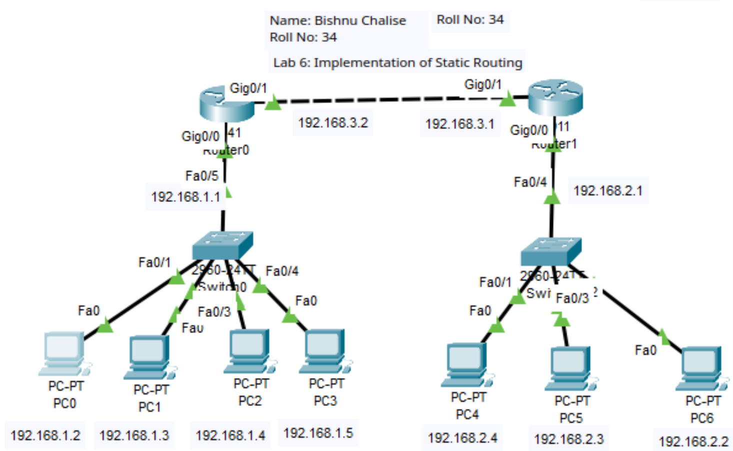


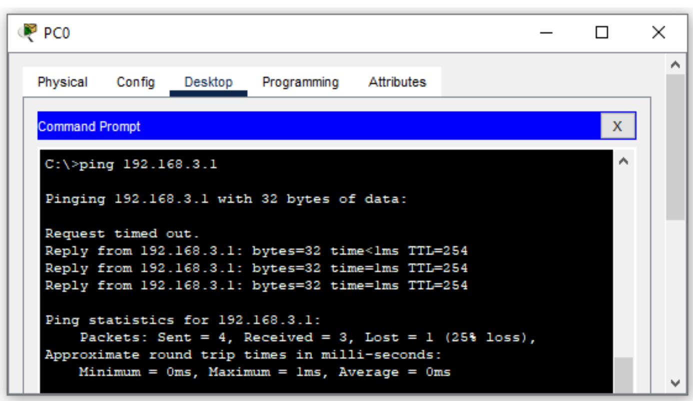

Name: Bishnu Chalise
Roll No: 34
Roll No: 34
Lab 6: Implementation of Static Routing
Gig0/1
Gig0/1
192.168.3.2
192.168.3.1
Gig0/0
Fa0/5
192.168.1.1
Fa0/4
192.168.2.1
Fa0/1
Fa0/4
Fa0/3
Fa0/1
Fa0/3
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC6
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5
192.168.2.4
192.168.2.3
192.168.2.2

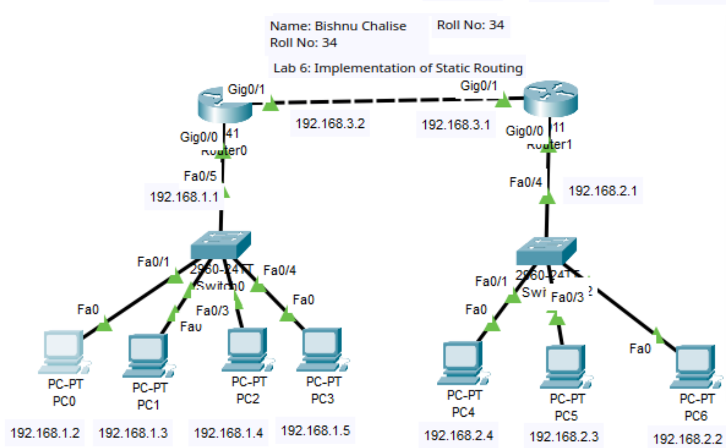

PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
C:\>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=74ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 74ms, Average = 18ms

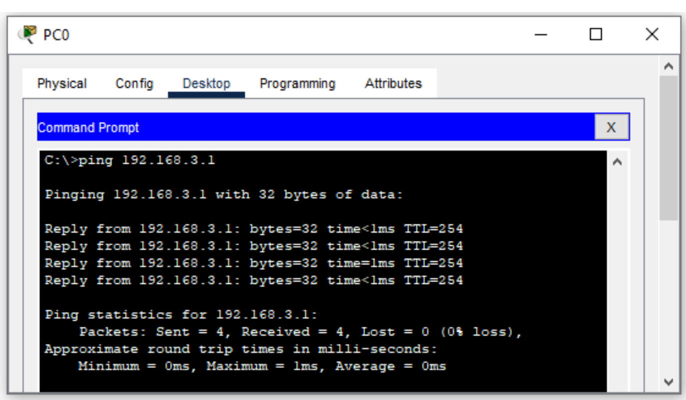

Name: Bishnu Chalise
Roll No: 34
Roll No: 34
Lab 6: Implementation of Static Routing
192.168.3.2
192.168.3.1
192.168.1.1
192.168.2.1
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5
192.168.2.4
192.168.2.3
192.168.2.2

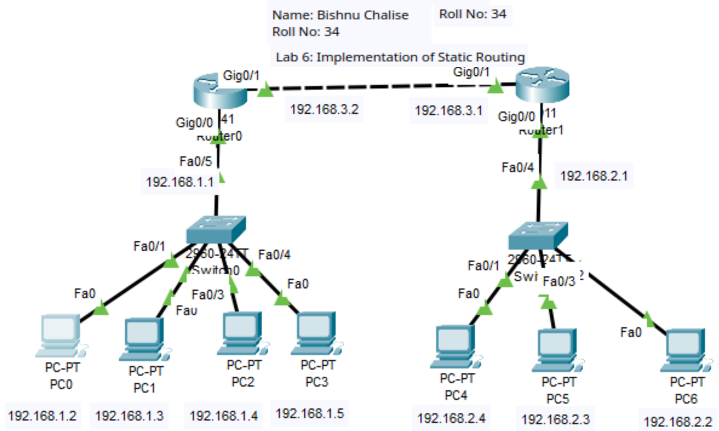

PC0
Command Prompt
C:\>ping 192.168.3.2
Pinging 192.168.3.2 with 32 bytes of data:
Reply from 192.168.3.2: bytes=32 time<1ms TTL=255
Reply from 192.168.3.2: bytes=32 time=1ms TTL=255
Reply from 192.168.3.2: bytes=32 time=1ms TTL=255
Reply from 192.168.3.2: bytes=32 time<1ms TTL=255
Ping statistics for 192.168.3.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

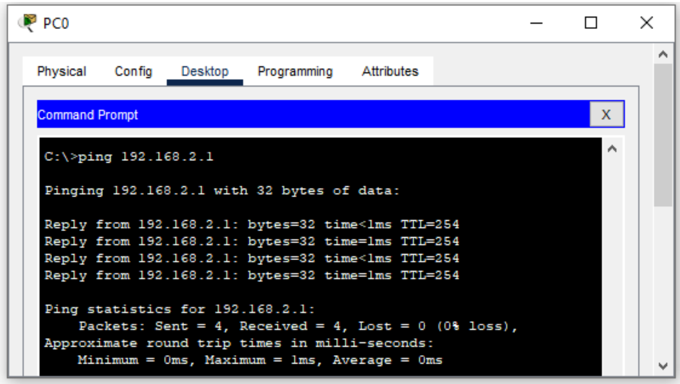

Name: Bishnu Chalise
Roll No: 34
Roll No: 34
Lab 6: Implementation of Static Routing
192.168.3.2
192.168.3.1
192.168.1.1
192.168.2.1
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5
192.168.2.4
192.168.2.3
192.168.2.2

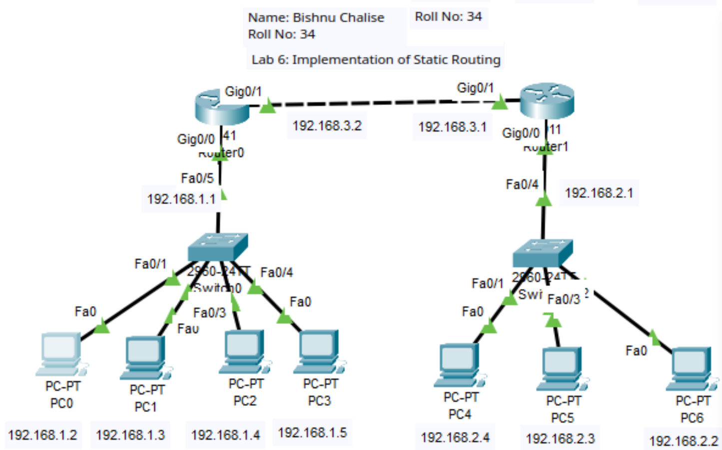

PC0
Command Prompt
C:\>ping 192.168.3.1
Pinging 192.168.3.1 with 32 bytes of data:
Request timed out.
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Reply from 192.168.3.1: bytes=32 time=1ms TTL=254
Reply from 192.168.3.1: bytes=32 time=1ms TTL=254
Ping statistics for 192.168.3.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

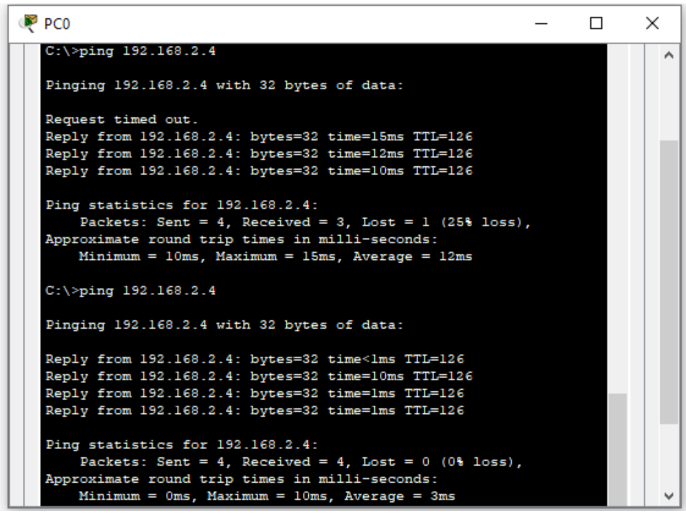

Name: Bishnu Chalise
Roll No: 34
Roll No: 34
Lab 6: Implementation of Static Routing
192.168.3.2
192.168.3.1
192.168.1.1
192.168.2.1
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5
192.168.2.4
192.168.2.3
192.168.2.2


PC0
Command Prompt
C:\>ping 192.168.3.1
Pinging 192.168.3.1 with 32 bytes of data:
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Reply from 192.168.3.1: bytes=32 time=1ms TTL=254
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Ping statistics for 192.168.3.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms









Conclusion:

In this experiment, we successfully implemented static routing to facilitate direct communication between multiple networks. Through careful configuration of routers and the establishment of static routes, we demonstrated how to efficiently direct data packets across a predetermined path.

Lab 7: Dynamic routing Implementation using RIP

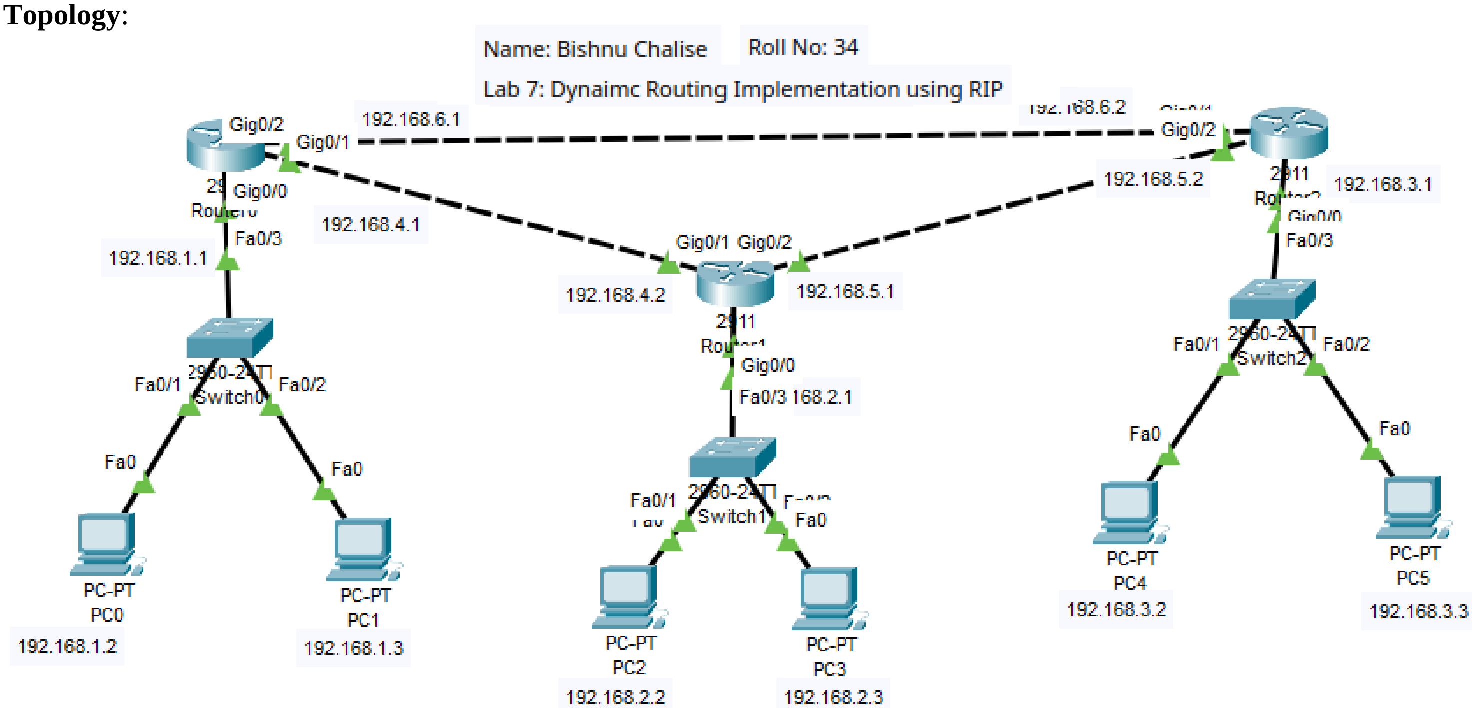
Objective: To configure and understand the dynamic routing protocol RIP.

Background: RIP (Routing Information Protocol) is a distance vector routing protocol used within an automatic system. The cost metric is the number of hops to reach the destination. The maximum number of hops that RIP can contain is 15 hops so, suitable for small networks.

IP Addressing Plan:

Device	Interface	IP	Subnet Mask	VLAN	Default Gateway
PC0	NIC	192.168.1.2	255.255.255.0	Default	192.168.1.1
PC1	NIC	192.168.1.3	255.255.255.0	Default	192.168.1.1
PC2	NIC	192.168.2.2	255.255.255.0	Default	192.168.2.1
PC3	NIC	192.168.2.3	255.255.255.0	Default	192.168.2.1
PC4	NIC	192.168.3.2	255.255.255.0	Default	192.168.3.1
PC5	NIC	192.168.3.3	255.255.255.0	Default	192.168.3.1
Router0	Gig0/0	192.168.1.1	255.255.255.0	Default	-
Router0	Gig0/1	192.168.4.1	255.255.255.0	Default	-
Router0	Gig0/2	192.168.6.1	255.255.255.0		-
Router1	Gig0/0	192.168.2.1	255.255.255.0	Default	-
Router1	Gig0/1	192.168.4.2	255.255.255.0	Default	-
Router1	Gig0/2	192.168.5.1	255.255.255.0		-
Router2	Gig0/0	192.168.3.1	255.255.255.0		-
Router2	Gig0/1	192.168.6.2	255.255.255.0		-
Router2	Gig0/2	192.168.5.2	255.255.255.0		-

Topology:



Procedure:

1. IP and Default gateway has been set in each PC.

Router0:

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#ip dhcp pool lan1
Router(dhcp-config)#network 192.168.1.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.1.1
```

Router1:

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#ip dhcp pool lan2
Router(dhcp-config)#network 192.168.2.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.2.1
```

Router2:

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.3.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#ip dhcp pool lan3
Router(dhcp-config)#network 192.168.3.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.3.1
```

2. IP address in the routers interfaces has been set.

Between Router0 to router1:

Router0:

```
Router(config)#int gig0/1
Router(config-if)#ip add 192.168.4.1 255.255.255.0
Router(config-if)#no shutdown
```

Router1:

```
Router(config)#int gig0/1
Router(config-if)#ip add 192.168.4.2 255.255.255.0
Router(config-if)#no shutdown
```

Between Router1 and Router2:

Router1:

```
Router(config)#int gig0/2
Router(config-if)#ip add 192.168.5.1 255.255.255.0
Router(config-if)#no shutdown
```

Router2:

```
Router(config)#int gig0/2
Router(config-if)#ip add 192.168.5.2 255.255.255.0
Router(config-if)#no shutdown
```

Between Router0 and Router2:

Router0:

```
Router(config)#int gig0/2

Router(config-if)#ip add 192.168.6.1 255.255.255.0
Router(config-if)#no shutdown
```

Router2:

```
Router(config)#int gig0/1
Router(config-if)#ip add 192.168.6.2 255.255.255.0
Router(config-if)#no shutdown
```

2.1. Following setup has been performed to illustrate dynamic routing using RIP

Router0:

```
Router(config)#router rip
Router(config-router)#network 192.168.1.0
Router(config-router)#network 192.168.2.0
Router(config-router)#network 192.168.3.0
Router(config-router)#network 192.168.4.0
Router(config-router)#network 192.168.5.0
Router(config-router)#network 192.168.6.0
```

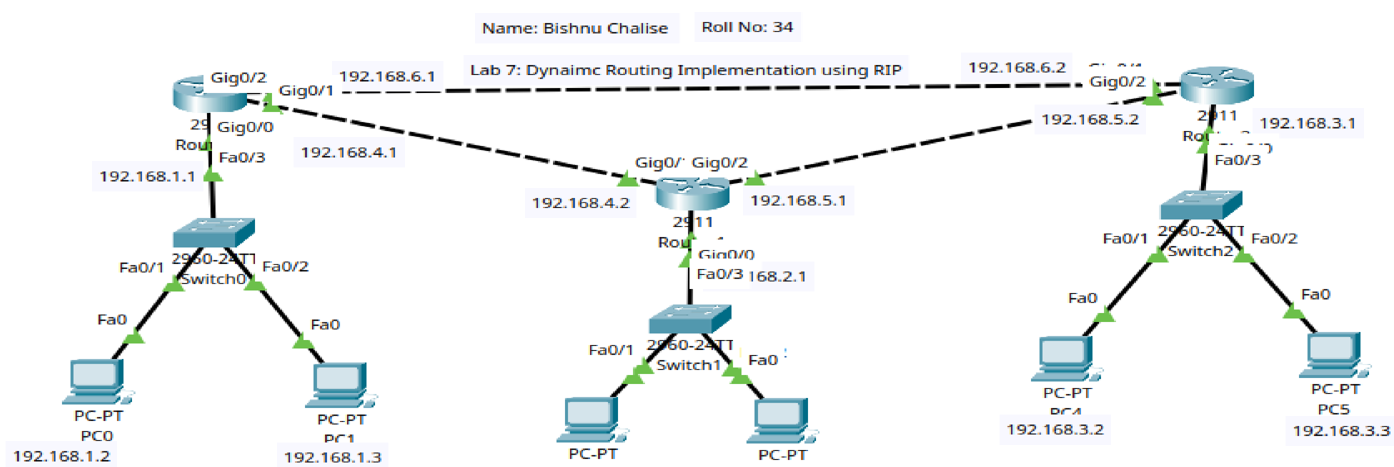
Router1:

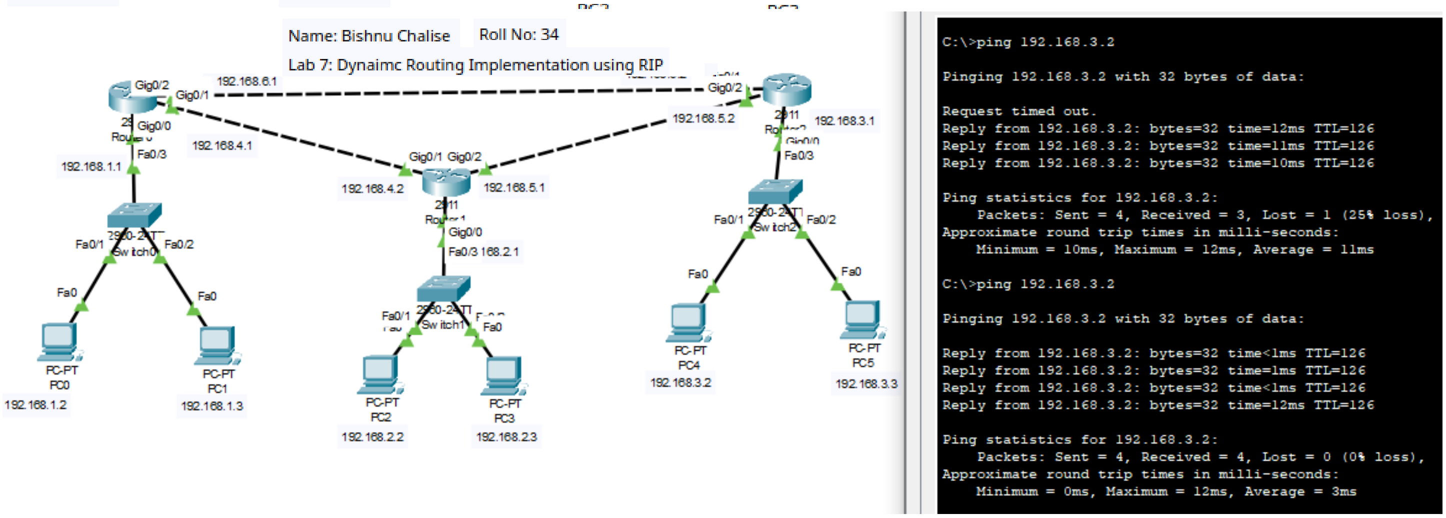
```
Router(config)#router rip
Router(config-router)#network 192.168.1.0
Router(config-router)#network 192.168.2.0
Router(config-router)#network 192.168.3.0
Router(config-router)#network 192.168.4.0
Router(config-router)#network 192.168.5.0
Router(config-router)#network 192.168.6.0
```

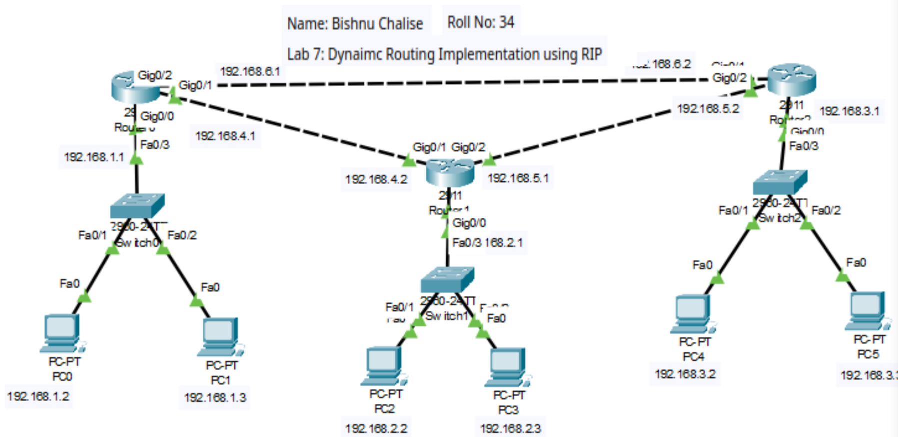
Router2:

```
Router(config)#router rip
Router(config-router)#network 192.168.1.0
Router(config-router)#network 192.168.2.0
Router(config-router)#network 192.168.3.0
Router(config-router)#network 192.168.4.0
Router(config-router)#network 192.168.5.0
Router(config-router)#network 192.168.6.0
```

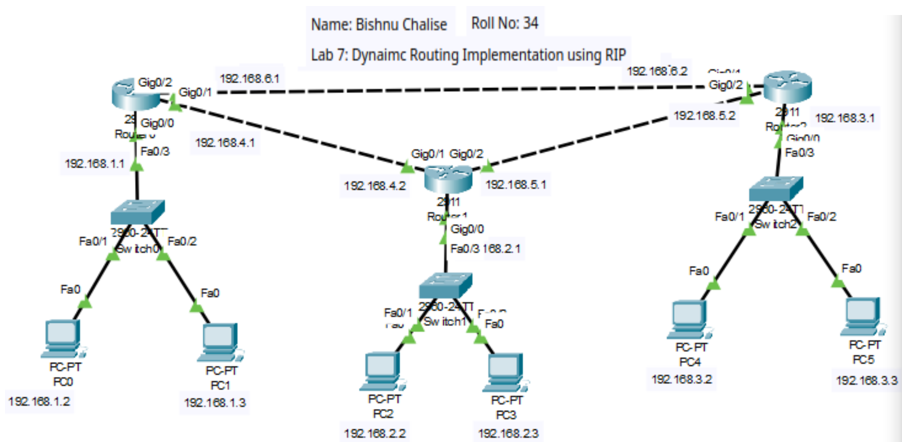
Validation:



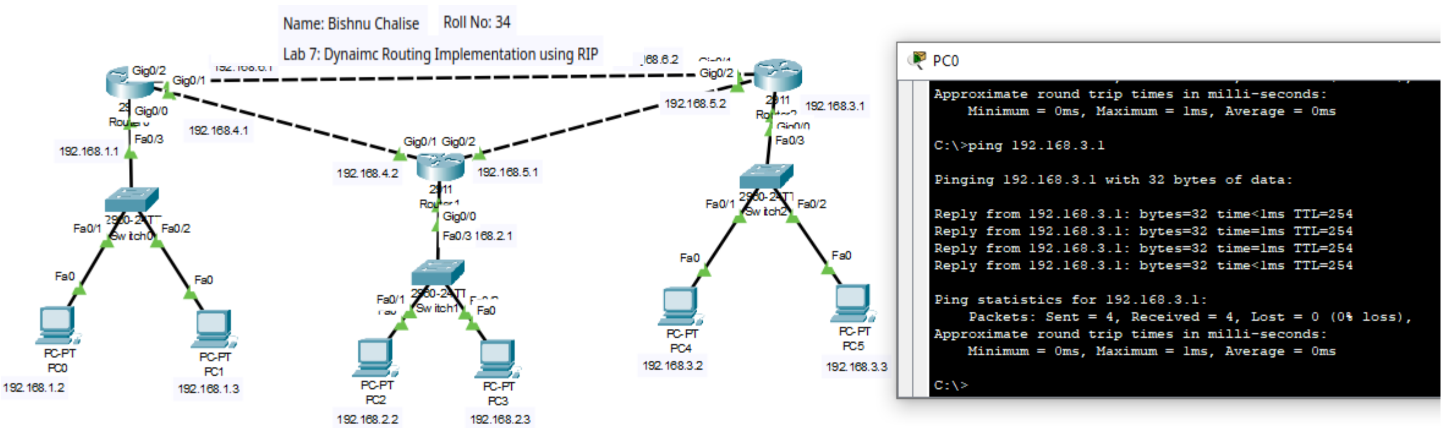


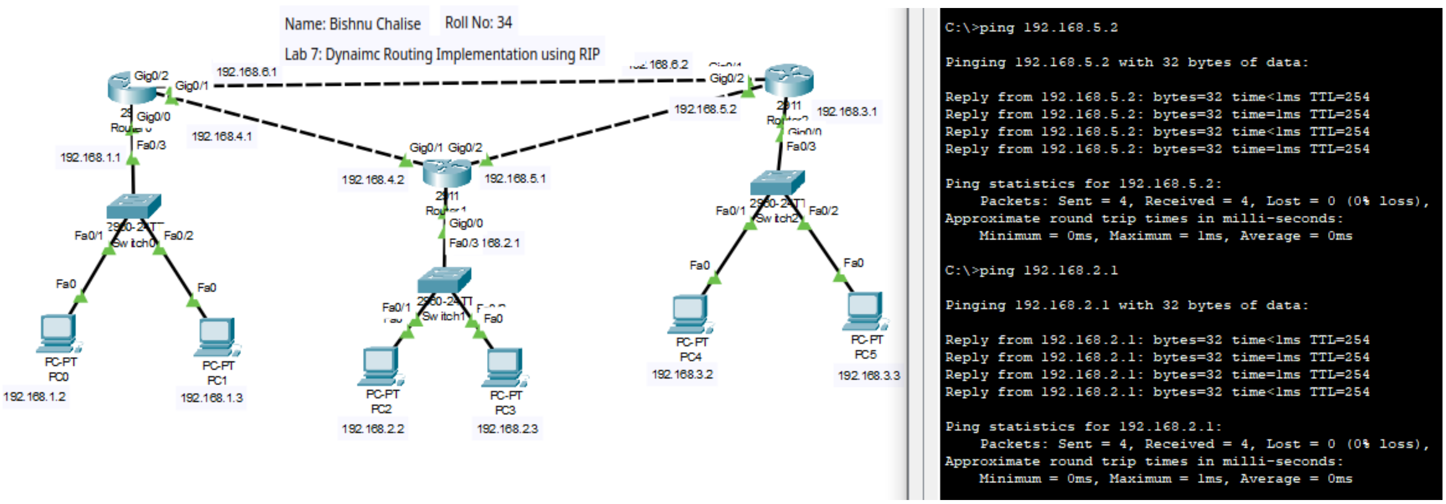



```
C:\>ping 192.168.4.1

Pinging 192.168.4.1 with 32 bytes of data:

Reply from 192.168.4.1: bytes=32 time<1ms TTL=255
Reply from 192.168.4.1: bytes=32 time<1ms TTL=255
Reply from 192.168.4.1: bytes=32 time=1ms TTL=255
Reply from 192.168.4.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.4.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 192.168.6.1

Pinging 192.168.6.1 with 32 bytes of data:

Reply from 192.168.6.1: bytes=32 time<1ms TTL=255
Reply from 192.168.6.1: bytes=32 time<1ms TTL=255
Reply from 192.168.6.1: bytes=32 time=1ms TTL=255
Reply from 192.168.6.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.6.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```




```
C:\>ping 192.168.6.2

Pinging 192.168.6.2 with 32 bytes of data:

Reply from 192.168.6.2: bytes=32 time<1ms TTL=254
Reply from 192.168.6.2: bytes=32 time<1ms TTL=254
Reply from 192.168.6.2: bytes=32 time<1ms TTL=254
Reply from 192.168.6.2: bytes=32 time=1ms TTL=254

Ping statistics for 192.168.6.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Reply from 192.168.4.2: bytes=32 time<1ms TTL=254
Reply from 192.168.4.2: bytes=32 time=34ms TTL=254
Reply from 192.168.4.2: bytes=32 time<1ms TTL=254
Reply from 192.168.4.2: bytes=32 time=1ms TTL=254

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 34ms, Average = 8ms
```





Conclusion: In this experiment, we successfully implemented dynamic routing using the Routing Information Protocol (RIP) in a simulated network environment. The primary objective was to understand how RIP facilitates the exchange of routing information between routers, allowing them to dynamically adjust to changes in network topology.

Lab 8: Dynamic Routing implementation using OSPF

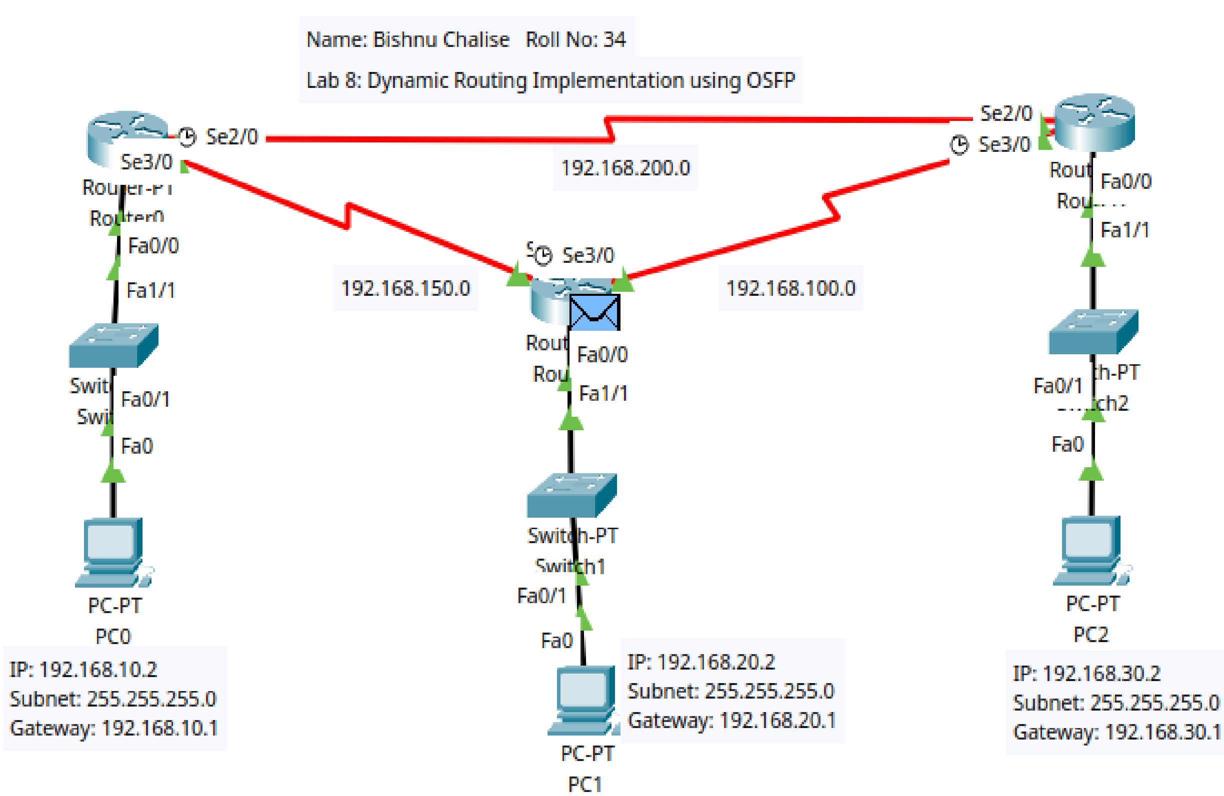
Objective: To configure and understand the OSPF as a dynamic routing protocol.

Background: **OSPF (Open Shortest Path First)** is a link-state routing protocol used in **IP networks** to find the best path for data to travel between routers. It's a dynamic routing protocol that enables routers to share information about the network topology, and it's commonly used in large enterprise networks.

IP Addressing

Device	Interface	IP	Subnet Mask	Default gateway
PC0	NIC	192.168.1.11	255.255.255.0	192.168.1.1
PC1	NIC	155.165.1.11	255.255.255.0	155.165.1.1
Router 0	Fa 0/0	192.168.1.1	255.255.255.0	-
Router 0	Se 2/0	10.0.0.2	255.0.0.0	-
Router 0	Se 3/0	20.0.0.1	255.0.0.0	-
Router 1	Fa 0/0	155.165.1.1	255.255.255.0	-
Router 1	Se 2/0	30.0.0.2	255.0.0.0	-
Router 1	Se 3/0	20.0.0.2	255.0.0.0	-
Router 2	Se 2/0	10.0.0.1	255.0.0.0	-
Router 2	Se 3/0	30.0.0.1	255.0.0.0	-

Topology



Procedure:

1. IP and Default gateway in each PC has been set.

    1.1. Default Gateway:

    Router0:

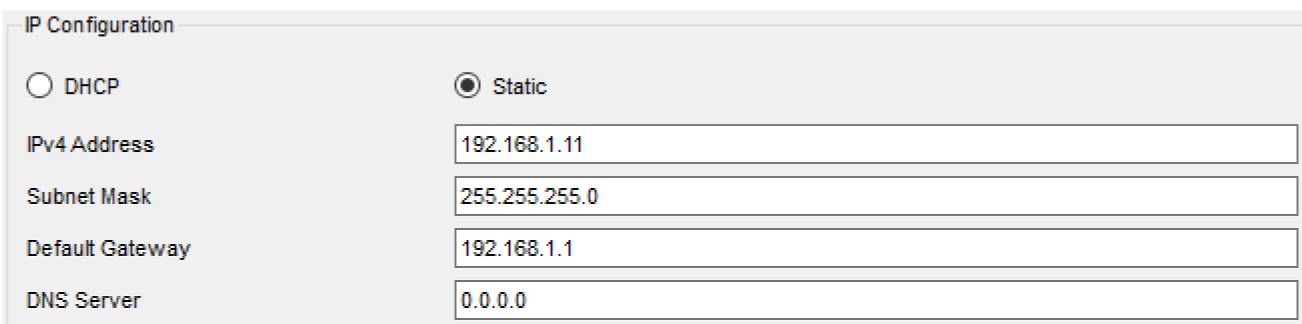
```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
```

    Router1:

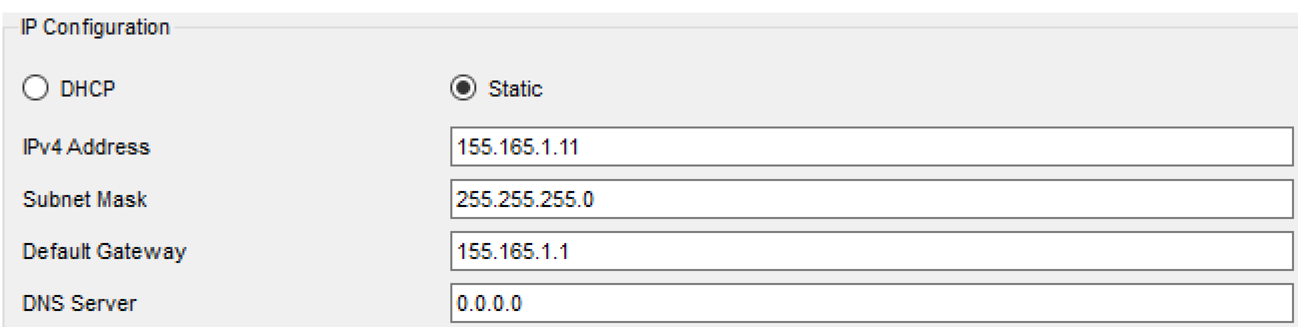
```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 155.165.1.1 255.255.0.0
Router(config-if)#no shutdown
```

    1.2. IP Address:

    PC0:



    PC1:



2. IP addresses in the routers Interfaces has been set as shown below.
   Between Router2 and router0:

    Router2:

```
Router(config)#interface Serial2/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
```

Router0:

```
Router(config)#interface Serial2/0
Router(config-if)#clock rate 64000

Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown
```

Between router0 and router1

Router0:

```
Router(config)#interface Serial3/0
Router(config-if)#clock rate 64000
Router(config-if)#ip address 20.0.0.1 255.0.0.0

Router(config-if)#no shutdown
```

Router1:

```
Router(config)#interface Serial3/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0

Router(config-if)#no shutdown
```

Between router2 and router1:

Router2:

```
Router(config)#interface Serial3/0
Router(config-if)#clock rate 64000
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown
```

Router1:

```
Router(config)#interface Serial2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown
```

3. The following OSPF configure has been set in each router.

Router0:

```
Router(config)#router ospf 1
Router(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router(config-router)#network 20.0.0.0 0.255.255.255 area 0
```

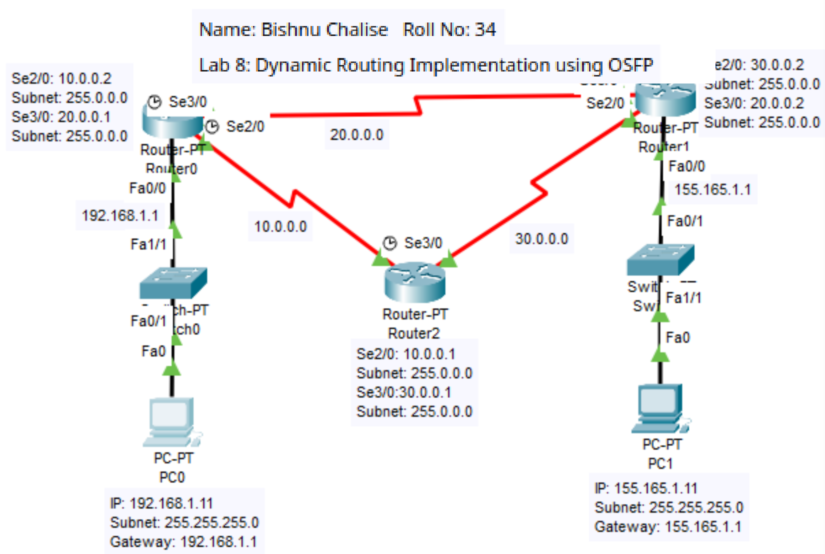
Router2:

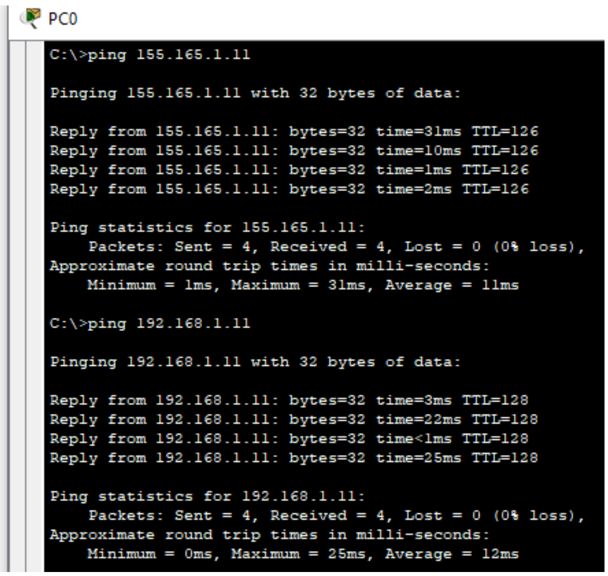
```
Router(config)#router ospf 1
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
```

Router1:

```
Router(config)#router ospf 1
Router(config-router)#network 20.0.0.0 0.255.255.255 area 0
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#network 155.165.1.0 0.0.255.255 area 0
```

Validation:





Conclusion: we successfully implemented dynamic routing using the **Open Shortest Path First (OSPF)** protocol. The primary objectives were to explore how OSPF dynamically updates routing tables and adapts to network topology changes, ensuring efficient and accurate routing of data packets within a network.

Lab 9: VLAN Setup

Objectives: Implement VLAN using packet tracer.

Background: A **Virtual Local Area Network (VLAN)** is a network segmentation technology used to create logically separated network domains within the same physical network infrastructure. VLANs allow devices in different geographical locations or departments to communicate as if they were on the same physical network while keeping other devices in the same infrastructure isolated. VLANs operate at Layer 2 (Data Link Layer) of the OSI model. Each VLAN is treated as a separate subnet, and devices in the same VLAN can communicate directly with each other. Devices in different VLANs require a Layer 3 device (such as a router or Layer 3 switch) for communication between VLANs, known as inter-VLAN routing.

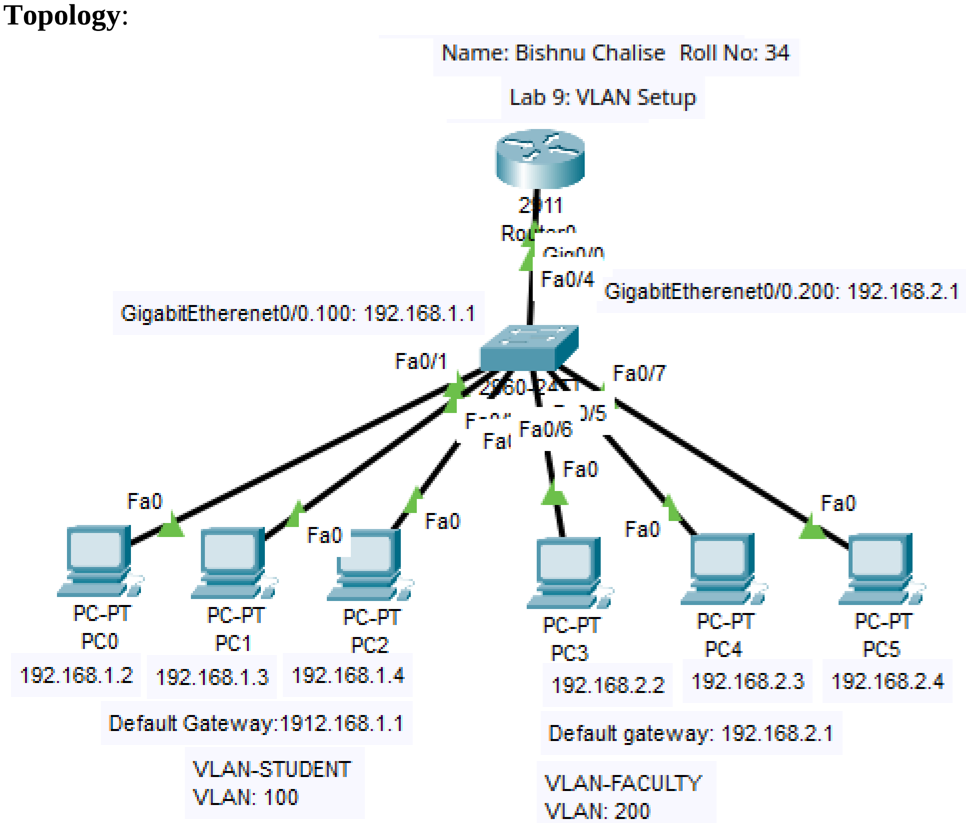
Types of VLANs:

Default VLAN: This is the VLAN that all ports on a switch are assigned to by default. It typically has an ID of 1.
Data VLAN: A VLAN used for carrying user-generated data traffic (e.g., VLAN 10 for accounting).
Voice VLAN: A VLAN specifically designated for VoIP traffic to improve the quality of service.
Management VLAN: Used for managing network devices. Only network administrators have access to this VLAN.
Native VLAN: This is the VLAN that handles untagged traffic on a trunk port.

IP Addressing Plan:

Device	Interface	IP Address	Subnet Mask	VLAN	Gateway
PC0	NIC	192.168.1.2	255.255.255.0	100	192.168.1.1
PC1	NIC	192.168.1.3	255.255.255.0	100	192.168.1.1
PC2	NIC	192.168.1.4	255.255.255.0	100	192.168.1.1
PC3	NIC	192.168.2.2	255.255.255.0	200	192.168.2.1
PC4	NIC	192.168.2.3	255.255.255.0	200	192.168.2.1
PC5	NIC	192.168.2.4	255.255.255.0	200	192.168.2.1
Router0	GigabitEthernet0/0.100	192.168.1.1	255.255.255.0	-	-
Router0	GigabitEthernet0/0.200	192.168.2.1	255.255.255.0	-	-

Topology:



Procedure:
1. The topology has been created as shown above.
2. The IP address to each PC has been assigned as shown above.
3. VLANs has been created as shown below:
    3.1. The following configurations has been performed in the switch:

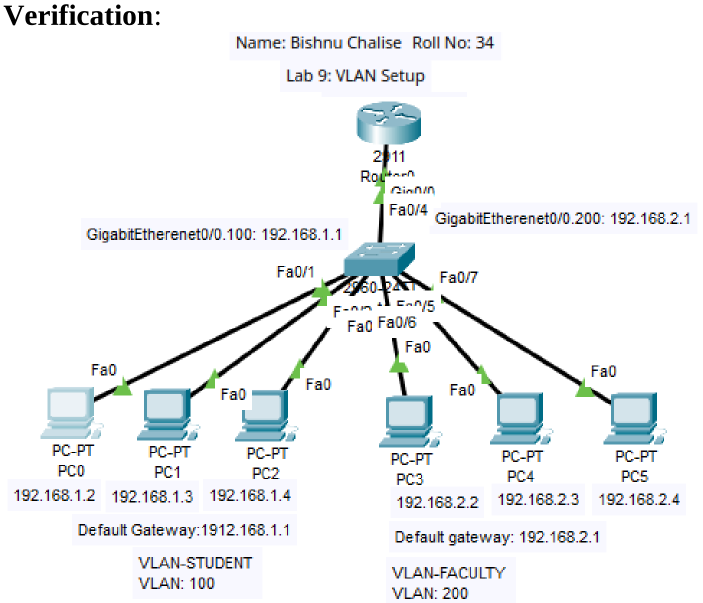
```
Switch>enable
Switch#show vlan br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 100
Switch(config-vlan)#name student
Switch(config-vlan)#vlan 200
Switch(config-vlan)#name faculty
Switch(config-vlan)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console
```

```
Switch#show vlan br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
100  student                          active
200  faculty                          active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fa0/1
Switch(config-if)#switchport access vlan 100
Switch(config-if)#switchport mode access
Switch(config-if)#int fa0/2
Switch(config-if)#switchport access vlan 100
Switch(config-if)#switchport mode access
Switch(config-if)#int fa0/3
Switch(config-if)#switchport access vlan 100
Switch(config-if)#switchport mode access
Switch(config-if)#int fa0/5
Switch(config-if)#switchport access vlan 200
Switch(config-if)#switchport mode access
Switch(config-if)#int fa0/6
Switch(config-if)#switchport access vlan 200
Switch(config-if)#switchport mode access
Switch(config-if)#int fa0/7
Switch(config-if)#switchport access vlan 200
Switch(config-if)#switchport mode access
Switch(config-if)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show vlan br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/4, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24, Gig0/1, Gig0/2
100  student                          active    Fa0/1, Fa0/2, Fa0/3
200  faculty                          active    Fa0/5, Fa0/6, Fa0/7
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fa0/4
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to up
```

3.2. The following configurations are performed in the router:

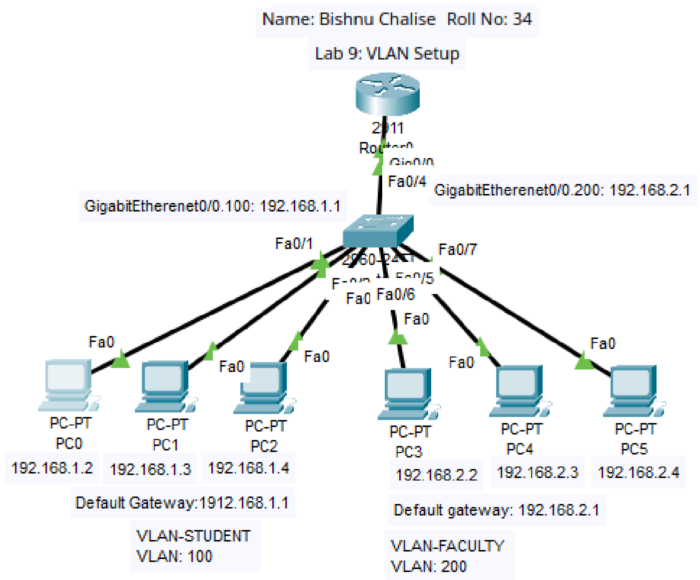
```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#int gig0/0.100
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.100, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.100, changed state to
up

Router(config-subif)#encapsulation dot1Q 100
Router(config-subif)#ip add 192.168.1.1 255.255.255.0
Router(config-subif)#no shut
Router(config-subif)#exit
Router(config)#int gig0/0.200
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.200, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.200, changed state to
up

Router(config-subif)#encapsulation dot1Q 200
Router(config-subif)#ip add 192.168.2.1 255.255.255.0
Router(config-subif)#no shut
Router(config-subif)#exit
```

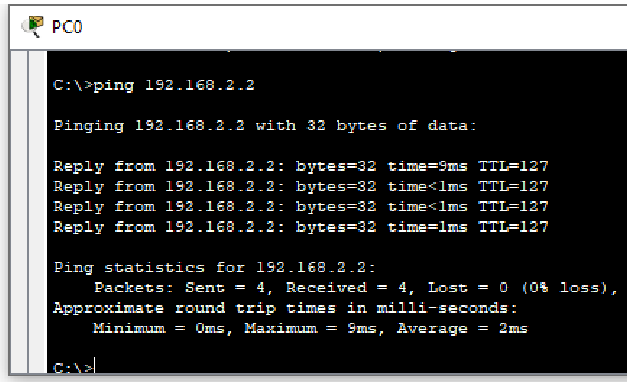
Verification:

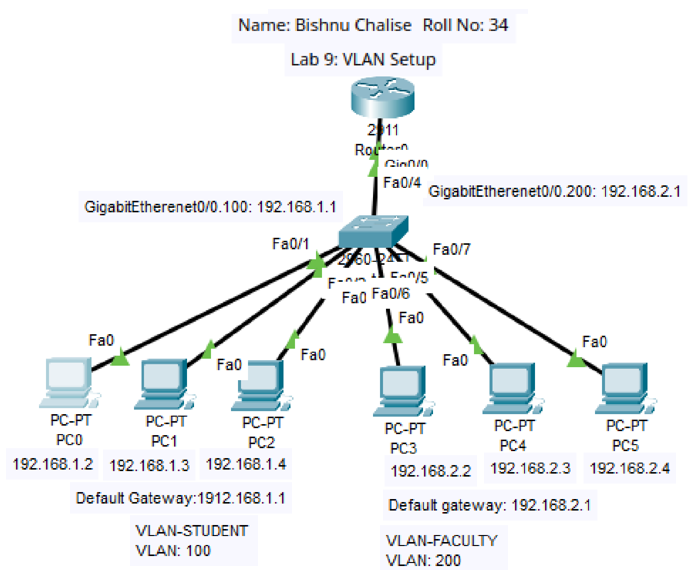


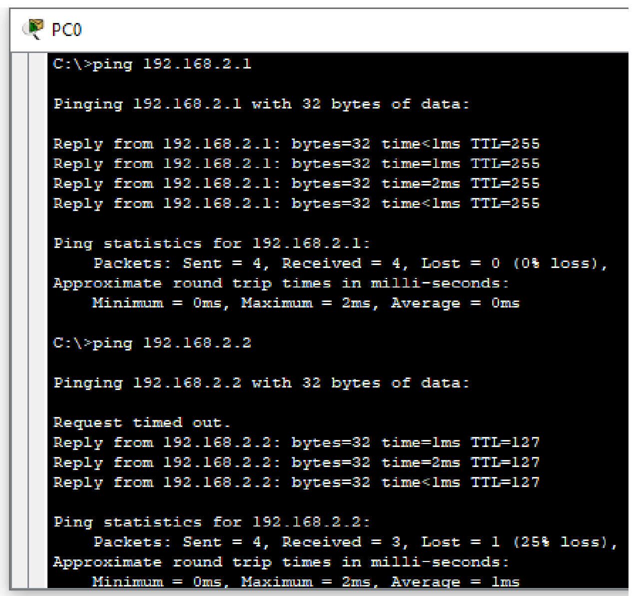
PC0

```
C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=7ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time=11ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 11ms, Average = 4ms

C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```









Conclusion: In this experiment, we successfully implemented VLAN (Virtual Local Area Network) configurations on network switches, demonstrating how VLANs can be used to segment network traffic logically, even within the same physical network. VLANs help improve network efficiency, security, and management by isolating different types of traffic and reducing broadcast domains.

Lab 10: Router Access list Configuration

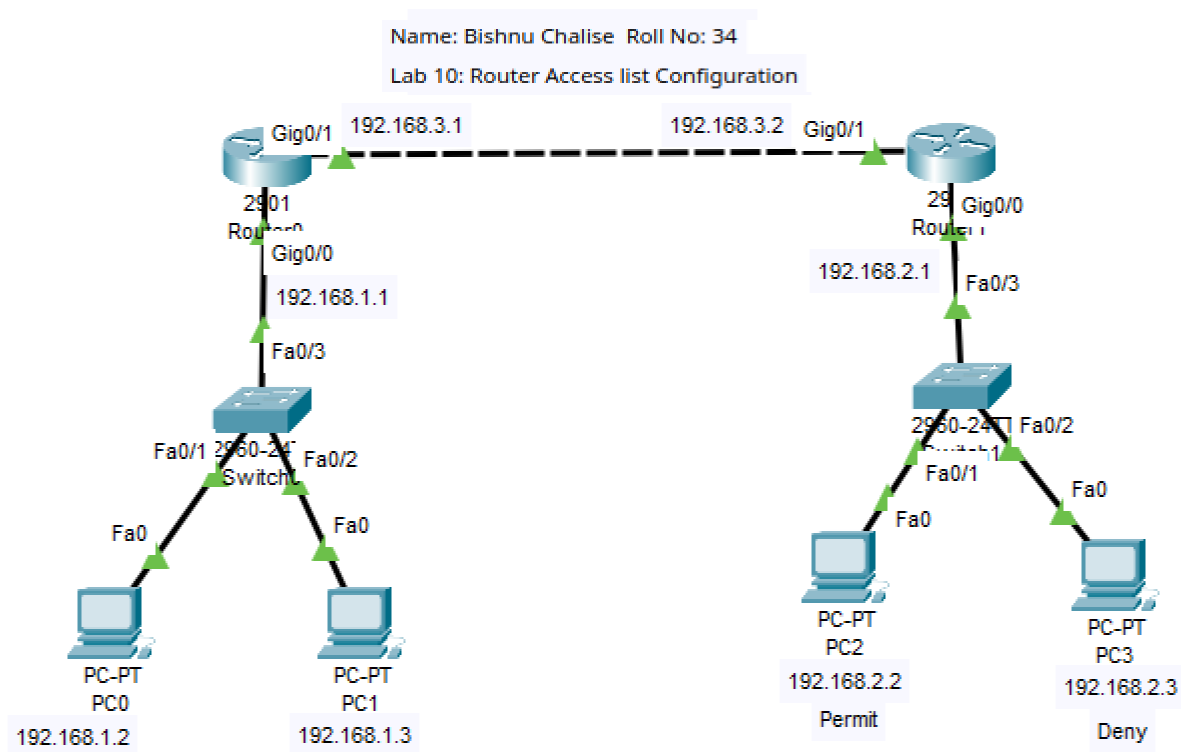
Objectives: Implement Router Access list Configuration using packet tracer.

Background: Access Control Lists (ACLs) are used in routers to control traffic flow and enhance security by allowing or denying specific packets based on criteria such as source or destination IP addresses, protocols, or ports. Configuring ACLs on a router helps manage access to and from the network, controlling who can enter or exit the network. There are two main types of ACLs:
Standard ACLs – Filters traffic based only on the source IP address.
Extended ACLs – Filters traffic based on various criteria such as source and destination IP address, protocol, and port numbers.

IP Addressing Plan:

Device	Interface	IP Address	Subnet Mask	Gateway
PC0	NIC	192.168.1.2	255.255.255.0	192.168.1.1
PC1	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC2	NIC	192.168.2.2	255.255.255.0	192.168.2.1
PC3	NIC	192.168.2.3	255.255.255.0	192.168.2.1
Router0	GigabitEthernet0/0	192.168.1.1	255.255.255.0	-
Router0	GigabitEthernet0/1	192.168.3.1	255.255.255.0	-
Router1	GigabitEthernet0/0	192.168.2.1	255.255.255.0	-
Router1	GigabitEthernet0/1	192.168.3.2	255.255.255.0	-

Topology:



Procedure:

1. The topology has been created as shown above.
2. The IP address to each PC has been assigned.
   Router0:

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.1.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#ip dhcp pool lan1
Router(dhcp-config)#network 192.168.1.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.1.1
Router(dhcp-config)#exit
```

   Router1:

```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.2.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#ip dhcp pool lan2
Router(dhcp-config)#network 192.168.2.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.2.1
Router(dhcp-config)#exit
```

3. IP address in the routers interfaces and Static route has been set.
   Router0:

```
Router(config)#int gig0/1
Router(config-if)#ip add 192.168.3.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
Router(config-if)#ip route 192.168.2.0 255.255.255.0 192.168.3.2
```

```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
S    192.168.2.0/24 [1/0] via 192.168.3.2
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/1
L       192.168.3.1/32 is directly connected, GigabitEthernet0/1

Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.1.1     YES manual up                    up
GigabitEthernet0/1     192.168.3.1     YES manual up                    up
Vlan1                  unassigned      YES unset  administratively down down
```

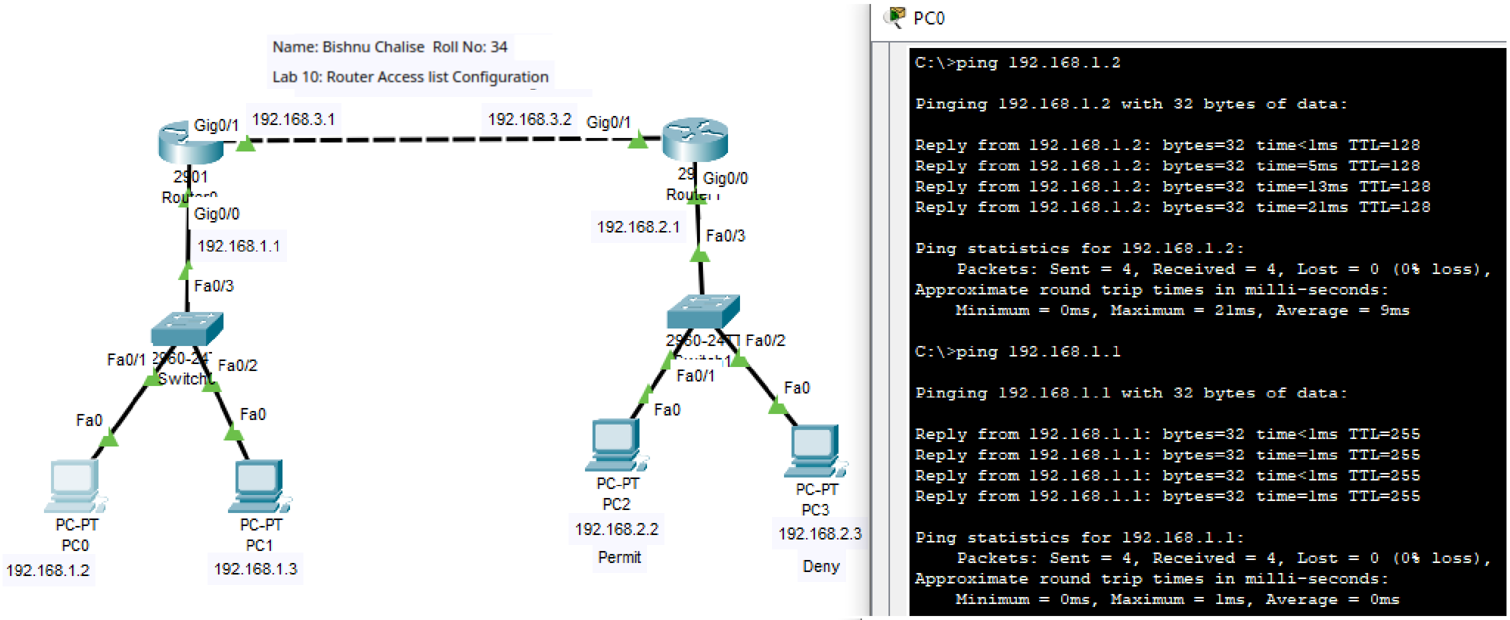
Router1:

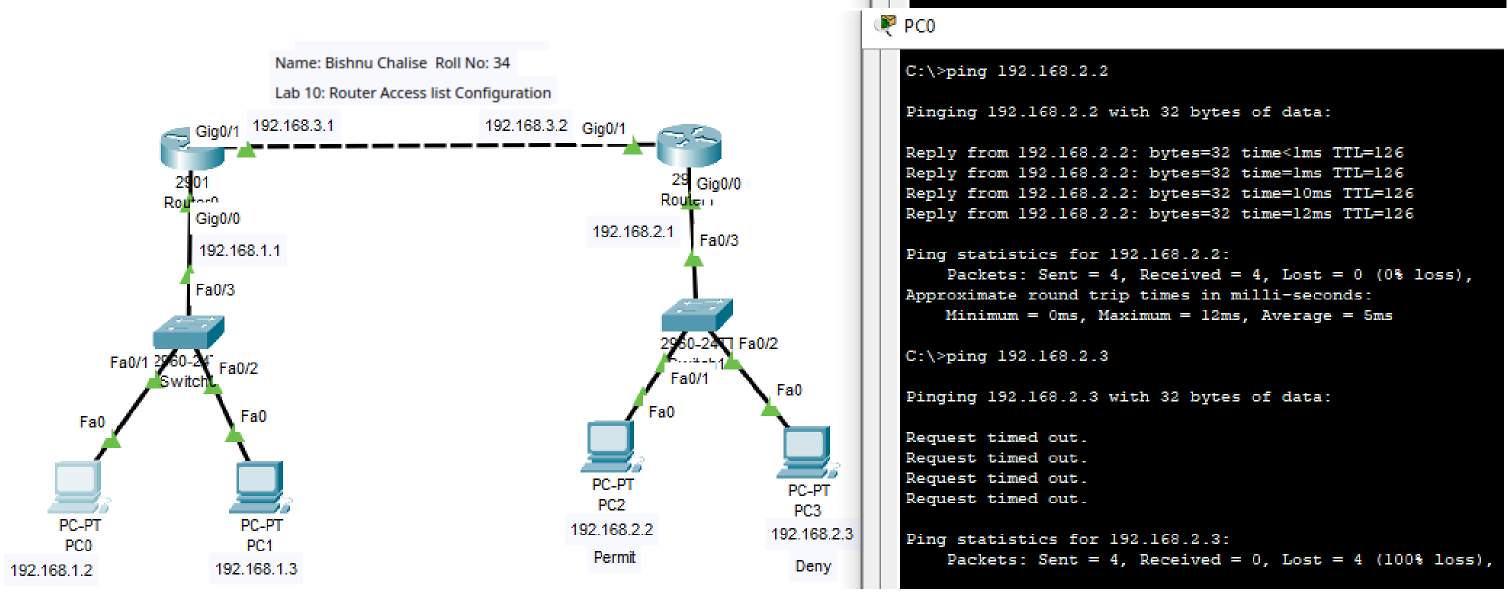
```
Router(config)#int gig0/1
Router(config-if)#ip add 192.168.3.2 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

Router(config-if)#ip route 192.168.1.0 255.255.255.0 192.168.3.1

Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

S    192.168.1.0/24 [1/0] via 192.168.3.1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/0
L       192.168.2.1/32 is directly connected, GigabitEthernet0/0
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/1
L       192.168.3.2/32 is directly connected, GigabitEthernet0/1

Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.2.1     YES manual up                    up
GigabitEthernet0/1     192.168.3.2     YES manual up                    up
Vlan1                  unassigned      YES unset  administratively down down
```

4. Access List Configuration has been done as shown below:
   The following configurations has been performed in the Router0:

```
Router(config)#access-list 1 deny host 192.168.2.3
Router(config)#access-list 1 permit any
Router(config)#int gig0/1
Router(config-if)#ip access-group 1 out
```

Validation:





Conclusion: In this experiment we successfully implemented a Router Access Control List (ACL) configuration using Cisco Packet Tracer to control network traffic between two subnets. The primary objective was to demonstrate how ACLs can be used to enhance network security by selectively allowing or denying traffic based on specified criteria.