Experiment No : 01

,, ,, ,,

Roll No : TE-43

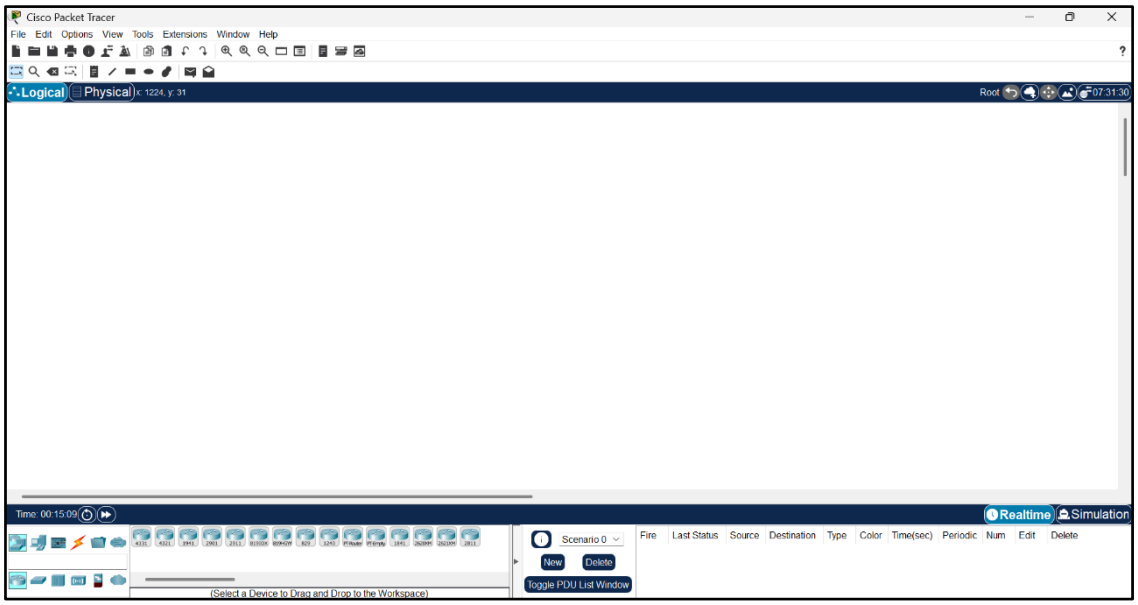
Name : Sanika Nilesh Patil

Title : Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool.

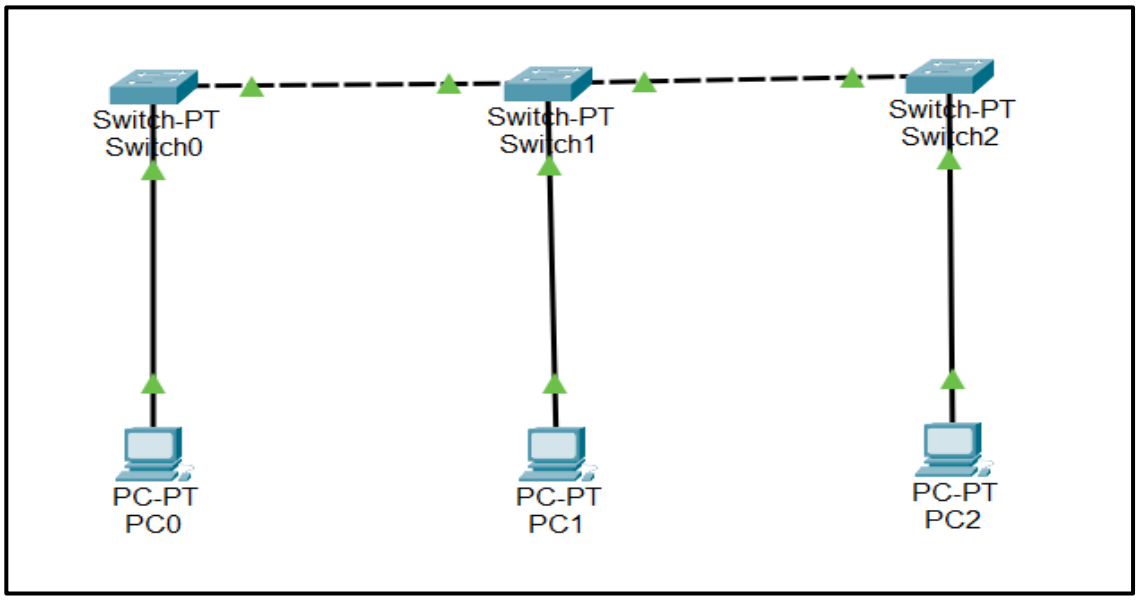
,, ,, ,,

Bus Topology :

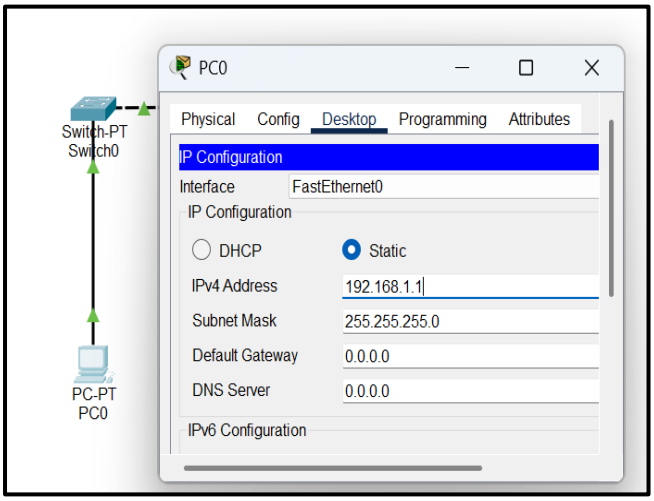
Step 1: Launch Packet Tracer.

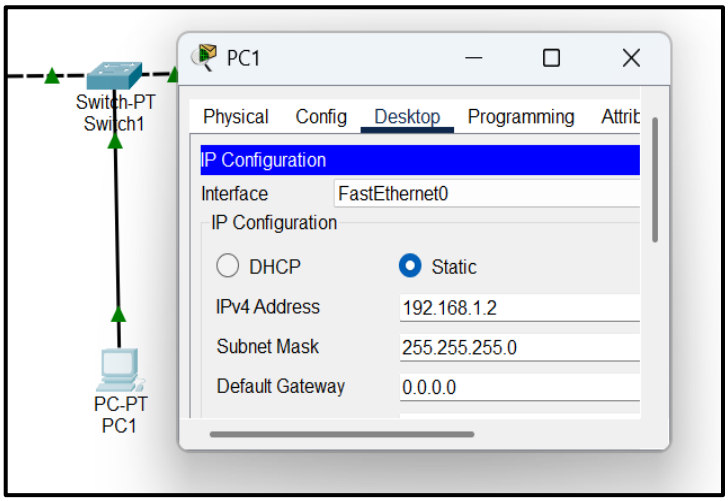


Step 2: Build bus topology.

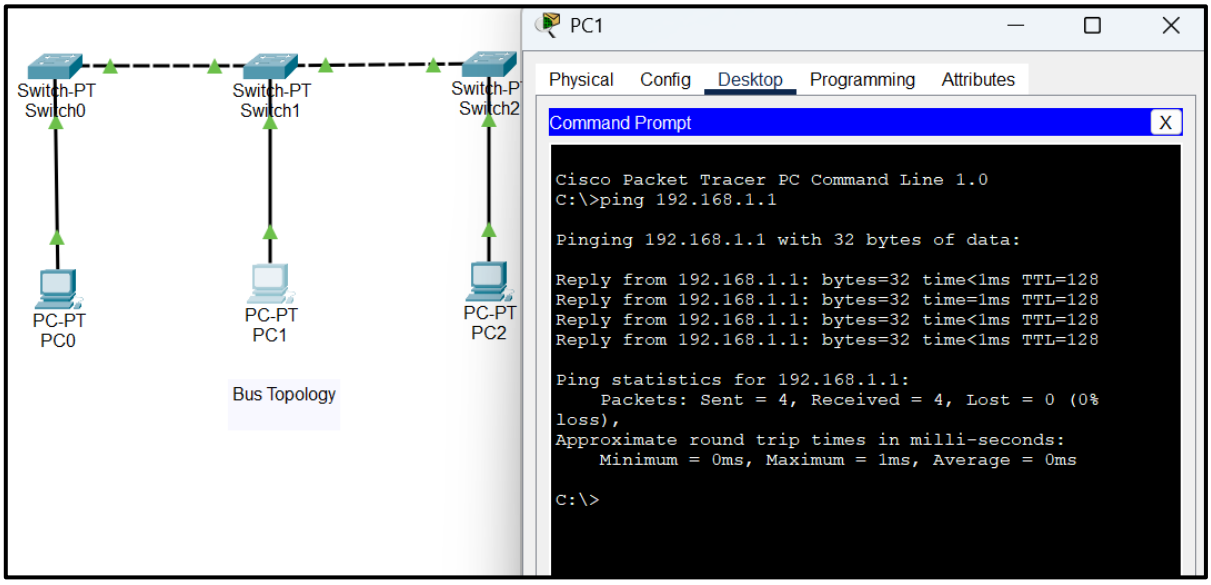


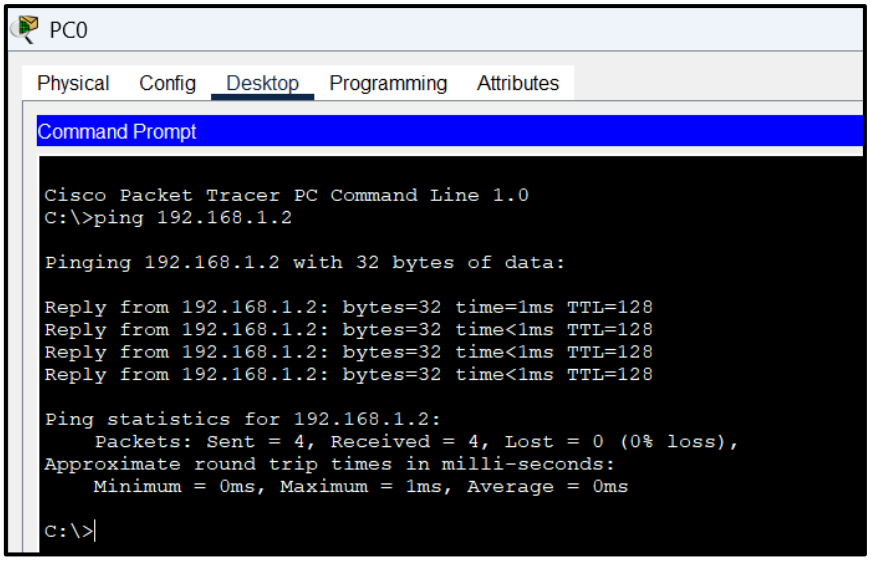
Step 3: Configure IP addresses and subnet masks on all PCs.





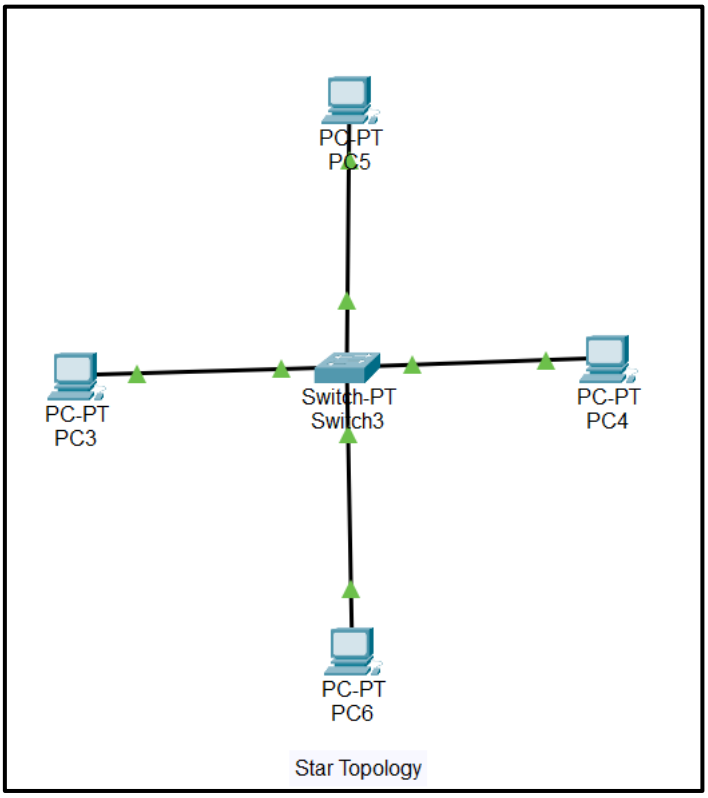
Step 4: Test network connectivity using the PING command between PCs.



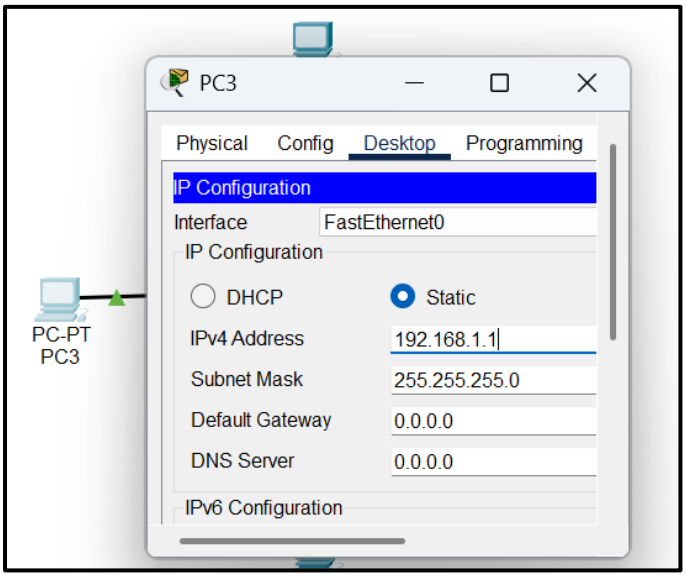


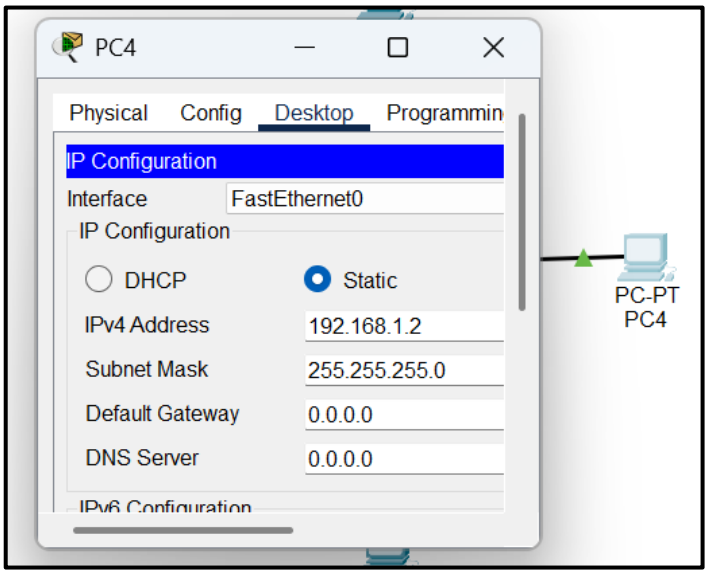
Star Topology :

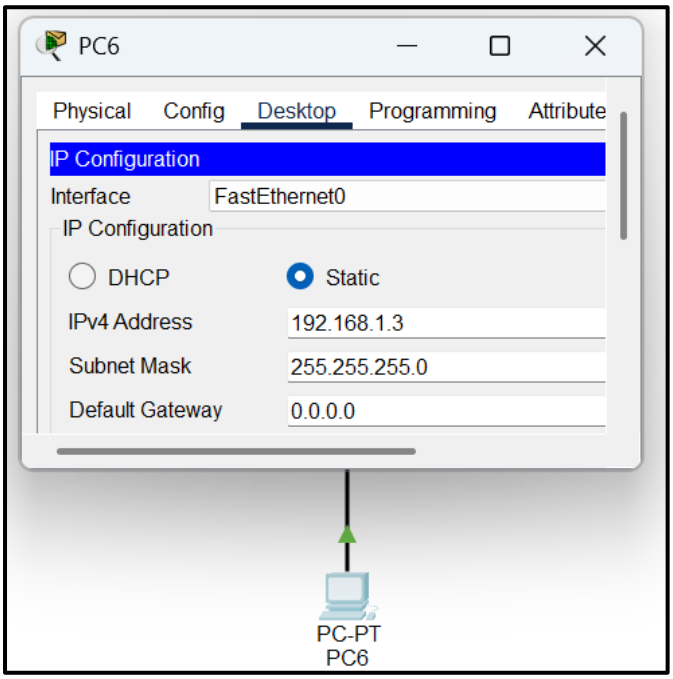
Step 1: Build star topology.

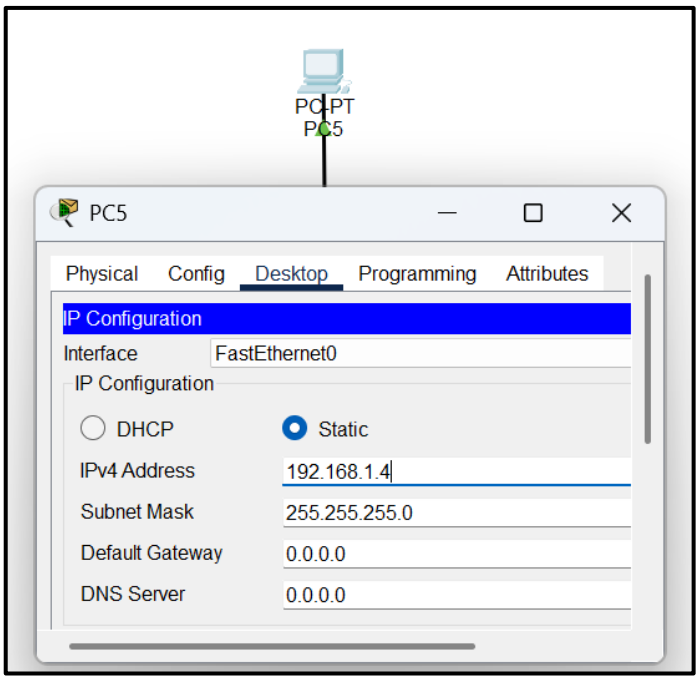


Step 2: Configure IP addresses and subnet masks on all PCs.

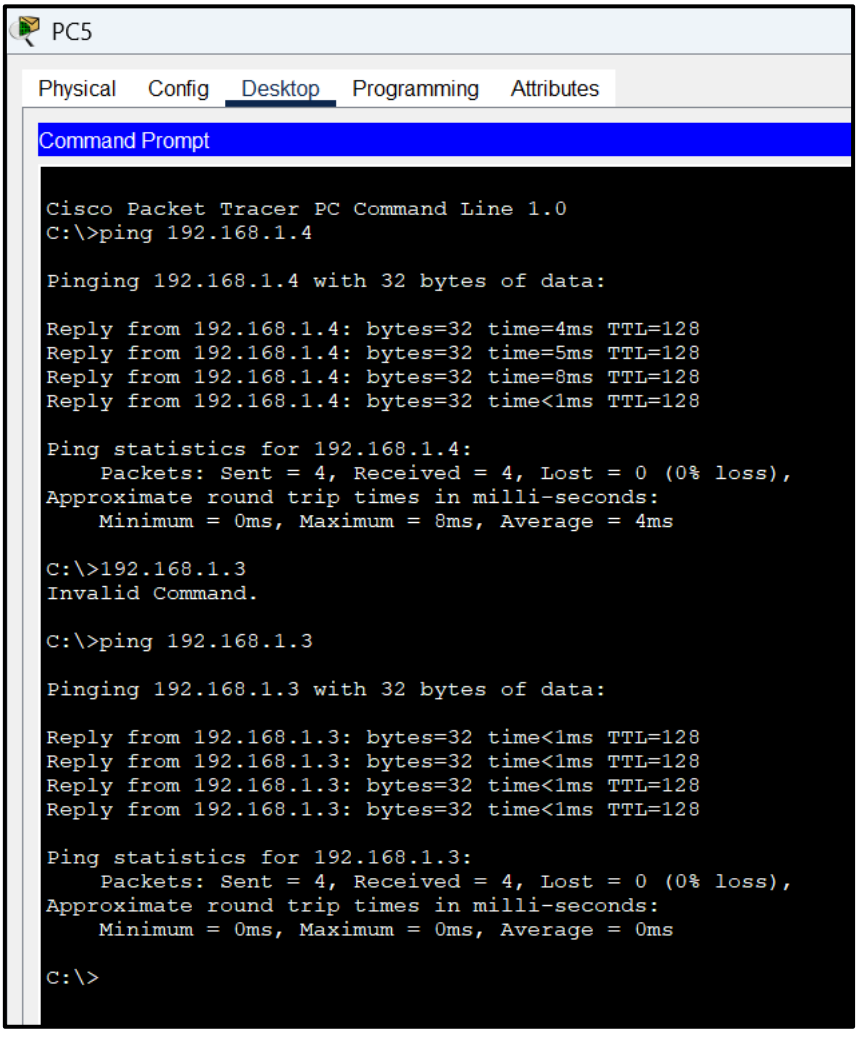






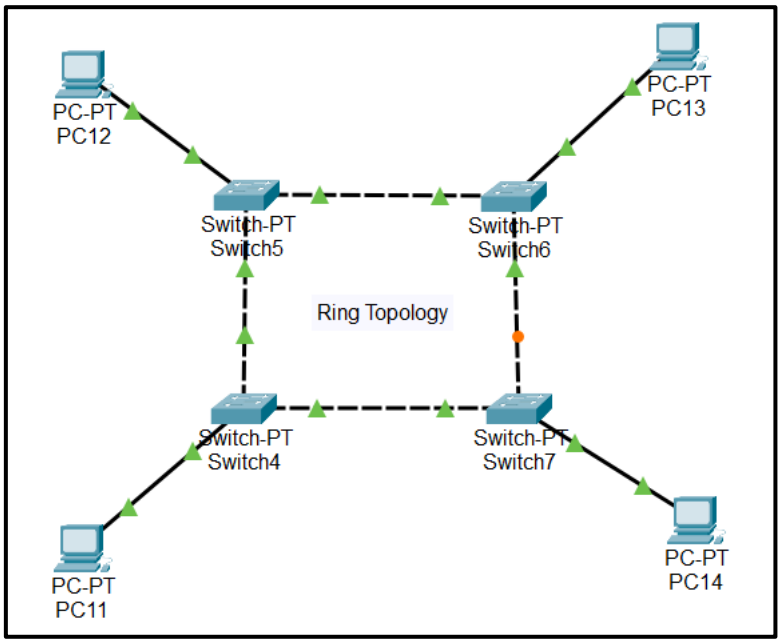


Step 3: Test network connectivity using the PING command between PCs.

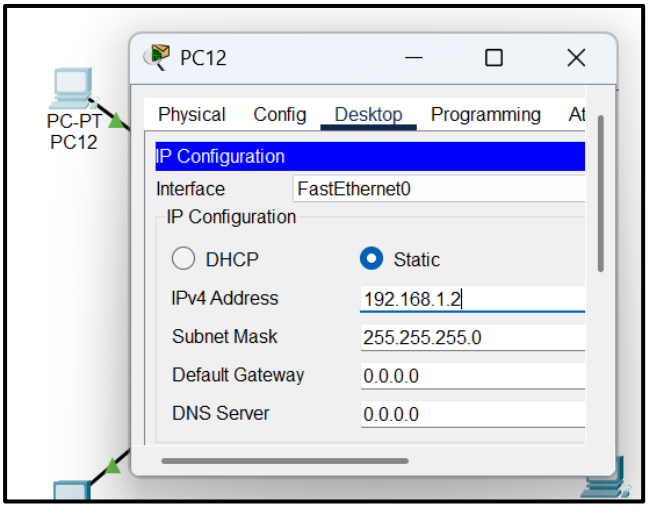


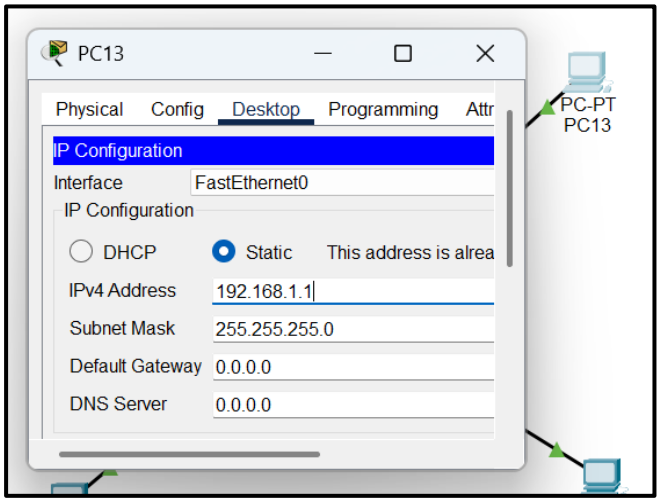
Ring Topology :

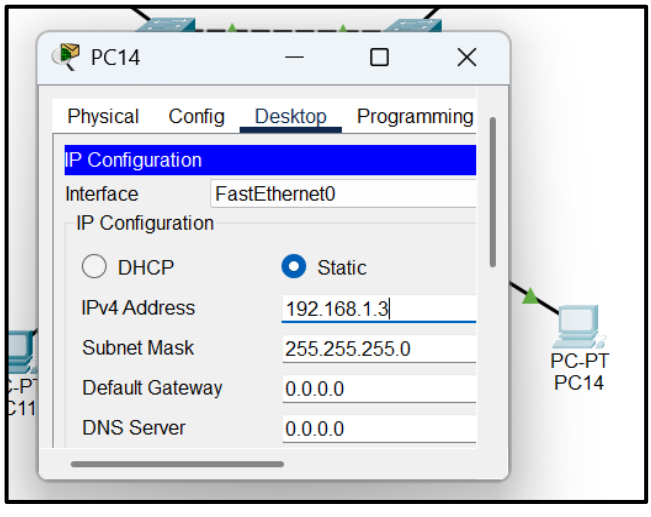
Step 1: Build Ring topology.

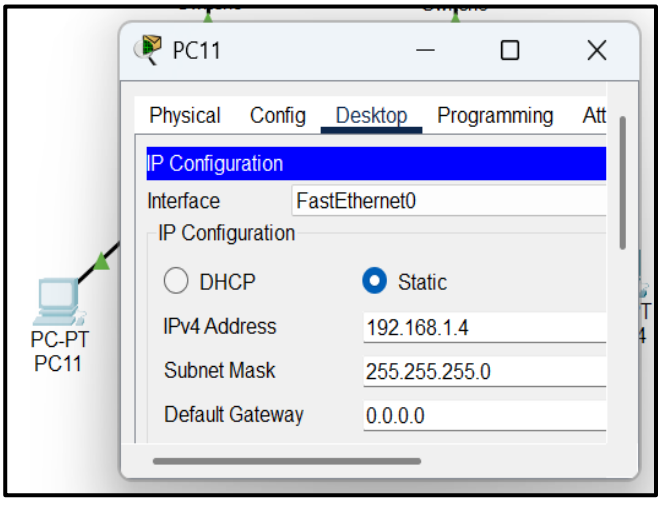


Step 2: Configure IP addresses and subnet masks on all PCs.

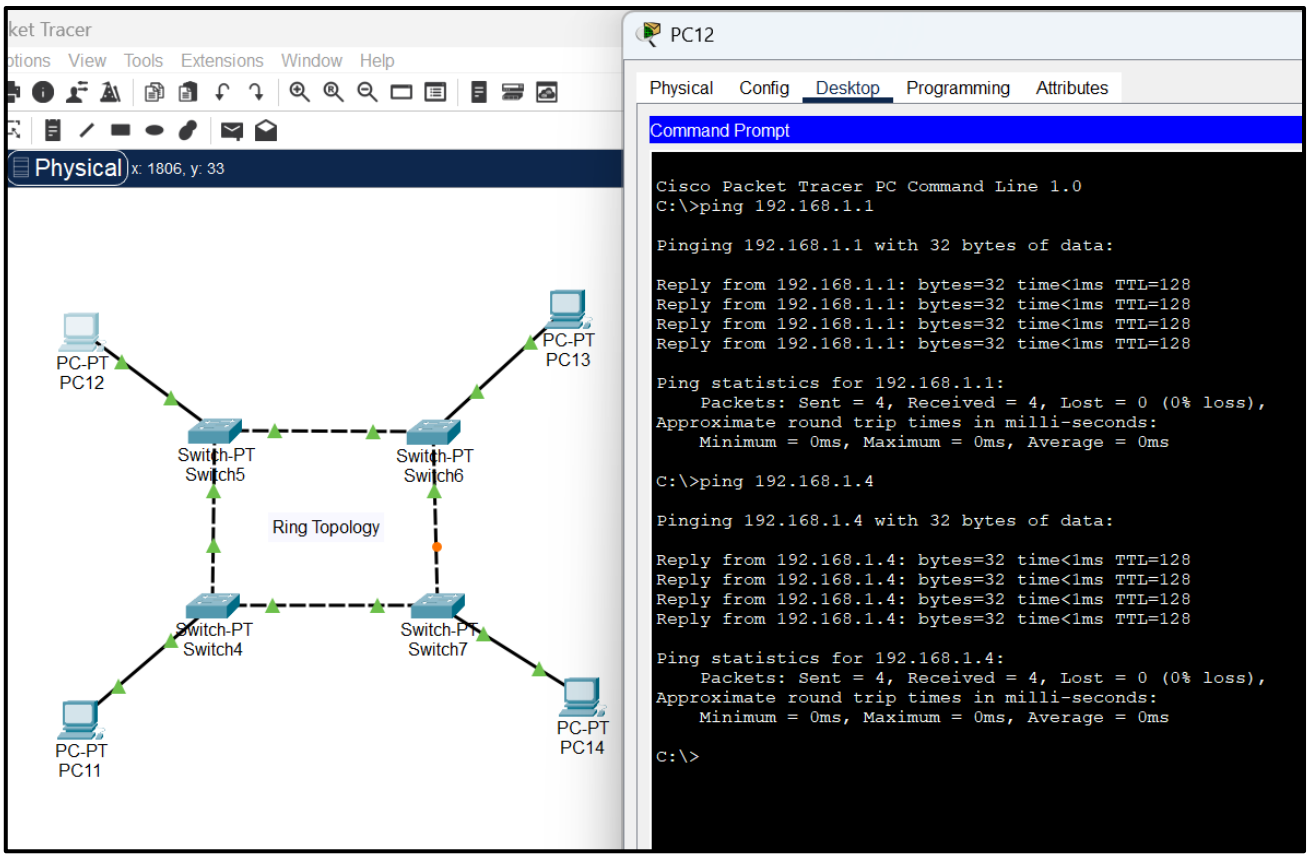


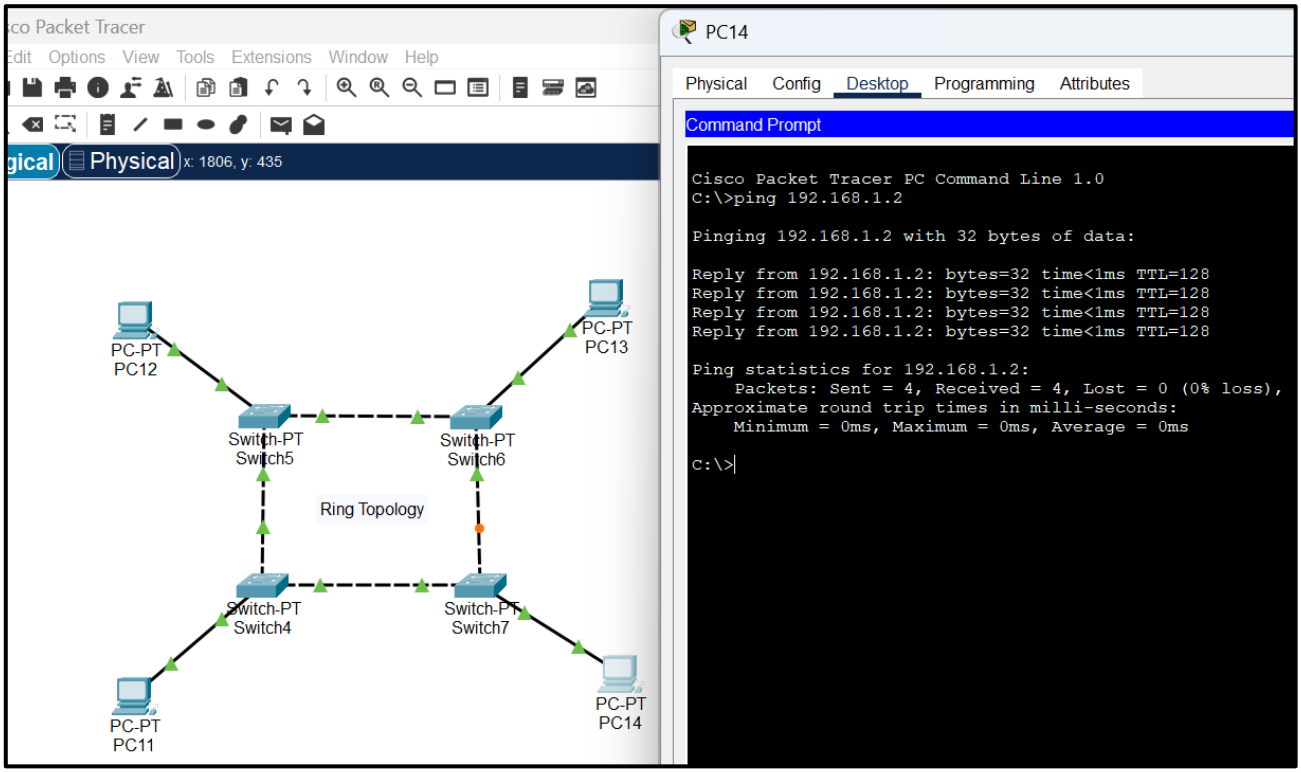






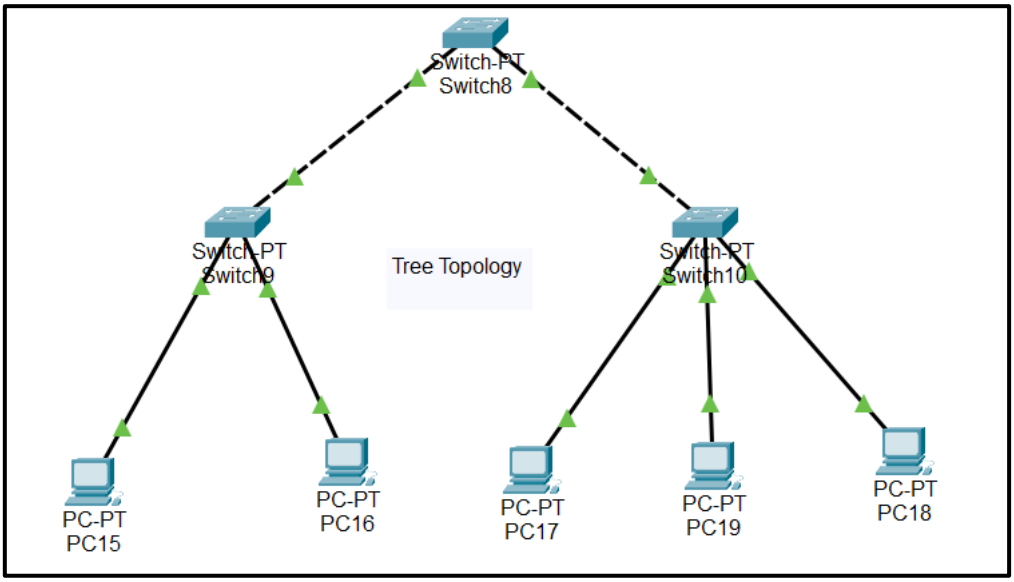
Step 3: Test network connectivity using the PING command between PCs.



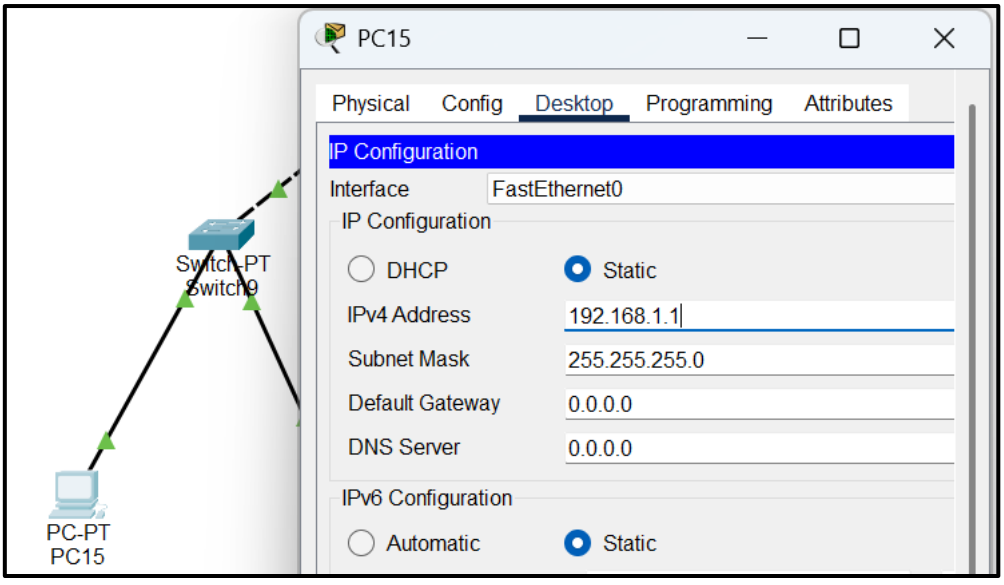


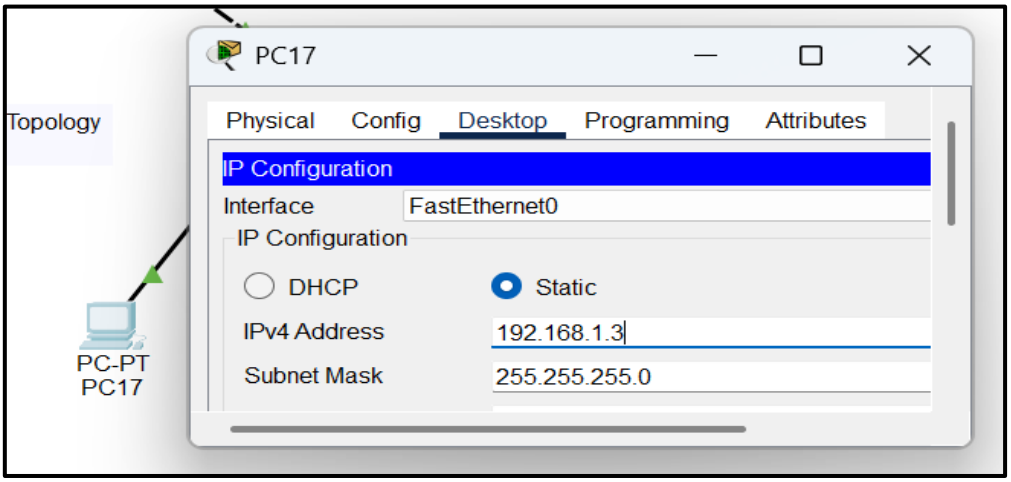
Tree Topology :

Step 1: Build Tree topology.

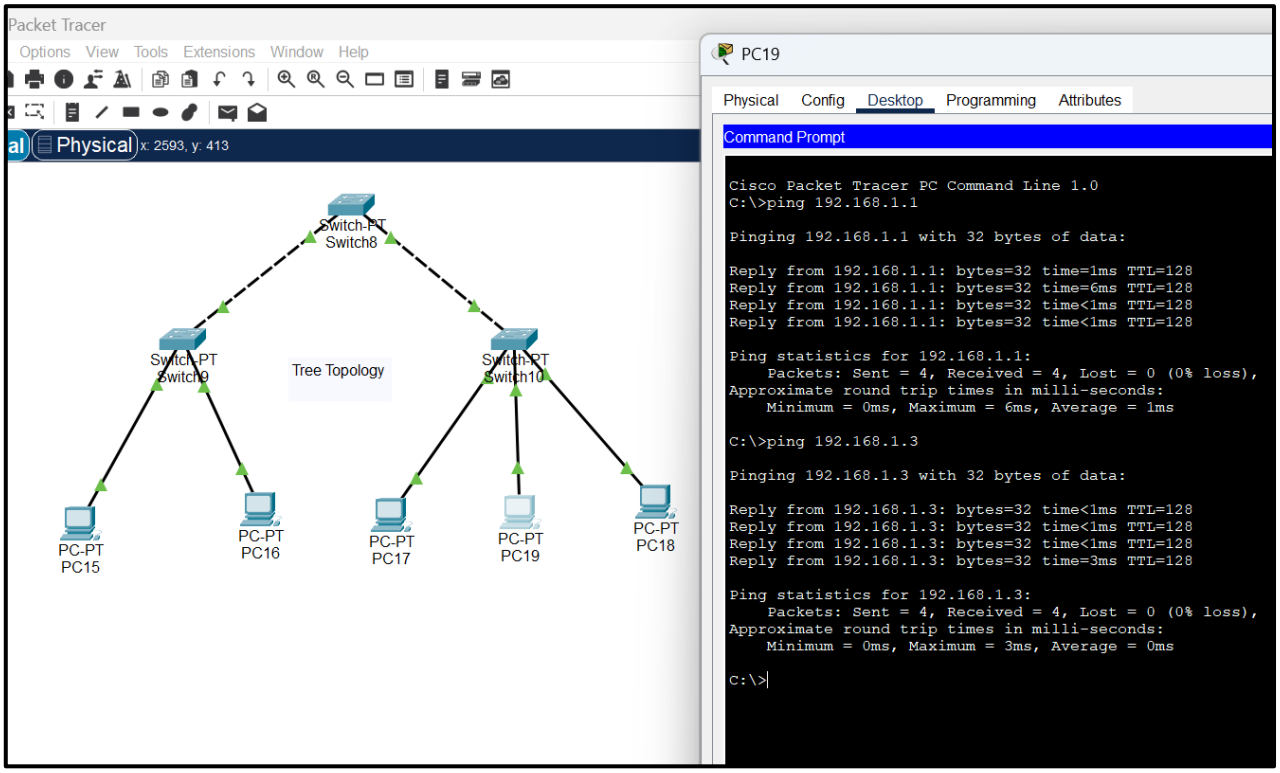


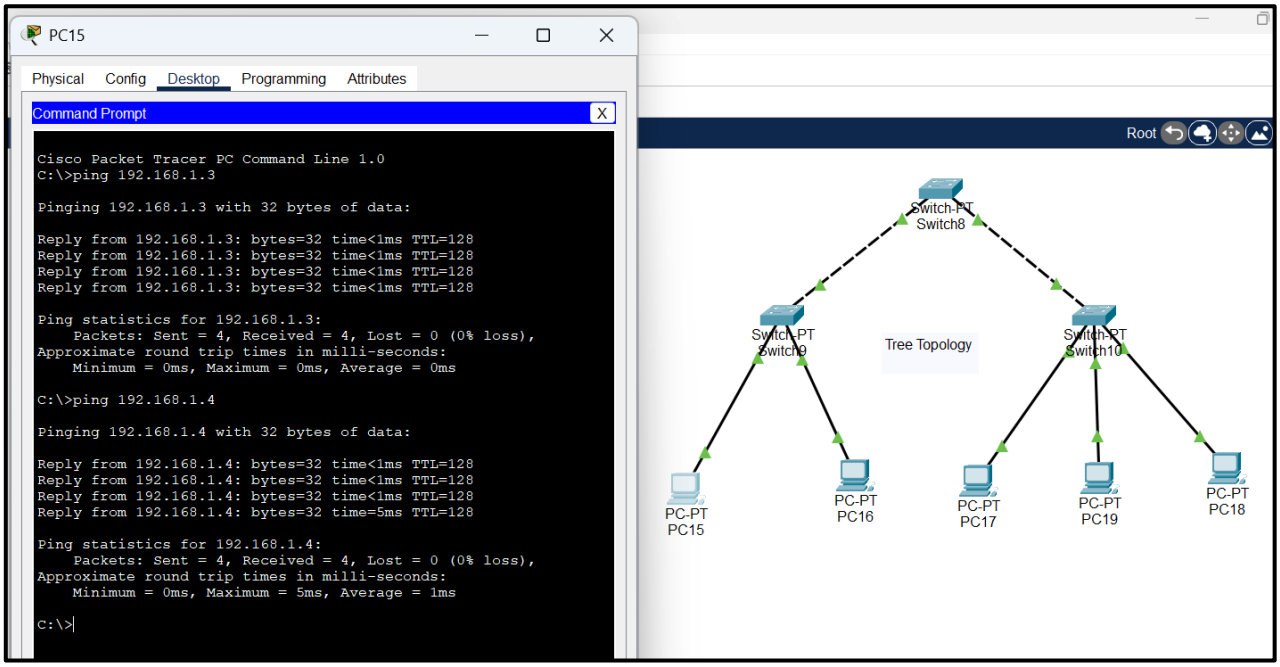
Step 2: Configure IP addresses and subnet masks on all PCs.





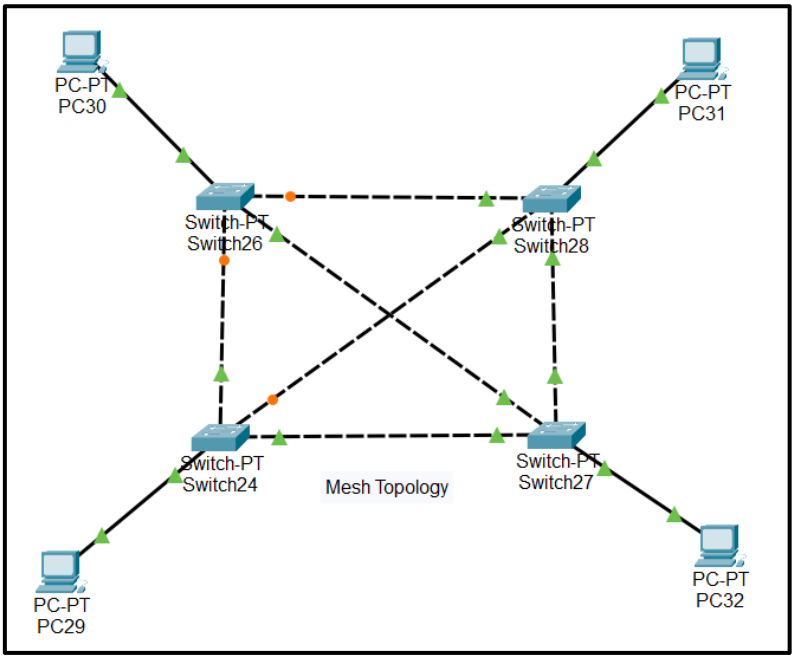
Step 3: Test network connectivity using the PING command between PCs.



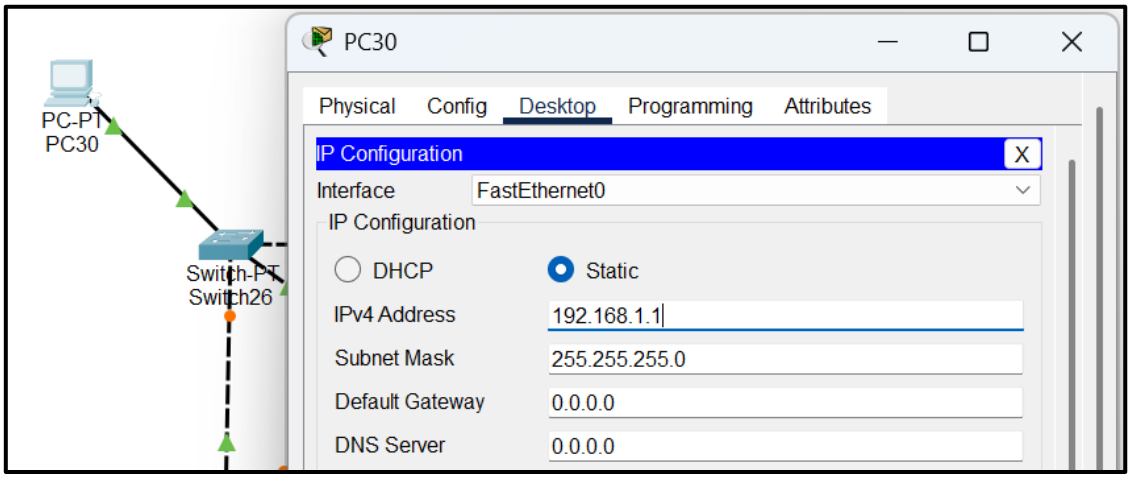


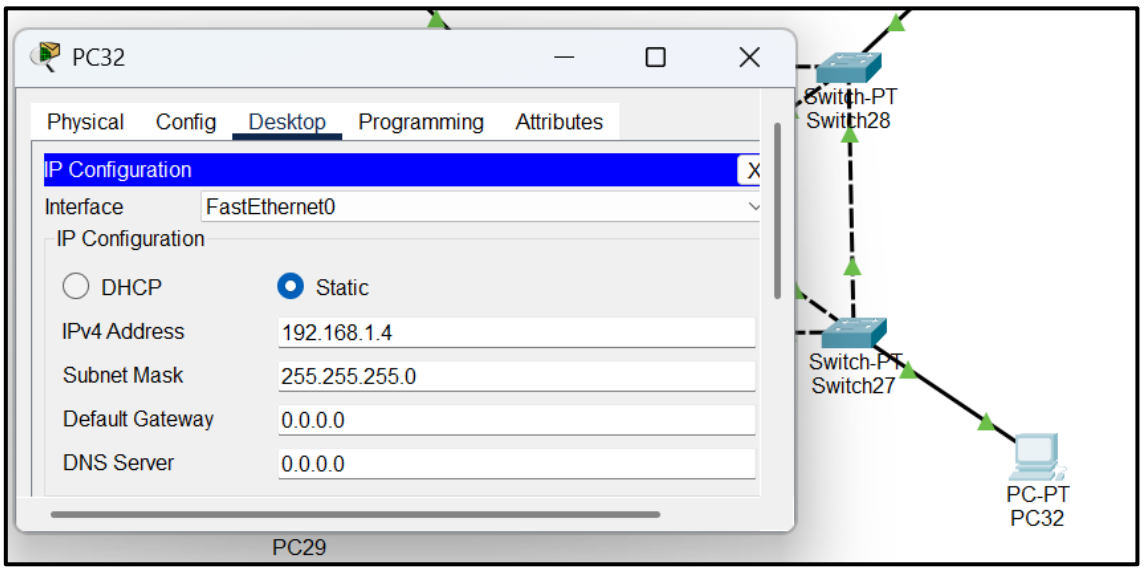
Mesh Topology :

Step 1: Build Mess topology.

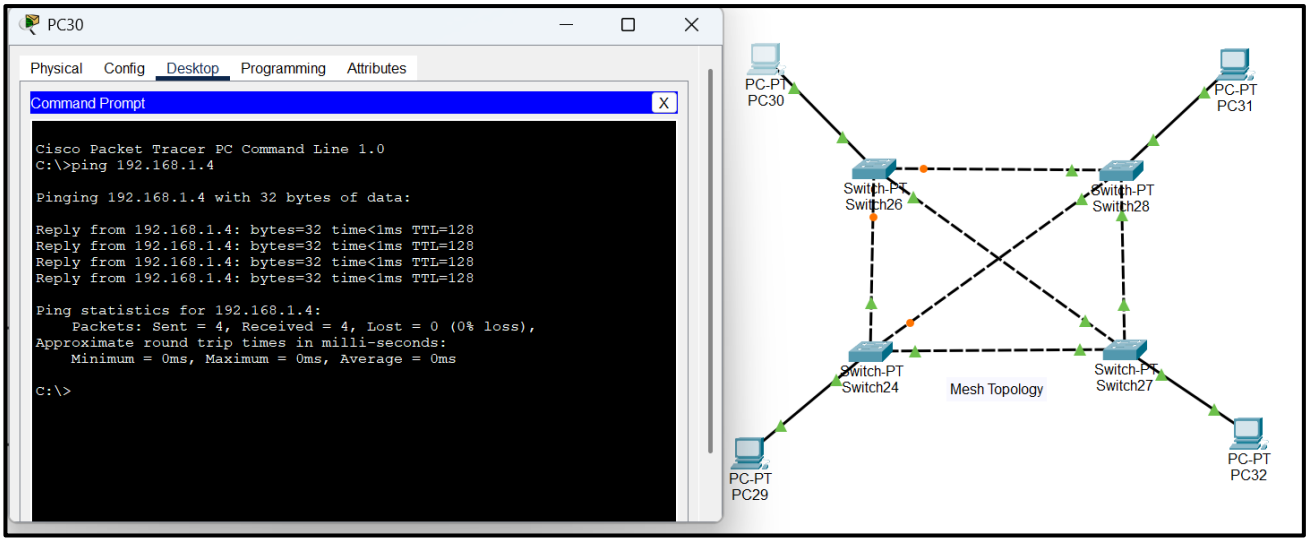


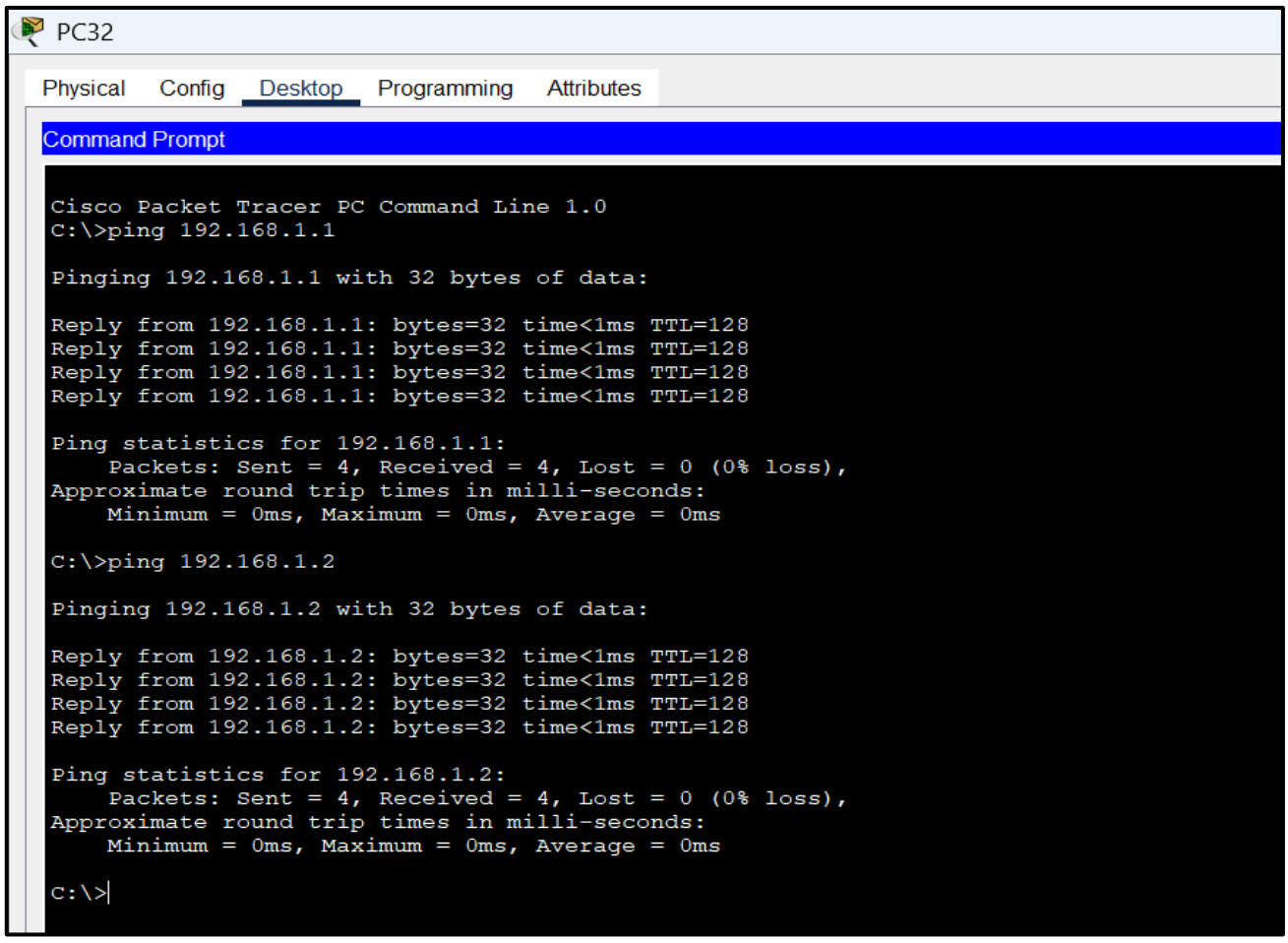
Step 2: Configure IP addresses and subnet masks on all PCs.





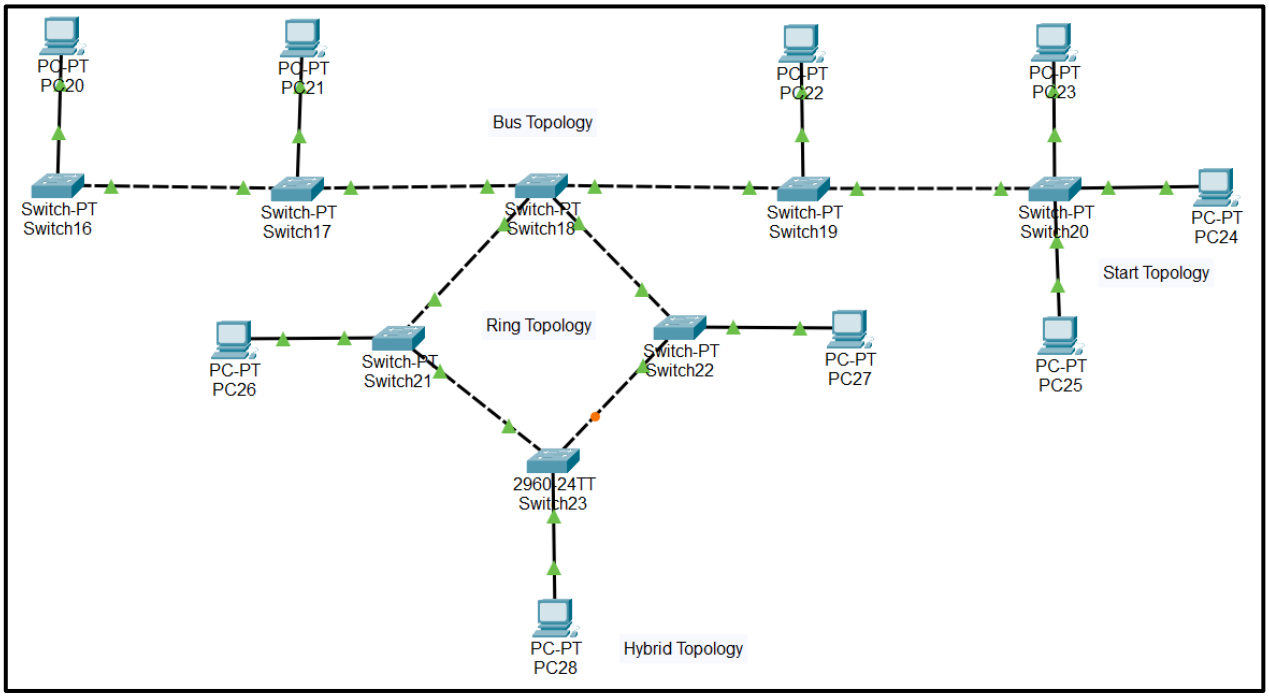
Step 3: Test network connectivity using the PING command between PCs.



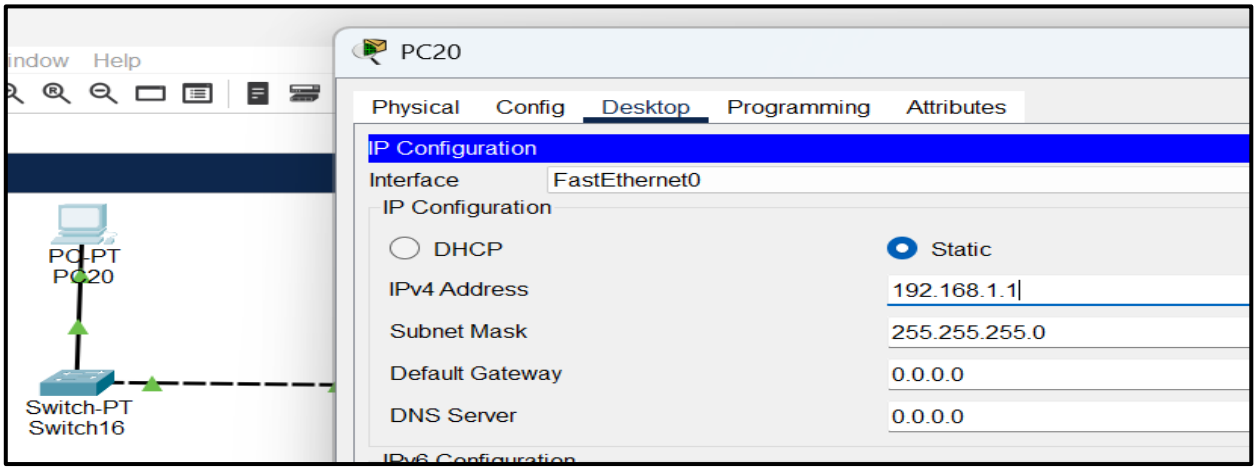


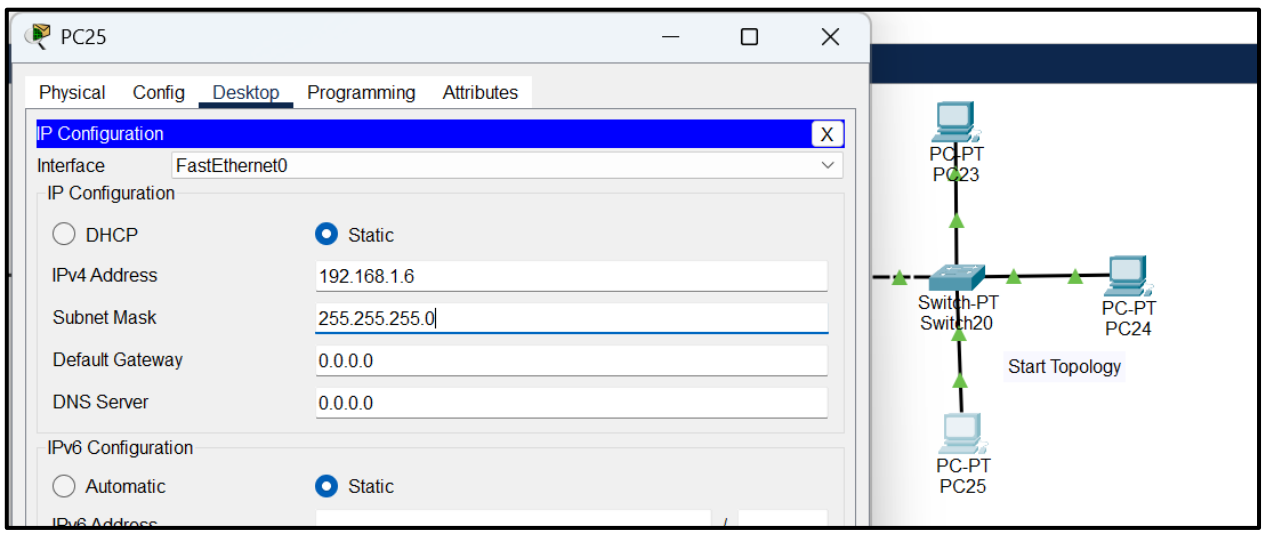
Hybrid Topology :

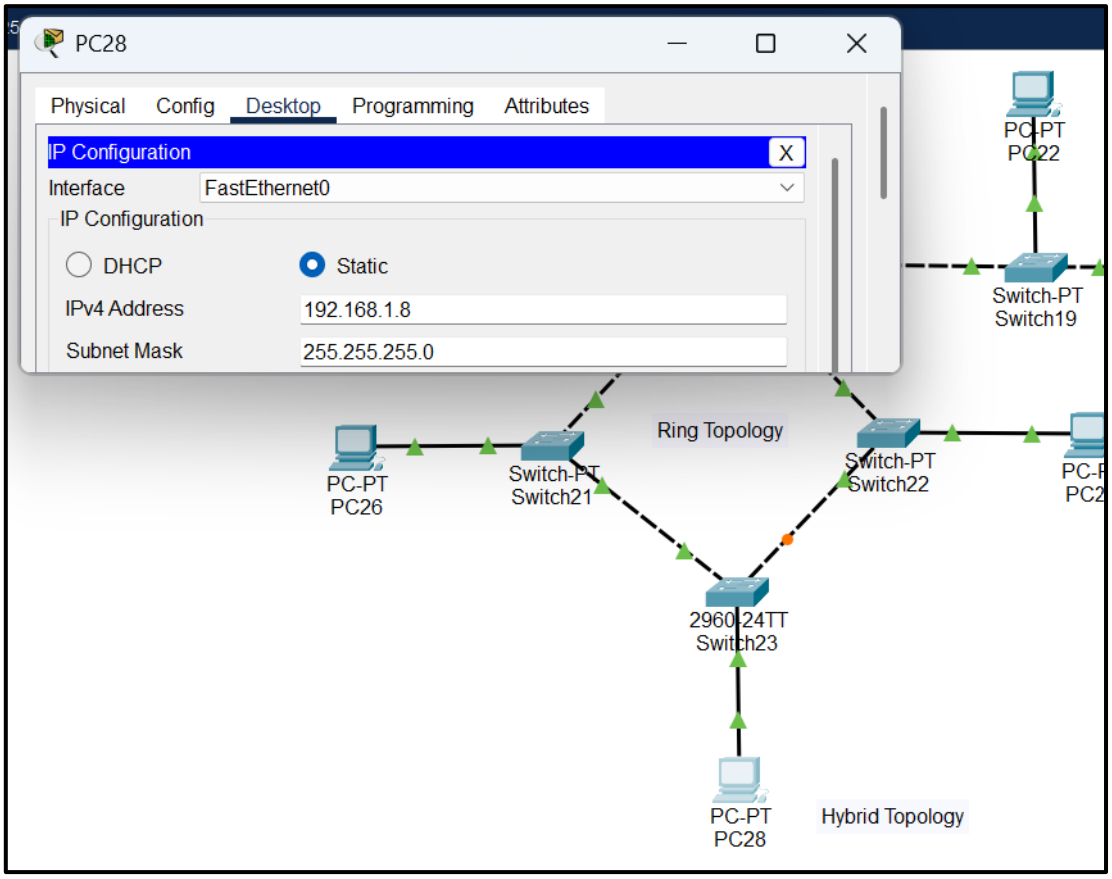
Step 1: Build Hybrid topology.



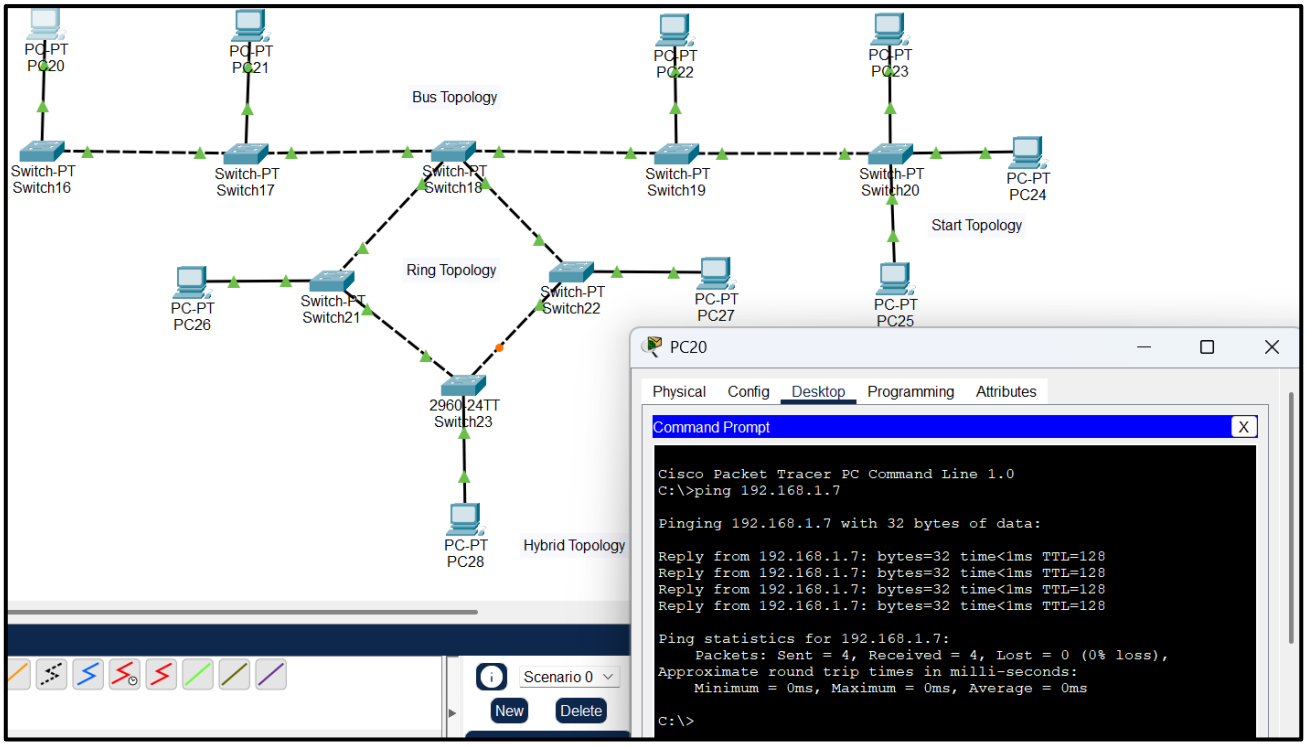
Step 2: Configure IP addresses and subnet masks on all PCs.







Step 3: Test network connectivity using the PING command between PCs.



****

Experiment No : 02

,, ,, ,,

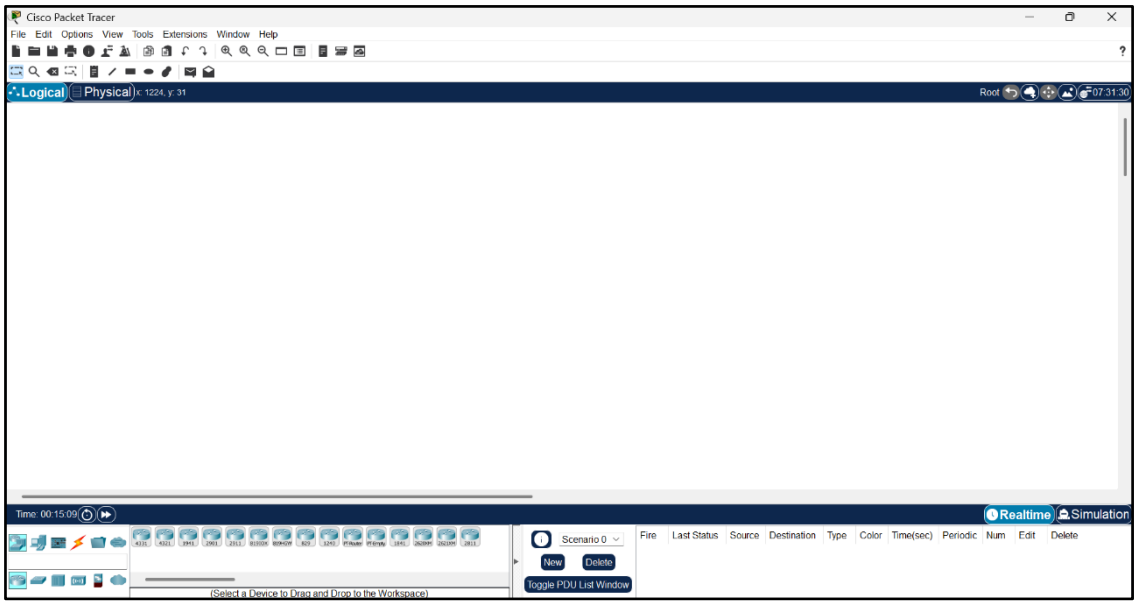
Roll No : TE-43

Name : Sanika Nilesh Patil

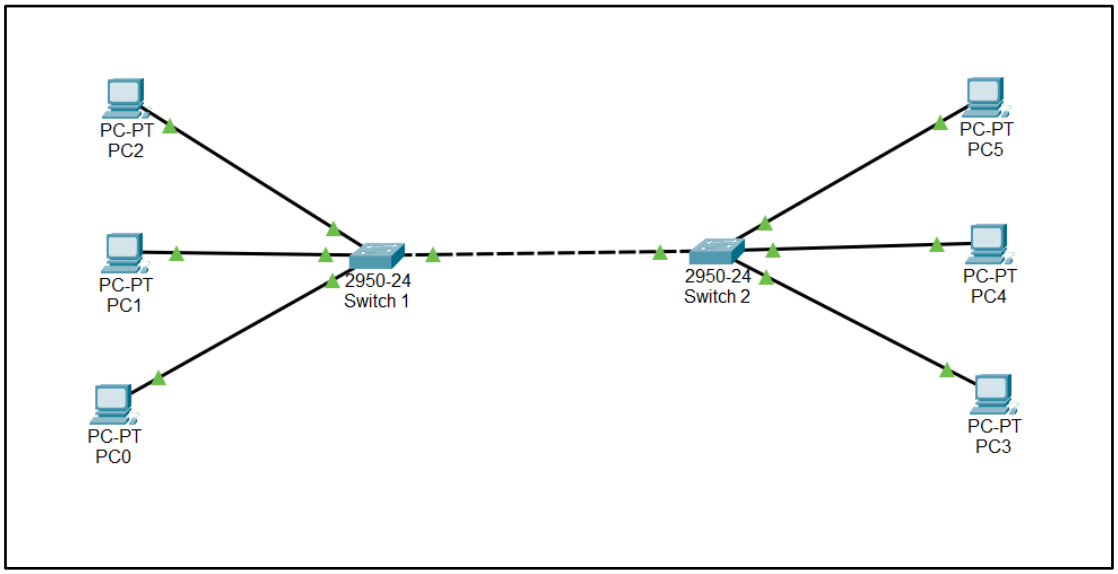
Title : Setup a wired LAN using Layer 2 Switch. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and demonstrating the PING packets captured traces using Wireshark Packet Analyzer Tool.

,, ,, ,,

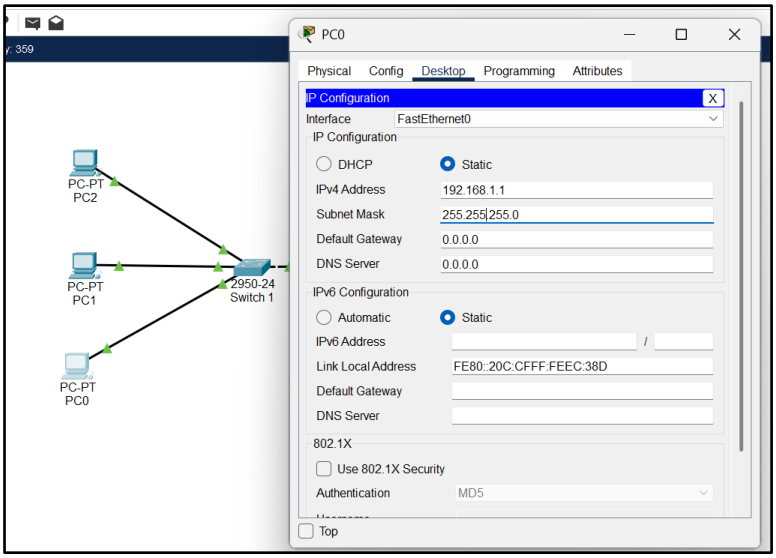
Step 1: Launch Packet Tracer.

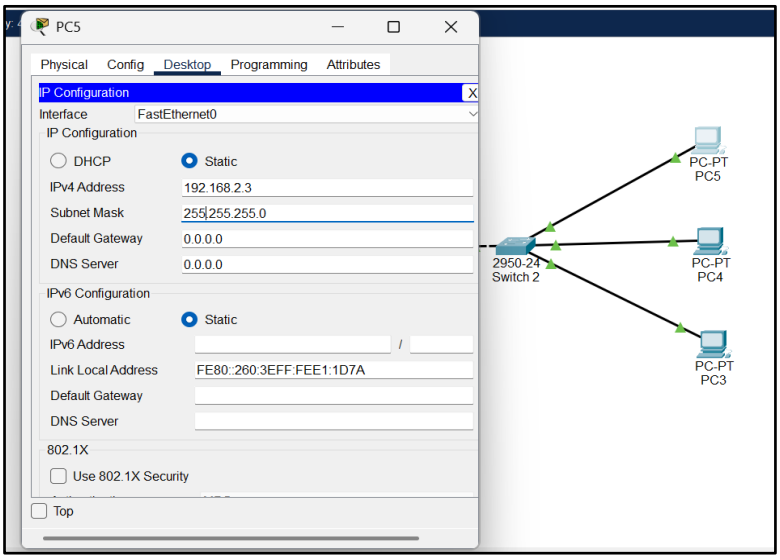


Step 2: Build the topology.

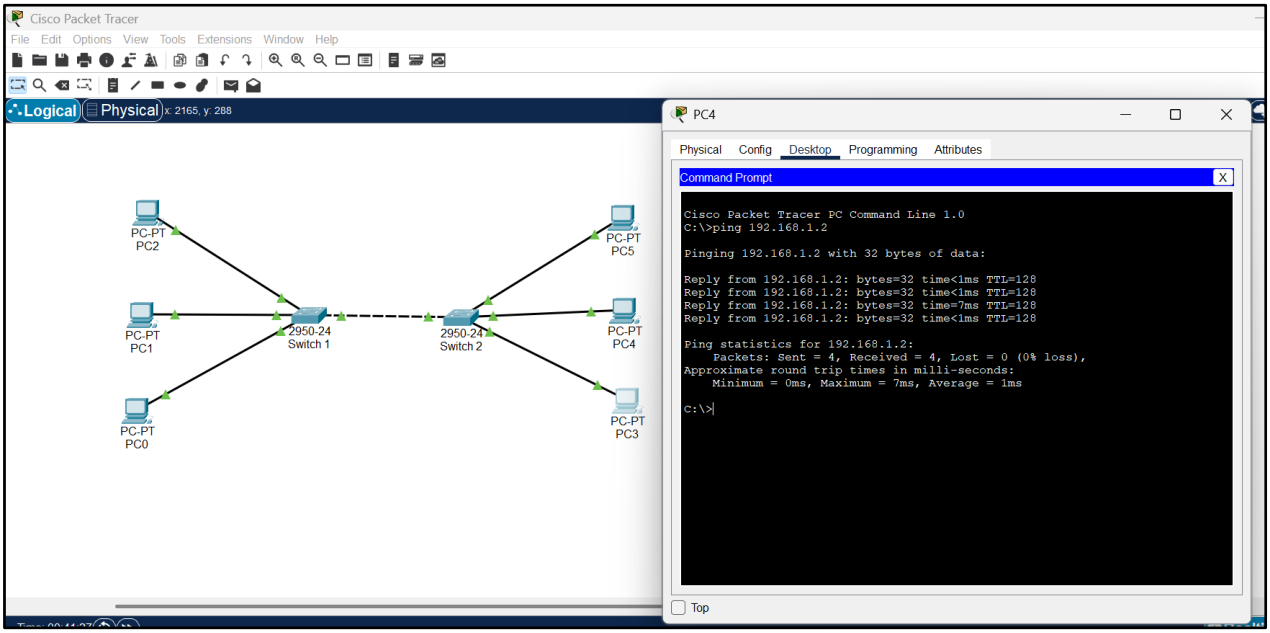


Step 3: Configure IP addresses and subnet masks on all PCs.

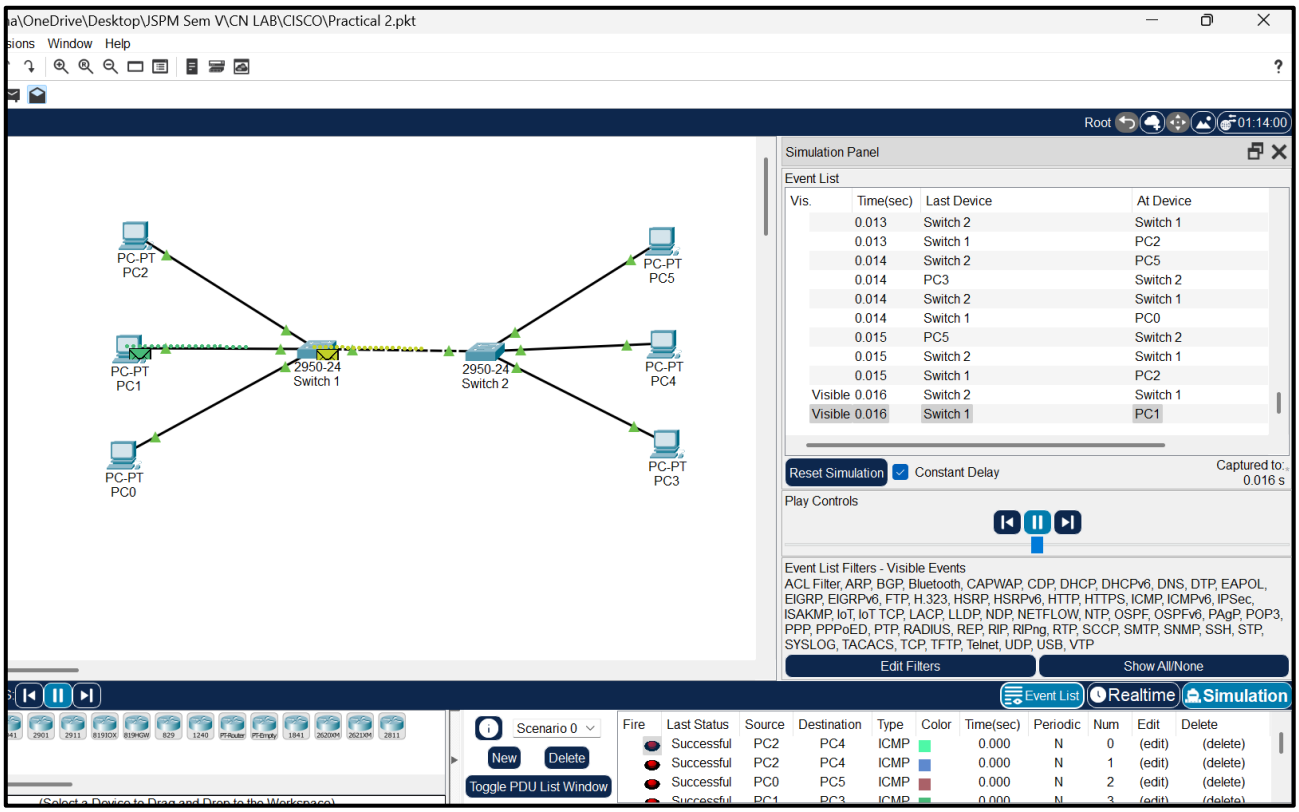




Step 4: Test network connectivity using the PING command between PCs.



Step 5: Capture and analyze the PING packets using Simulation Mode in Cisco Packet Tracer to demonstrate packet flow between the PCs.



****

Experiment No : 03

,, ,, ,,

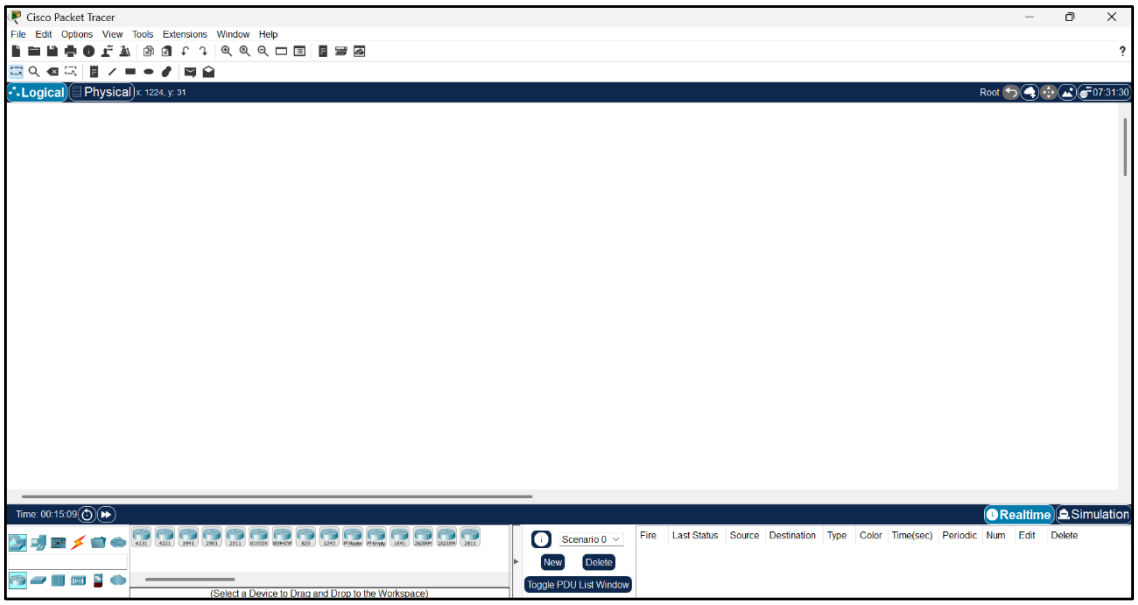
Roll No : TE-43

Name : Sanika Nilesh Patil

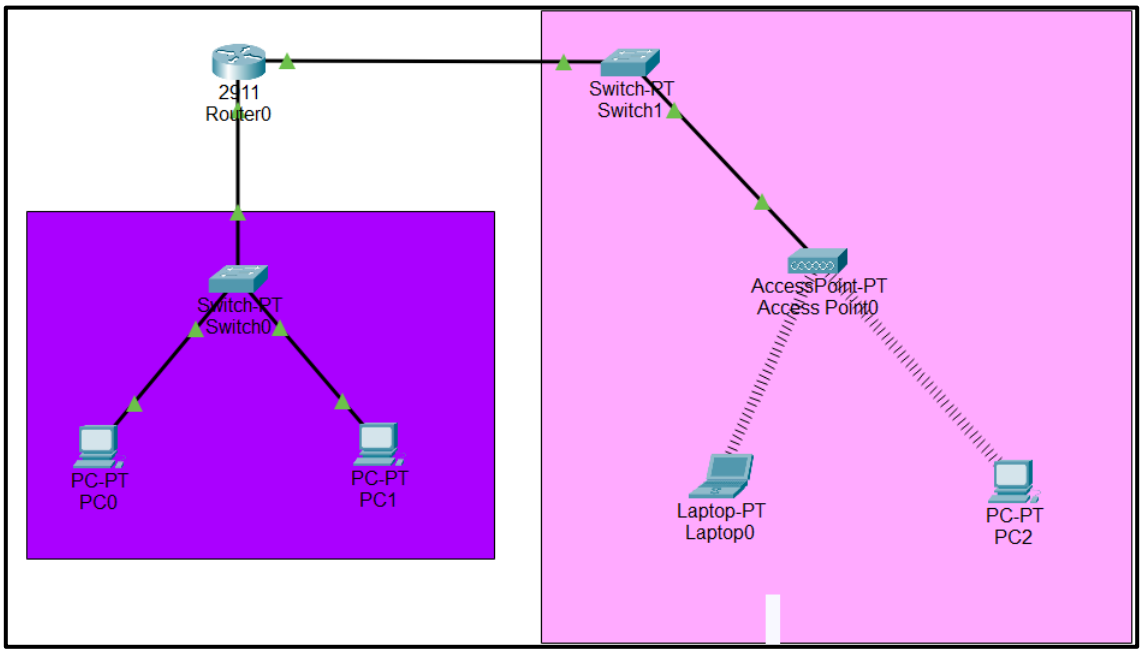
Title : Setup a WAN which contains wired as well as wireless LAN by using a packet tracer tool. Demonstrate transfer of a packet from LAN 1 (wired LAN) to LAN2 (Wireless LAN).

,, ,, ,,

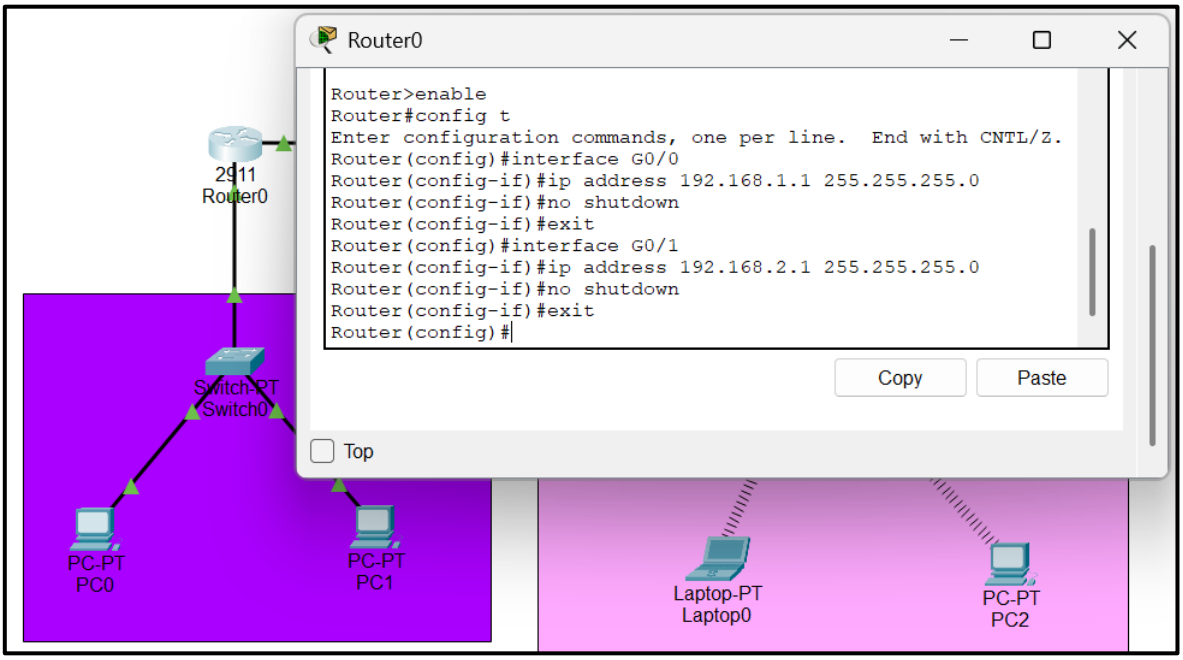
Step 1: Launch Packet Tracer.



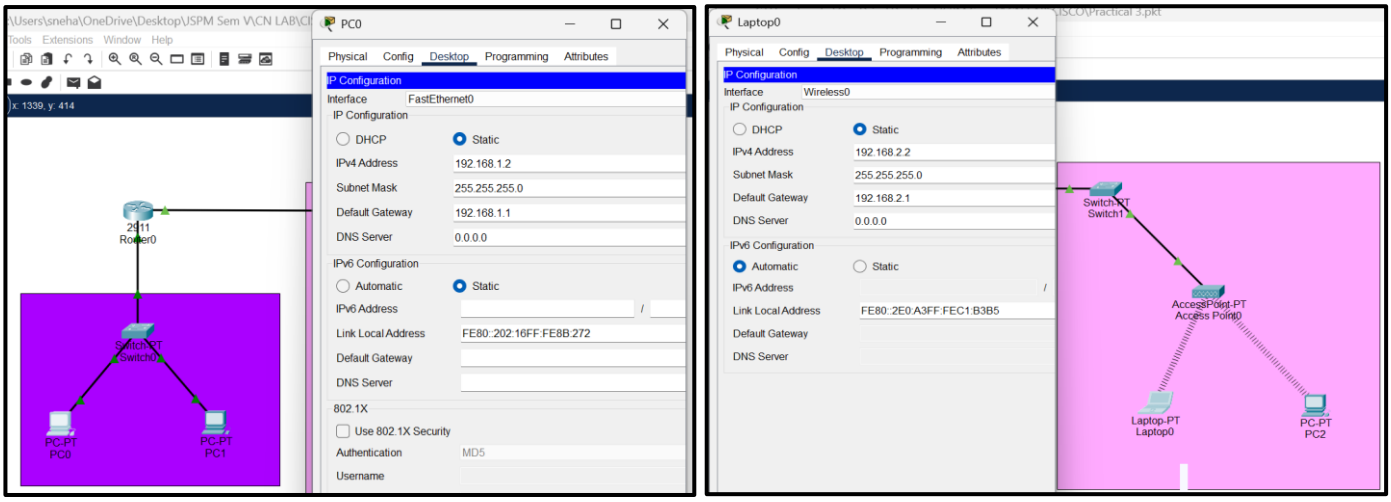
Step 2: Build the topology.



Step 3: Configure router.

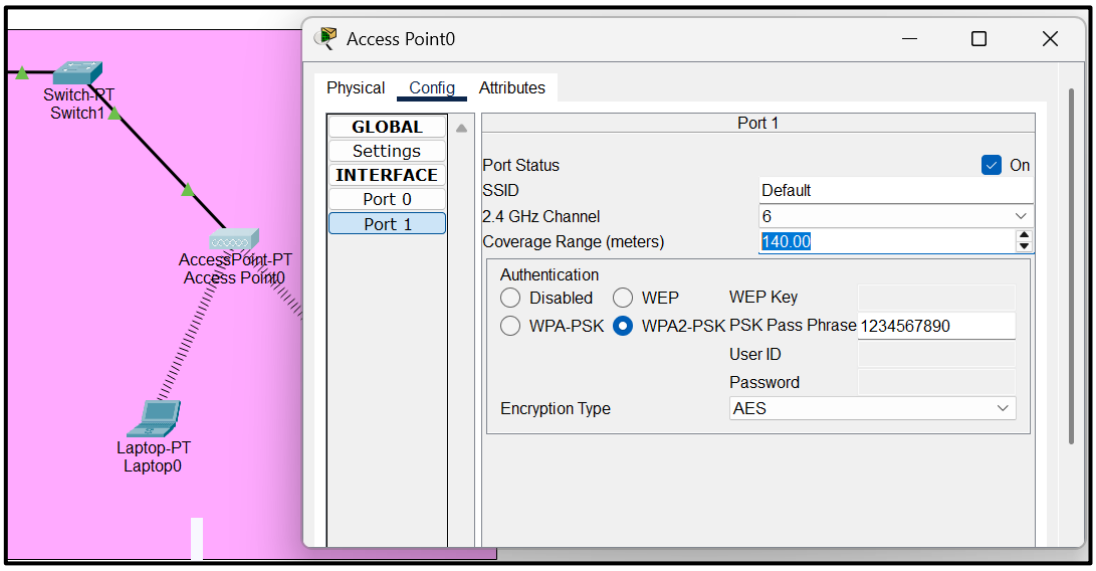


Step 4: Configure IP addresses and subnet masks on all PCs.

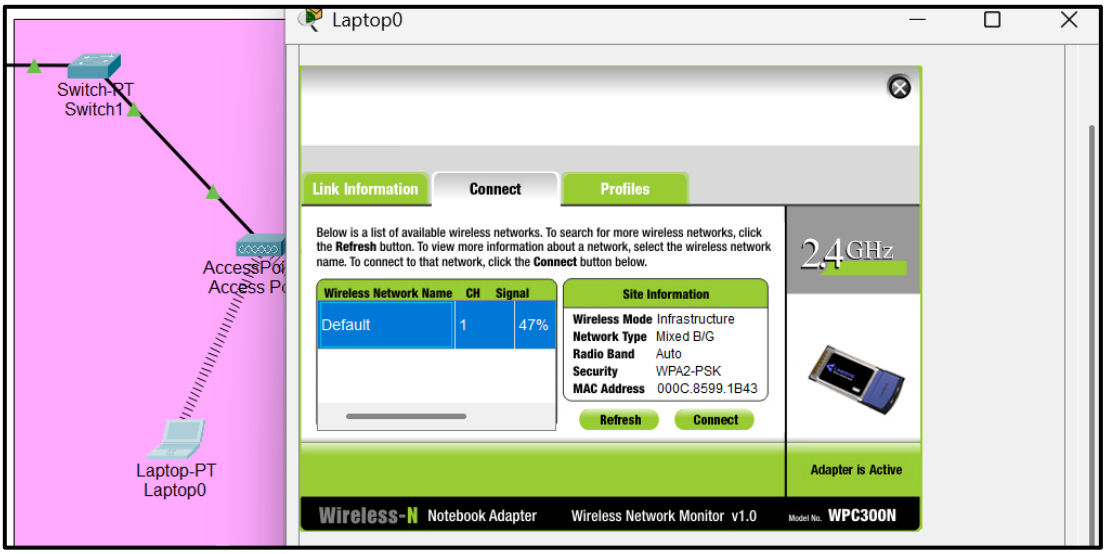


Step 5: Configure the wireless network.

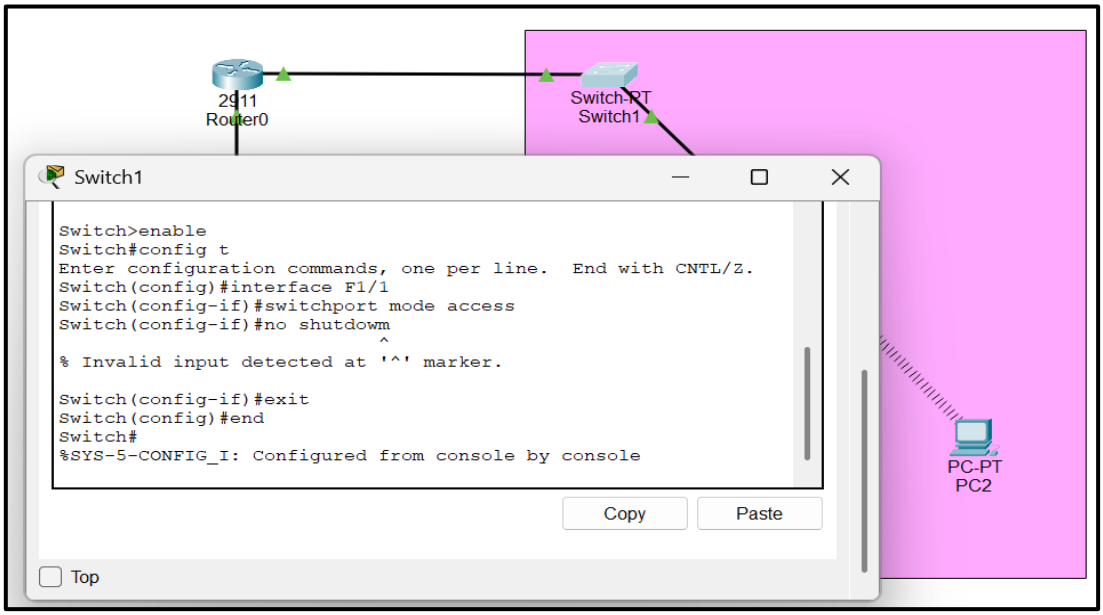
1. Click the Access Point → Config Tab → Port 1
2. Set SSID (network name) → Default
3. Set Wi-Fi password → 1234567890



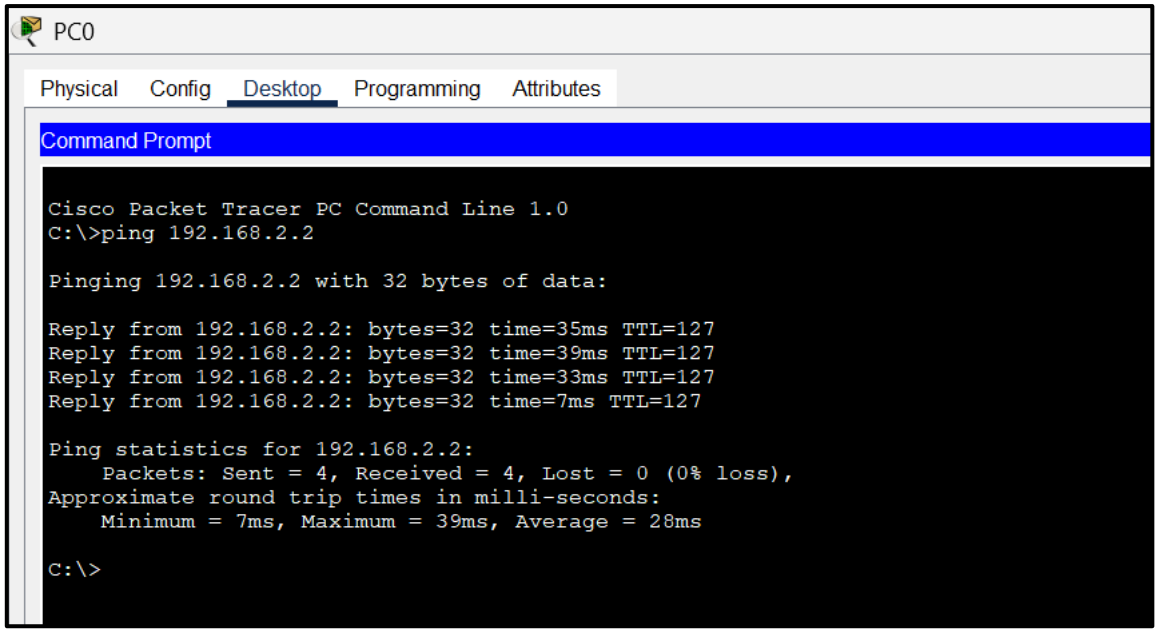
Step 6: Connect Wireless laptop and PC.

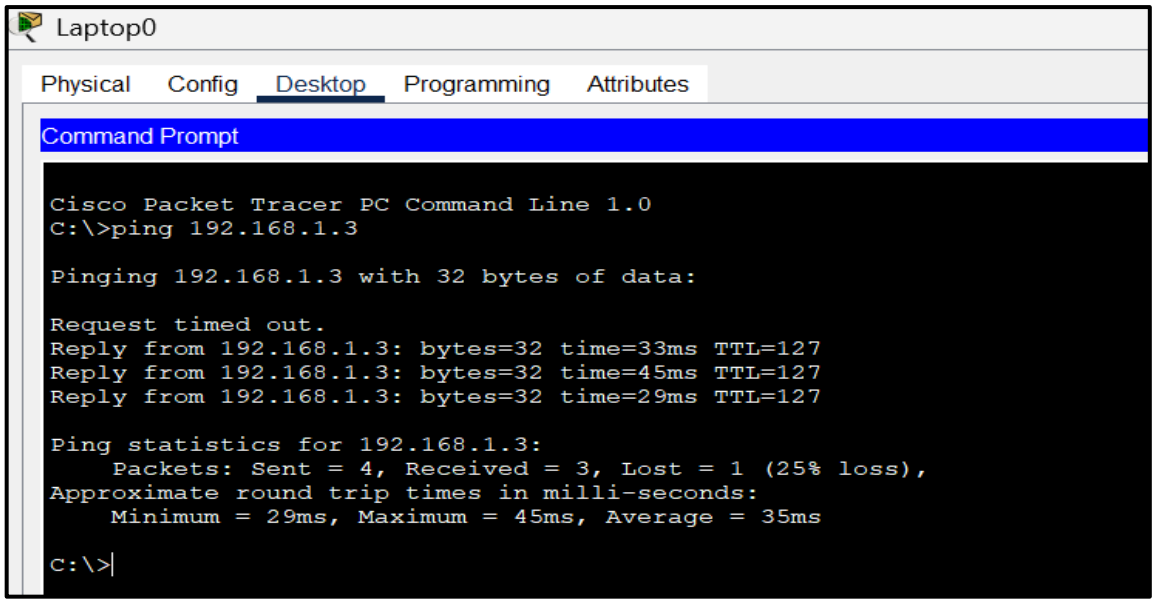


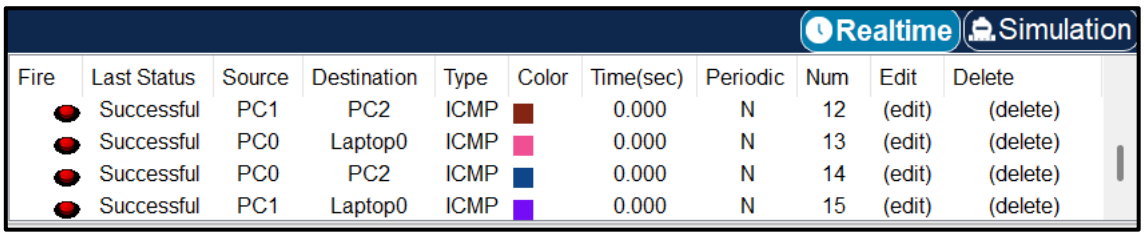
Step 7: Configure switch (Wireless network) for wireless connection .



Step 8: Verify network connectivity between wired and wireless network.







Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	PC2	ICMP		0.000	N	12	(edit)	(delete)
	Successful	PC0	Laptop0	ICMP		0.000	N	13	(edit)	(delete)
	Successful	PC0	PC2	ICMP		0.000	N	14	(edit)	(delete)
	Successful	PC1	Laptop0	ICMP		0.000	N	15	(edit)	(delete)

****

Experiment No : 04

,, ,, ,,

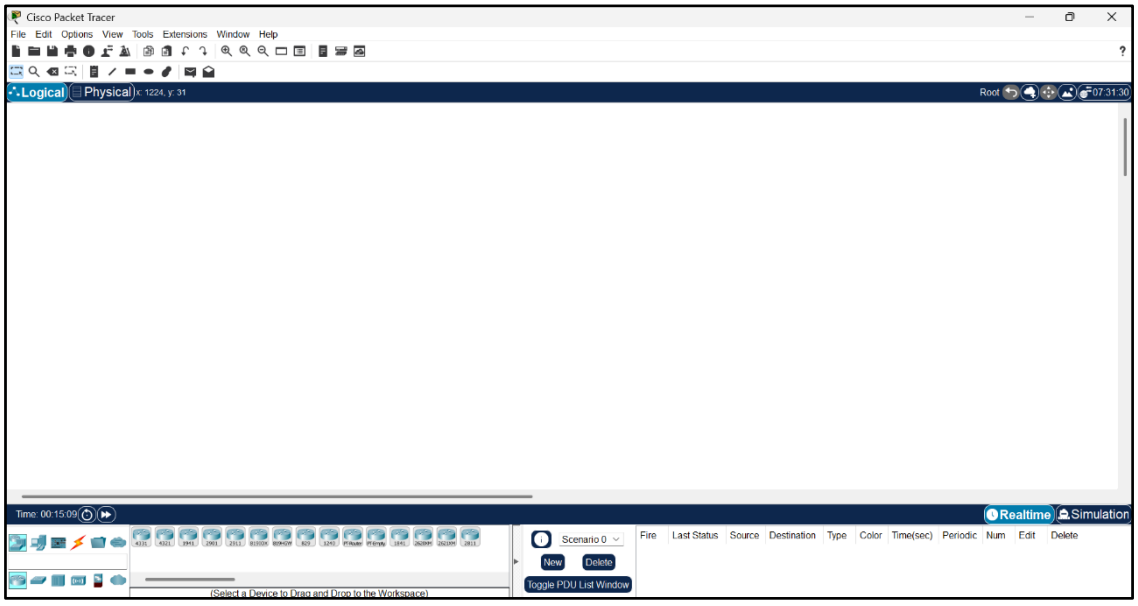
Roll No : TE- 43

Name : Sanika Nilesh Patil

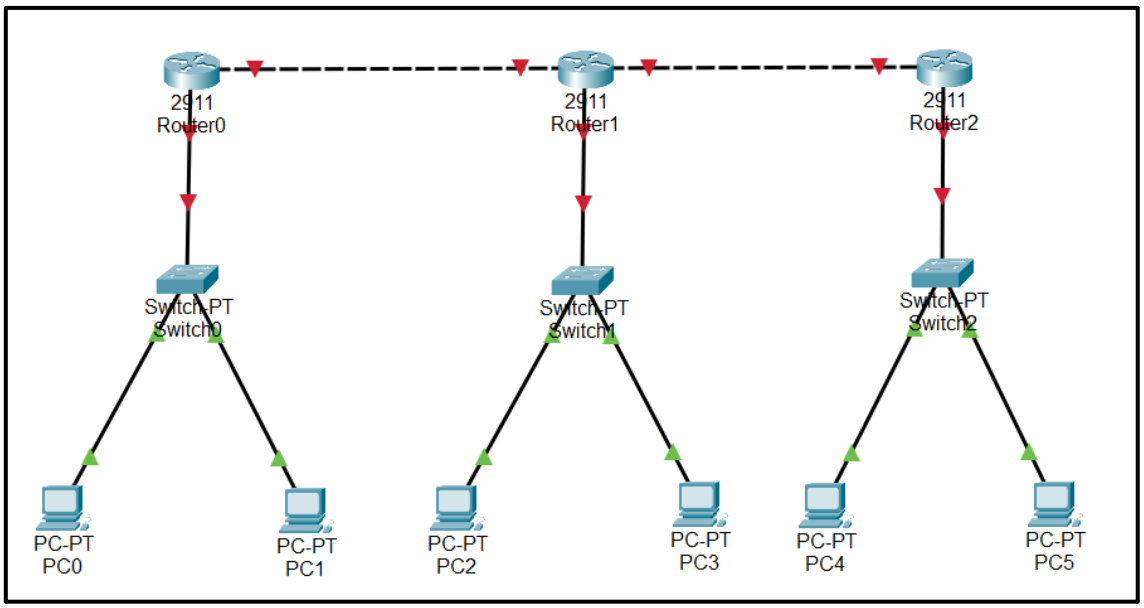
Title : Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF/BGP.

,, ,, ,,

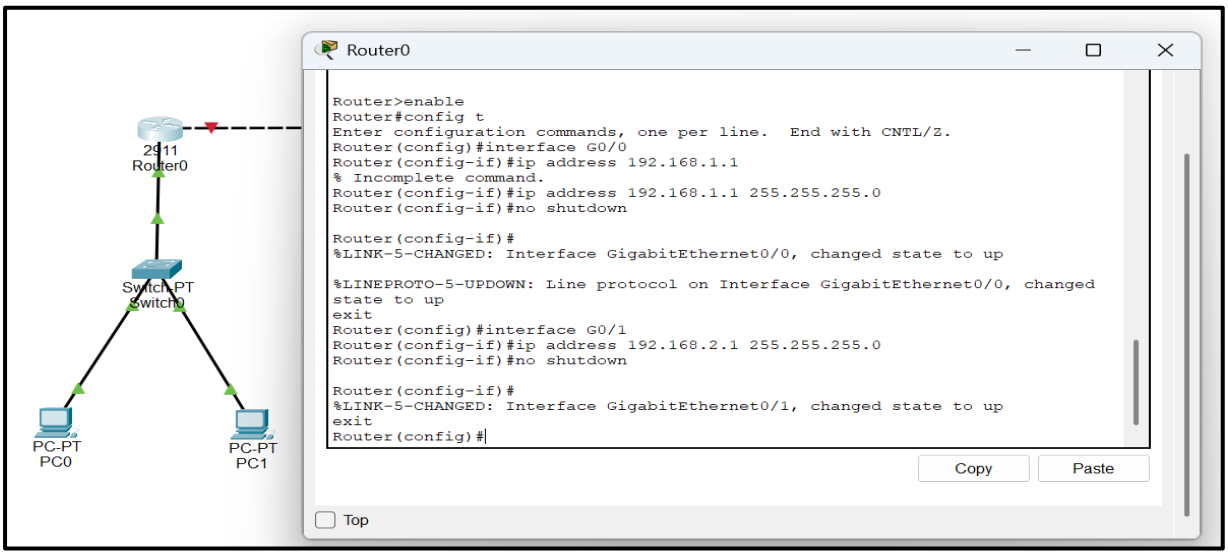
Step 1: Launch Packet Tracer.

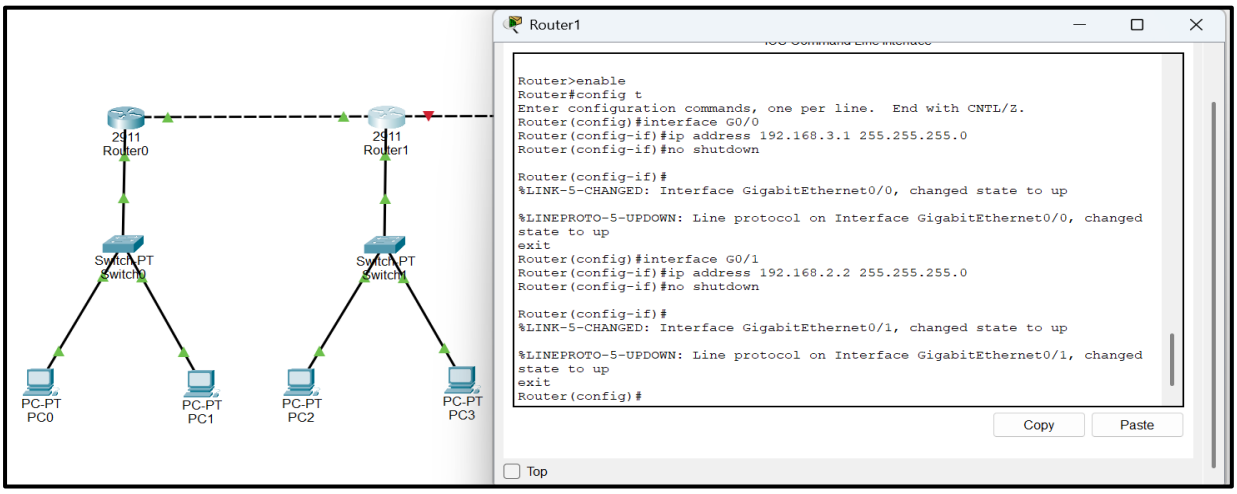


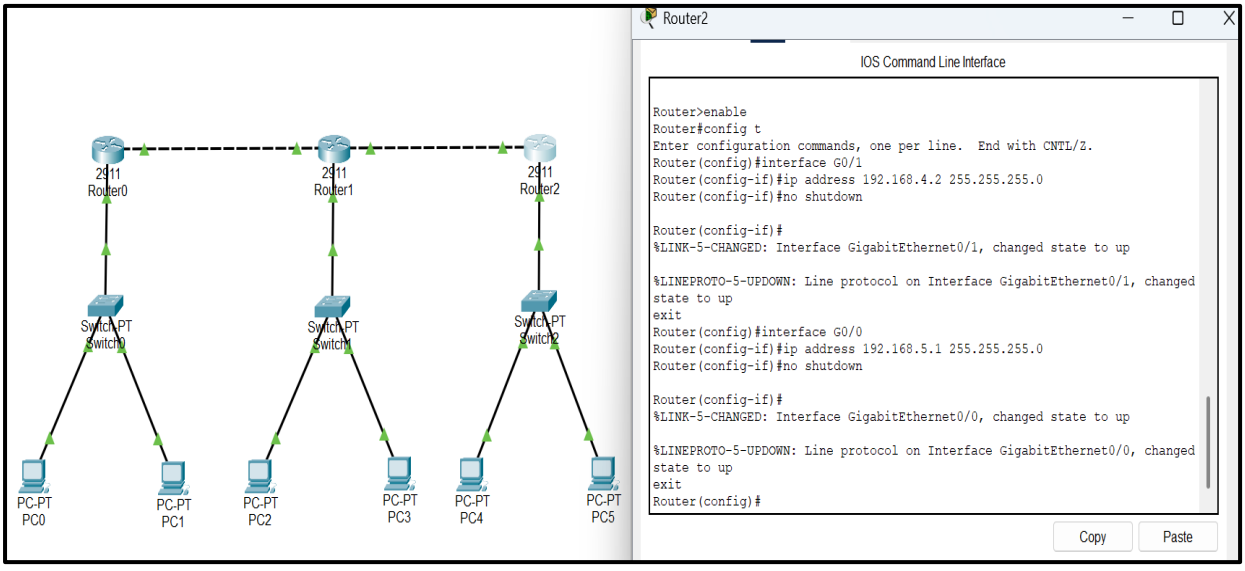
Step 2: Build the topology.



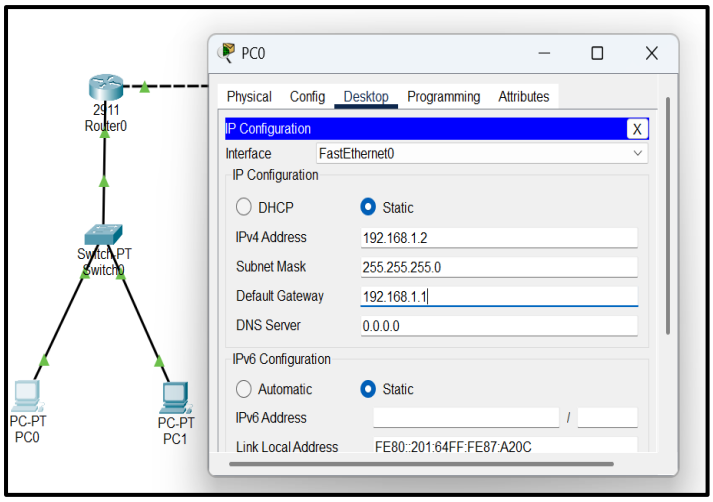
Step 3: Configure routers.

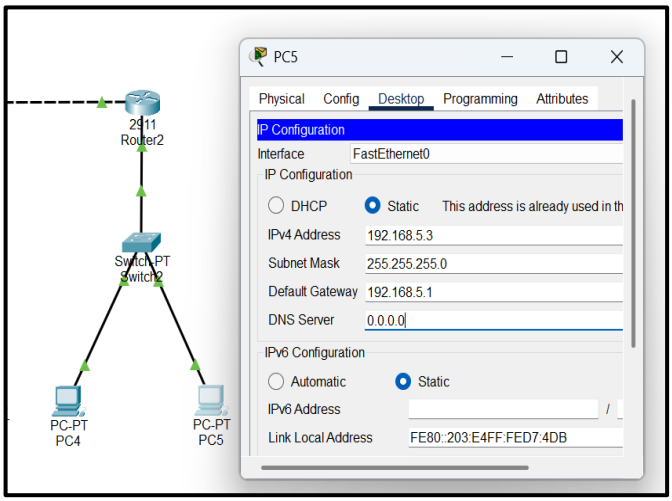




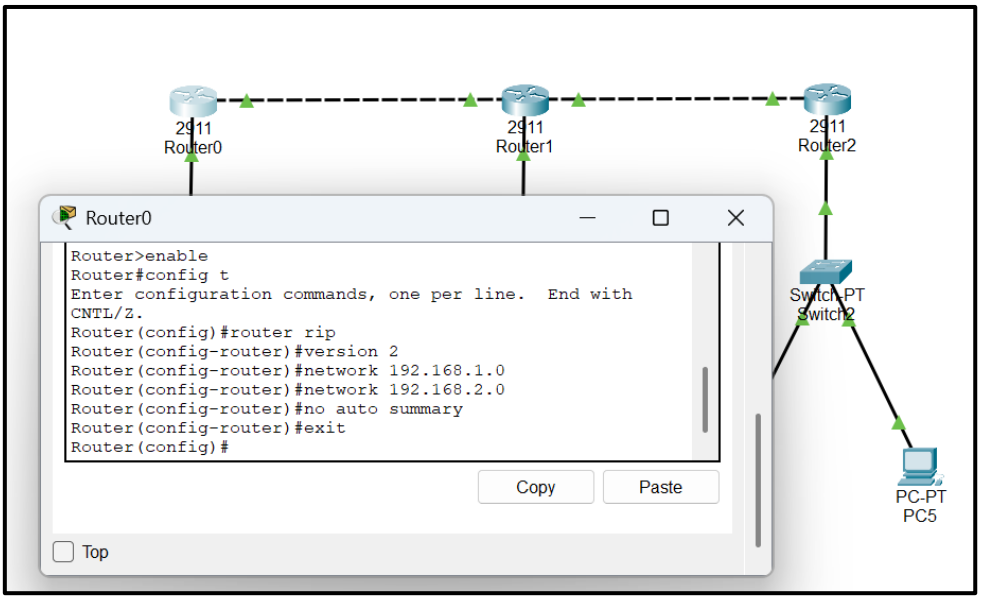


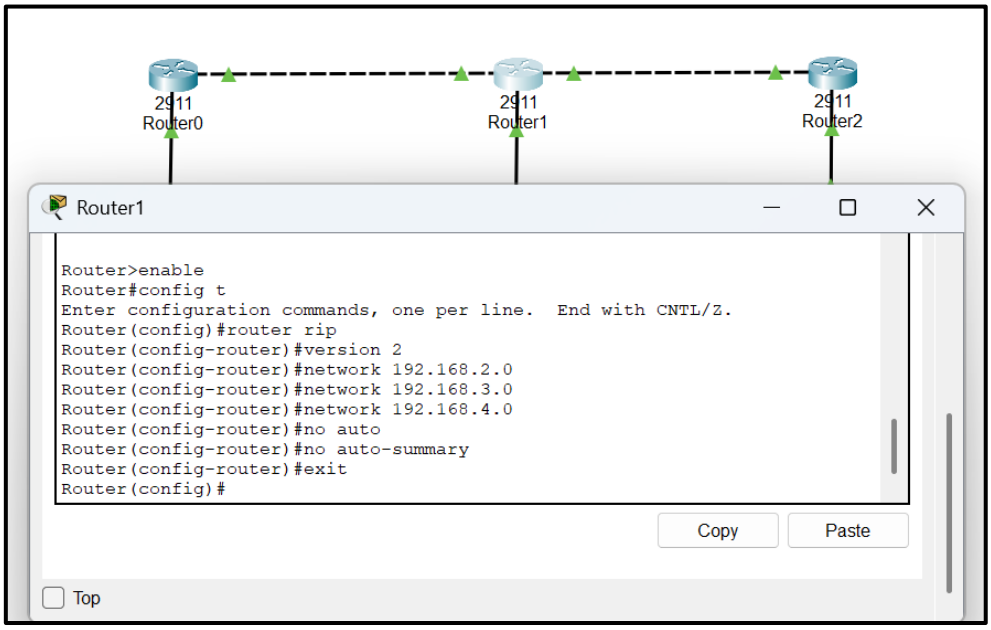
Step 4: Configure IP addresses and subnet masks and Default gateway on all PCs.



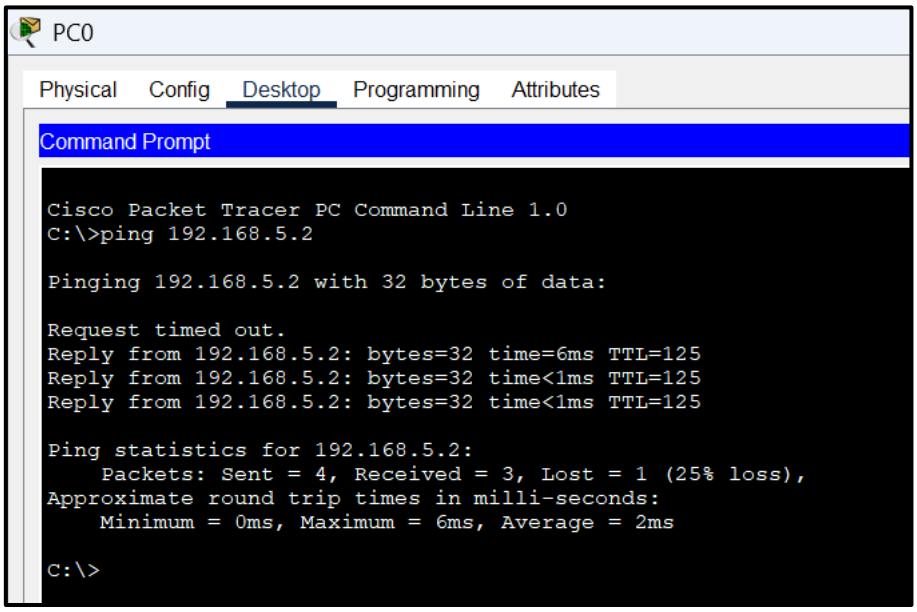


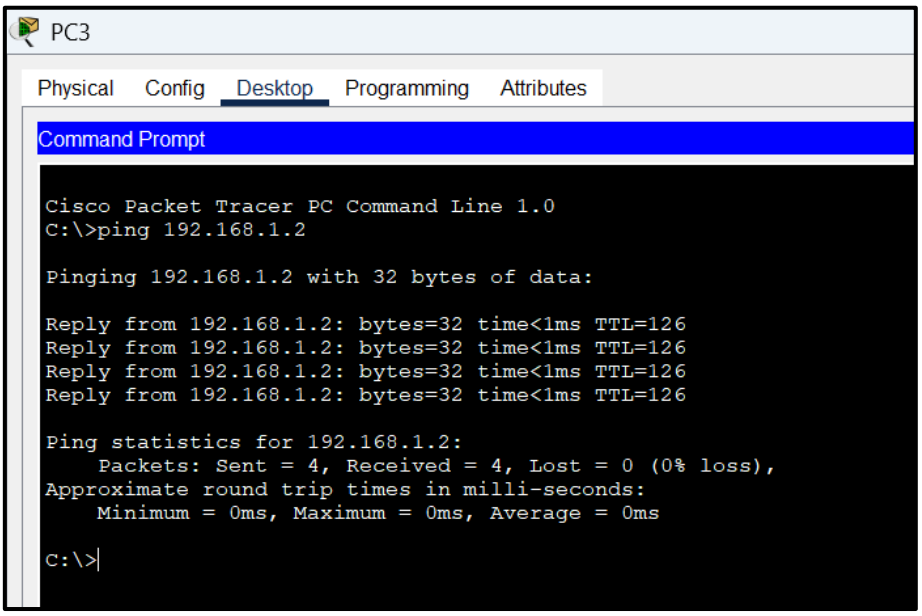
Step 5: Configuration of routers using **RIP** protocol.

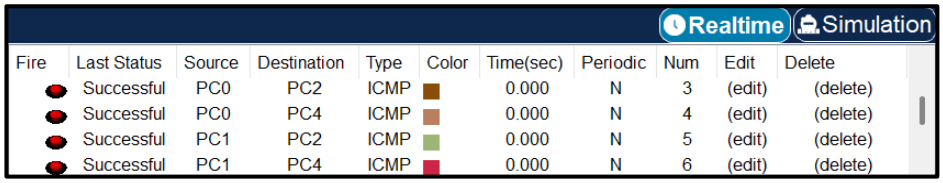




Step 6: Verify Connectivity (Use ping command from PC0, PC1 to PC3, PC4, etc).







Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC2	ICMP		0.000	N	3	(edit)	(delete)
	Successful	PC0	PC4	ICMP		0.000	N	4	(edit)	(delete)
	Successful	PC1	PC2	ICMP		0.000	N	5	(edit)	(delete)
	Successful	PC1	PC4	ICMP		0.000	N	6	(edit)	(delete)

****

Experiment No : 07

”” ”” ””

Roll No : TE-43

Name : Sanika Nilesh Patil

Title : Implement Socket Programming using Python:

a) TCP Client and TCP Server

b) UDP Client and UDP Server

”” ”” ””

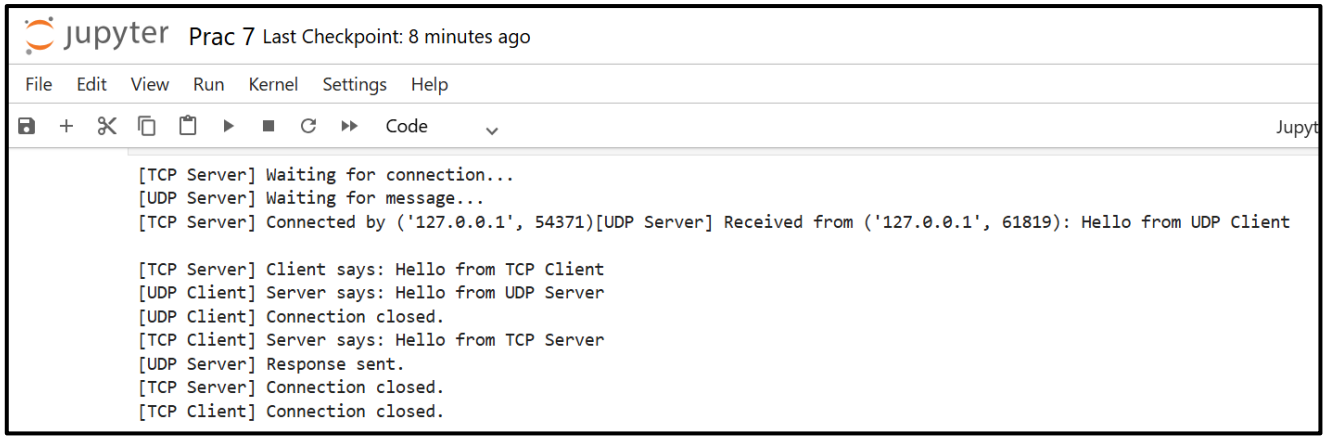
Source Code :

```
import socket
import threading
import time

# ------------------
# TCP Server
# ------------------
def tcp_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('localhost', 12345))
    server_socket.listen(1)
    print("[TCP Server] Waiting for connection...")

    conn, addr = server_socket.accept()
    print(f"[TCP Server] Connected by {addr}")
    data = conn.recv(1024).decode()
    print(f"[TCP Server] Client says: {data}")
    conn.sendall("Hello from TCP Server".encode())
    conn.close()
    print("[TCP Server] Connection closed.")

# ------------------
# TCP Client
# ------------------
```

```
def tcp_client():
    time.sleep(1)  # wait for server to start
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 12345))
    client_socket.sendall("Hello from TCP Client".encode())
    data = client_socket.recv(1024).decode()
    print(f"[TCP Client] Server says: {data}")
    client_socket.close()
    print("[TCP Client] Connection closed.")

# --------------------
# UDP Server
# --------------------
def udp_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    server_socket.bind(('localhost', 12346))  # use different port
    print("[UDP Server] Waiting for message...")
    data, addr = server_socket.recvfrom(1024)
    print(f"[UDP Server] Received from {addr}: {data.decode()}")
    server_socket.sendto("Hello from UDP Server".encode(), addr)
    print("[UDP Server] Response sent.")

# --------------------
# UDP Client
# --------------------
def udp_client():
    time.sleep(1)  # wait for server to start
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    server_address = ('localhost', 12346)
    client_socket.sendto("Hello from UDP Client".encode(), server_address)
    data, addr = client_socket.recvfrom(1024)
    print(f"[UDP Client] Server says: {data.decode()}")
    client_socket.close()
    print("[UDP Client] Connection closed.")
```

```
# ------------------
# Run all using threads
# ------------------
threads = []

# TCP
threads.append(threading.Thread(target=tcp_server))
threads.append(threading.Thread(target=tcp_client))

# UDP
threads.append(threading.Thread(target=udp_server))
threads.append(threading.Thread(target=udp_client))

# Start all threads
for t in threads:
    t.start()
# Wait for all threads to finish
for t in threads:
    t.join()
```

Output :



****

Experiment No : 08

,, ,, ,,

Roll No : TE-43

Name : Sanika Nilesh Patil

Title : Write a program using TCP socket for wired network for following

1. Say Hello to each other 2. File transfer 3. Calculator.

,, ,, ,,

- **Source Code :**

❖ **Server Side Program :**

```
import socket

# Step 1: Create socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Step 2: Bind server to host and port
server_host = '0.0.0.0'   # Listen on all interfaces
server_port = 5001
server_socket.bind((server_host, server_port))
print(f"Server listening on {server_host}:{server_port}")

# Step 3: Listen and accept connection
server_socket.listen(1)
conn, addr = server_socket.accept()
print(f"Connected by {addr}")

# Step 4: Exchange hello messages
conn.sendall(b"Hello Client")
client_msg = conn.recv(1024).decode()
print("Client says:", client_msg)

# Step 5: Receive file
filename = "received_file.txt"
```

```
with open(filename, "wb") as f:
    print("Receiving file...")
    while True:
        data = conn.recv(1024)
        if not data:
            break
        f.write(data)

print(f"File '{filename}' received successfully.")

# Step 6: Close connection
conn.close()
server_socket.close()
print("Server closed.")
```

❖ **Client Side Program :**

```
import socket

# Step 1: Create socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Step 2: Connect to the server
server_ip = '127.0.0.1'  # Use 'localhost' if on same machine, else server's IP
server_port = 5001
client_socket.connect((server_ip, server_port))
print("Connected to server.")

# Step 3: Exchange hello messages
server_msg = client_socket.recv(1024).decode()
print("Server says:", server_msg)
client_socket.sendall(b"Hello Server")

# Step 4: Send a file
file_to_send = "sample.txt"
with open(file_to_send, "rb") as f:
```

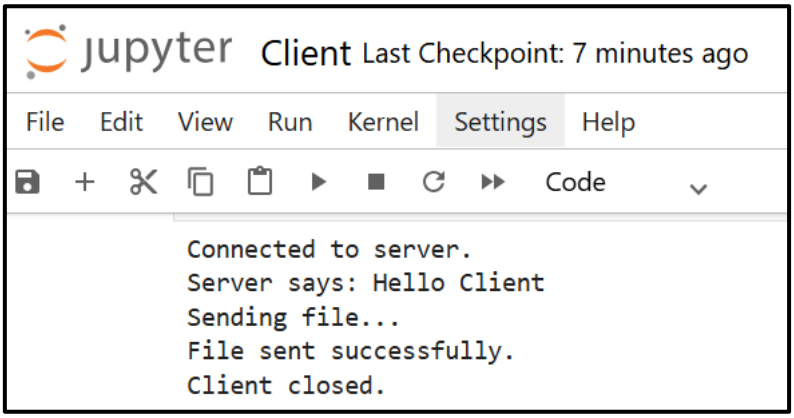
```
    print("Sending file...")
    data = f.read(1024)
    while data:
        client_socket.sendall(data)
        data = f.read(1024)

print("File sent successfully.")

# Step 5: Close connection
client_socket.close()
print("Client closed.")
```

❖ **Sample Text File :**

```
with open("sample.txt", "w") as f:
    f.write("This is a sample file for TCP file transfer test.\nHello from Client!")
print("Sample file created.")
```

Output :

❖ **Client Side :**



❖ **Server Side :**

jupyter Server Last Checkpoint: 8 minutes ago

File Edit View Run Kernel Settings Help

Code

```
print("Server closed.")

Server listening on 0.0.0.0:5001
Connected by ('127.0.0.1', 63974)
Client says: Hello Server
Receiving file...
File 'received_file.txt' received successfully.
Server closed.
```

❖ **Received text file :**

jupyter received_file.txt Last Checkpoint: 8 minutes ago

File Edit View Settings Help

```
1 with open("sample.txt", "w") as f:
2     f.write("This is a sample file for TCP file transfer test.\nHello from Client!")
3 print("Sample file created.")
```

****

Experiment No : 10

,, ,, ,,

Roll No : TE-43

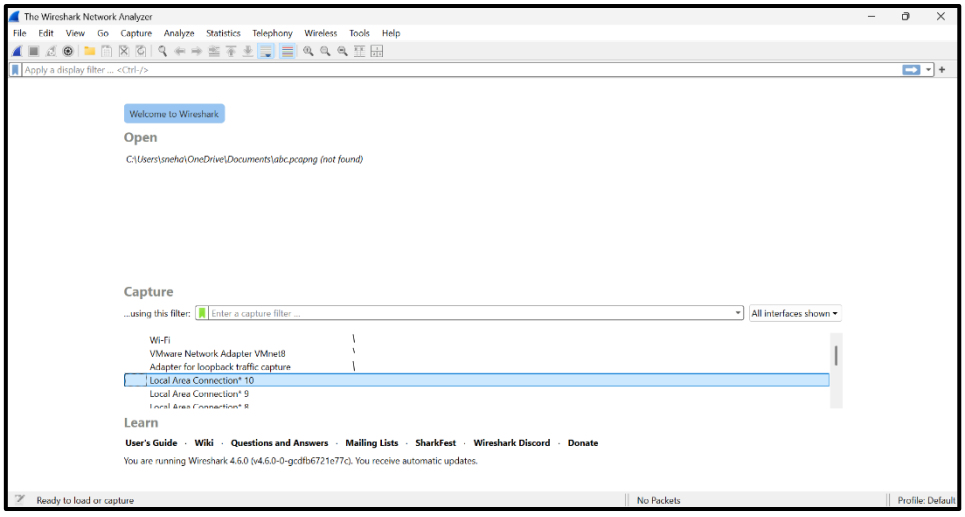
Name : Sanika Nilesh Patil

Title : Capture packets using Wireshark, write the exact packet capture filter expressions to accomplish the following and save the output in file:

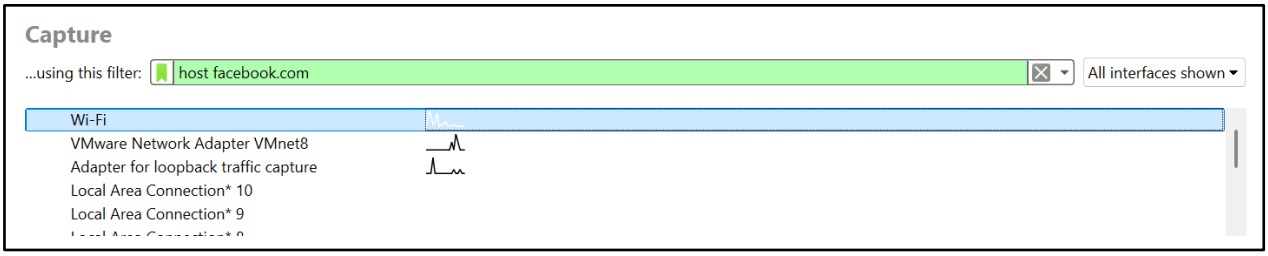
1. Capture all TCP traffic to/from Facebook, during the time when you log in to your Facebook account.
2. Capture all HTTP traffic to/from Facebook, when you log in to your Facebook account
3. Write a DISPLAY filter expression to count all TCP packets (captured under item #1) that have the flags SYN, PSH, and RST set. Show the fraction of packets that had each flag set.
4. Count how many TCP packets you received from / sent to Facebook, and how many of each were also HTTP packets.

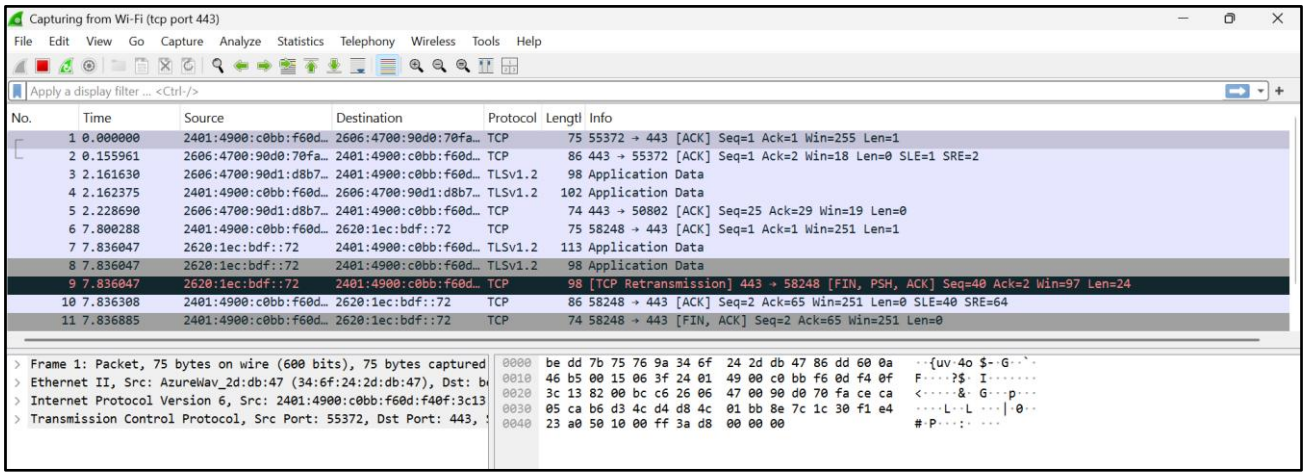
,, ,, ,,

Step 1: Launch Wireshark.

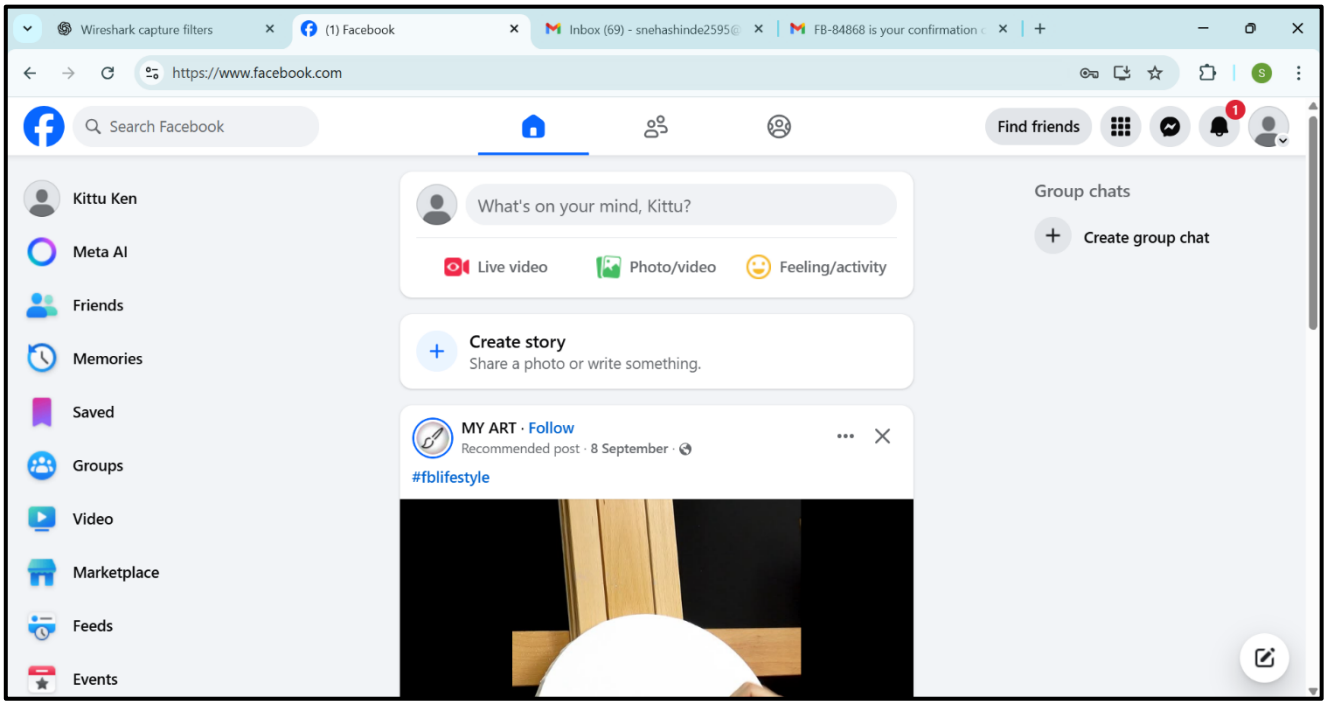


Step 2: Select active network interface and start capturing filters.

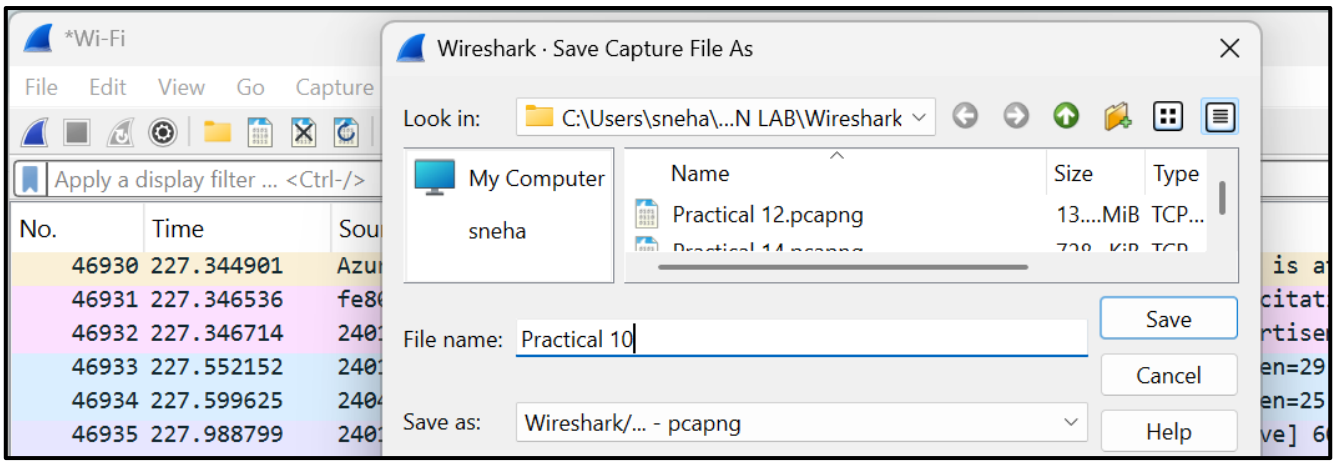




Step 3: Login to your Facebook account.

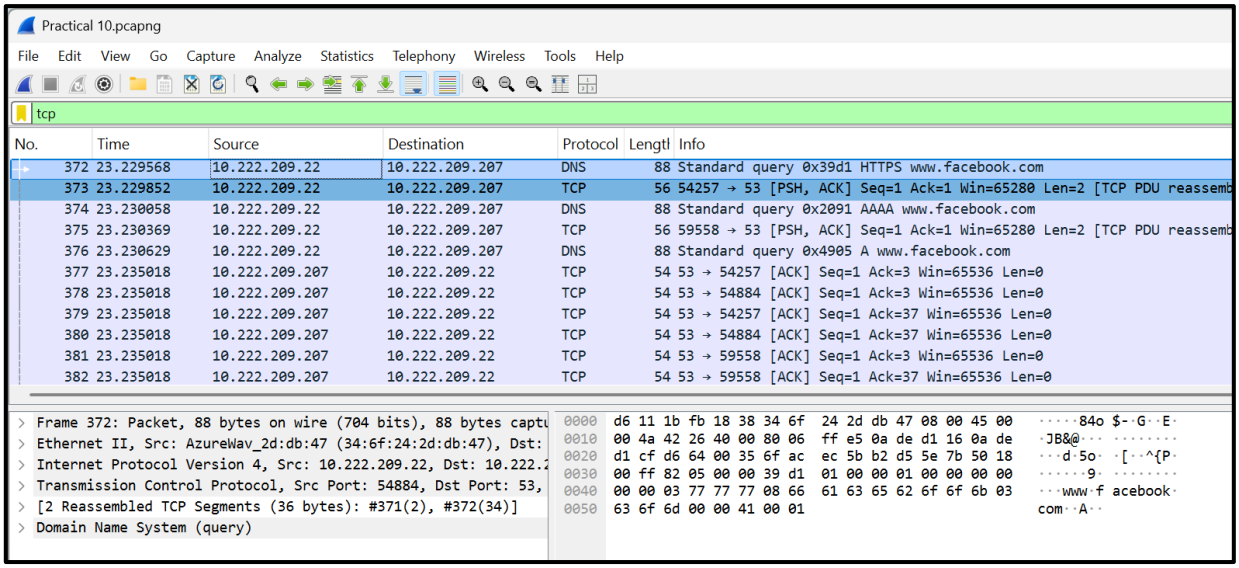


Step 4: Stop capturing and save file.

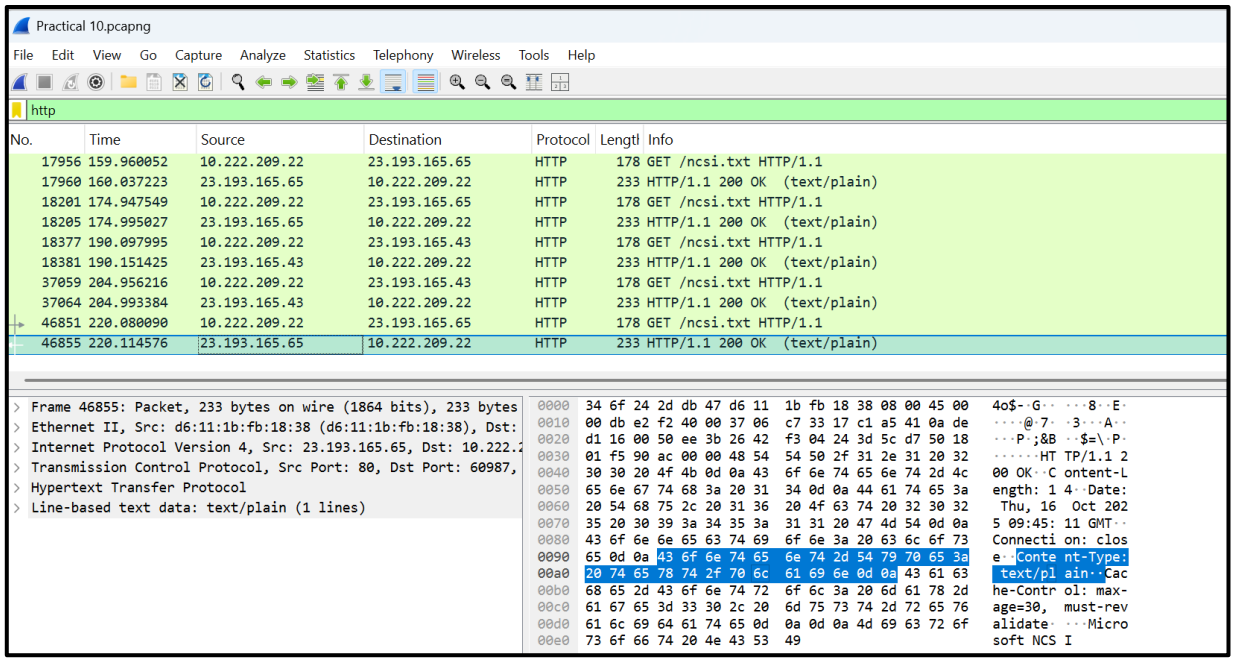


Step 5: Apply display filters :

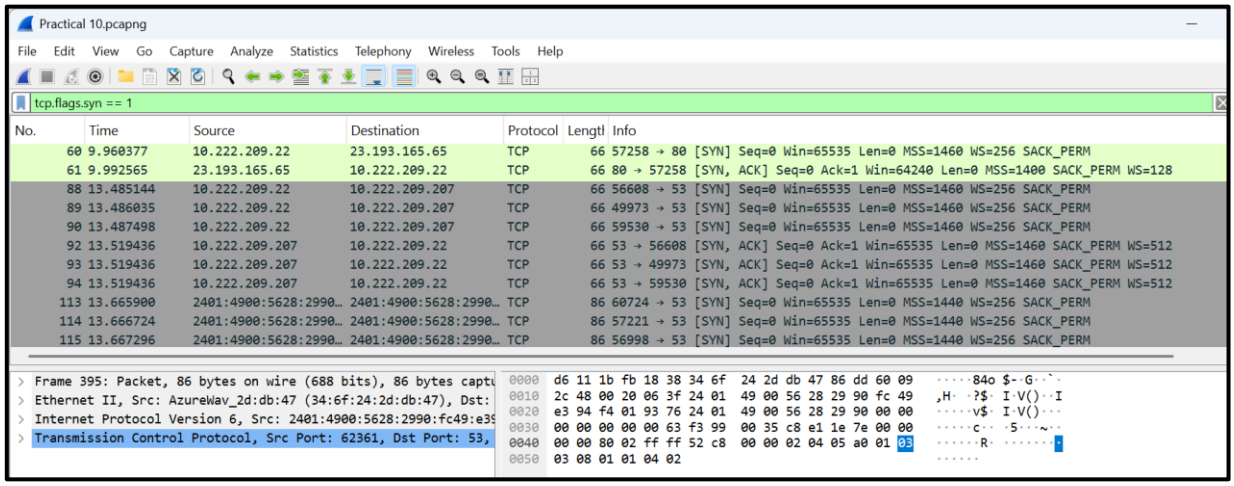
1. TCP Traffic :



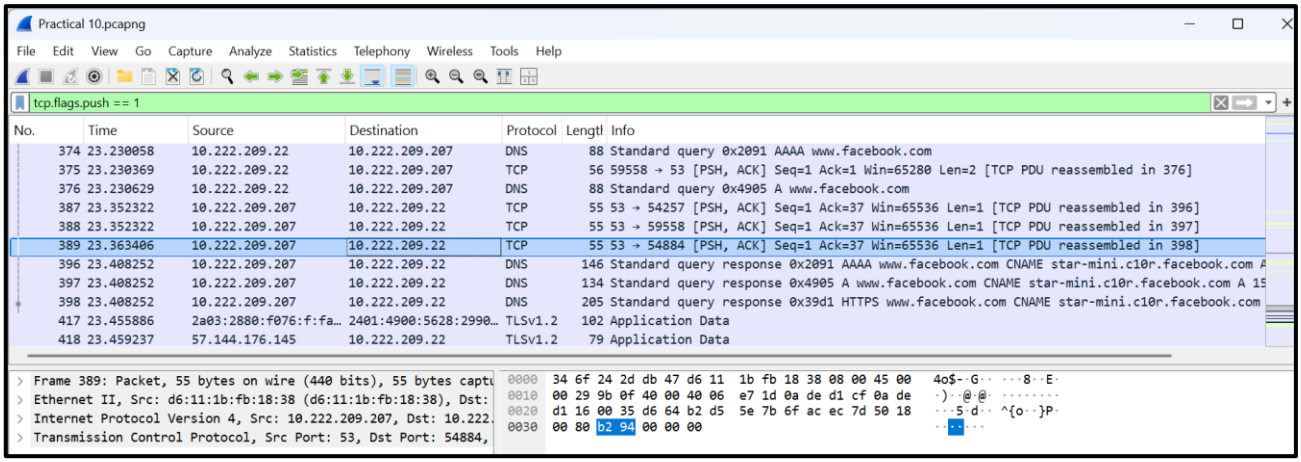
2. HTTP Traffic :



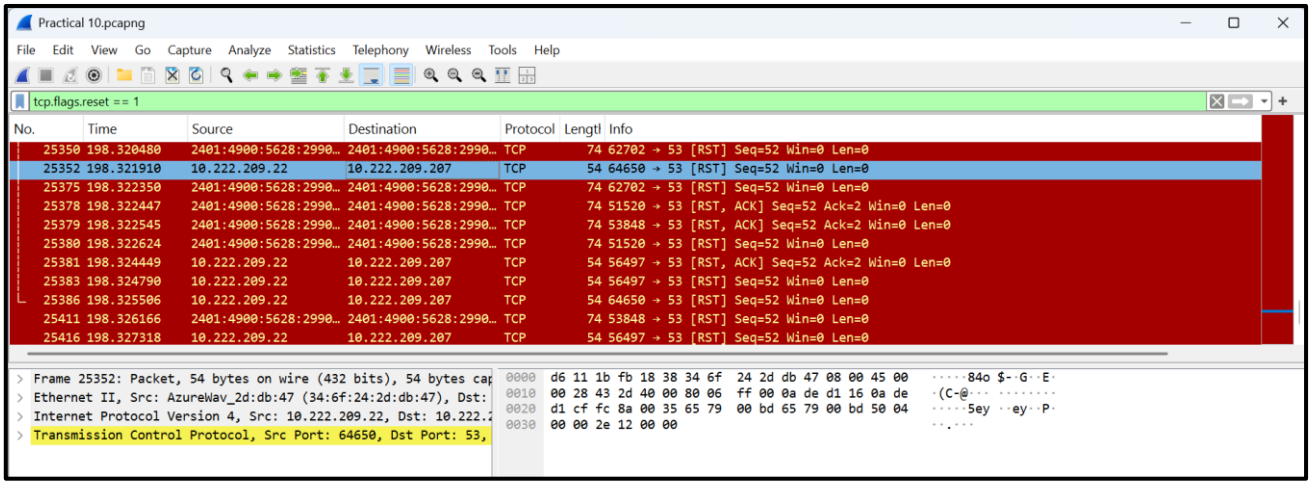
3. TCP SYN Packets :



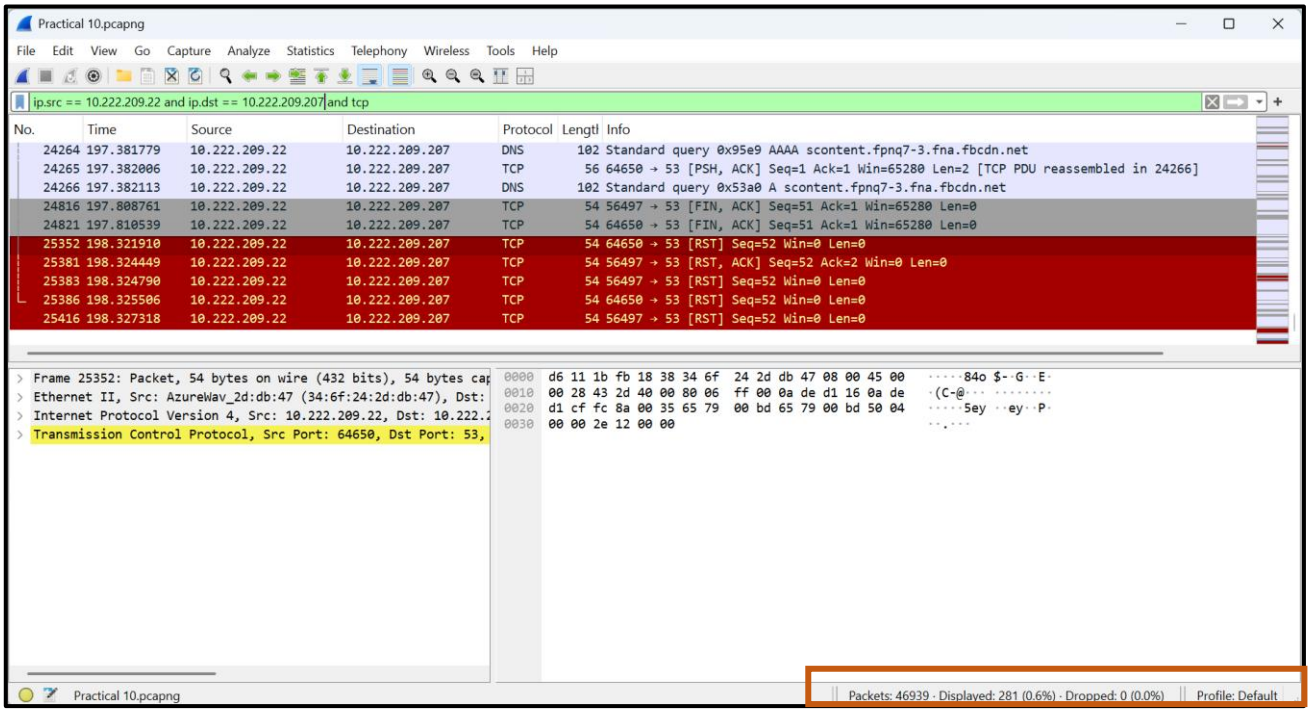
4. TCP PSH Packets :



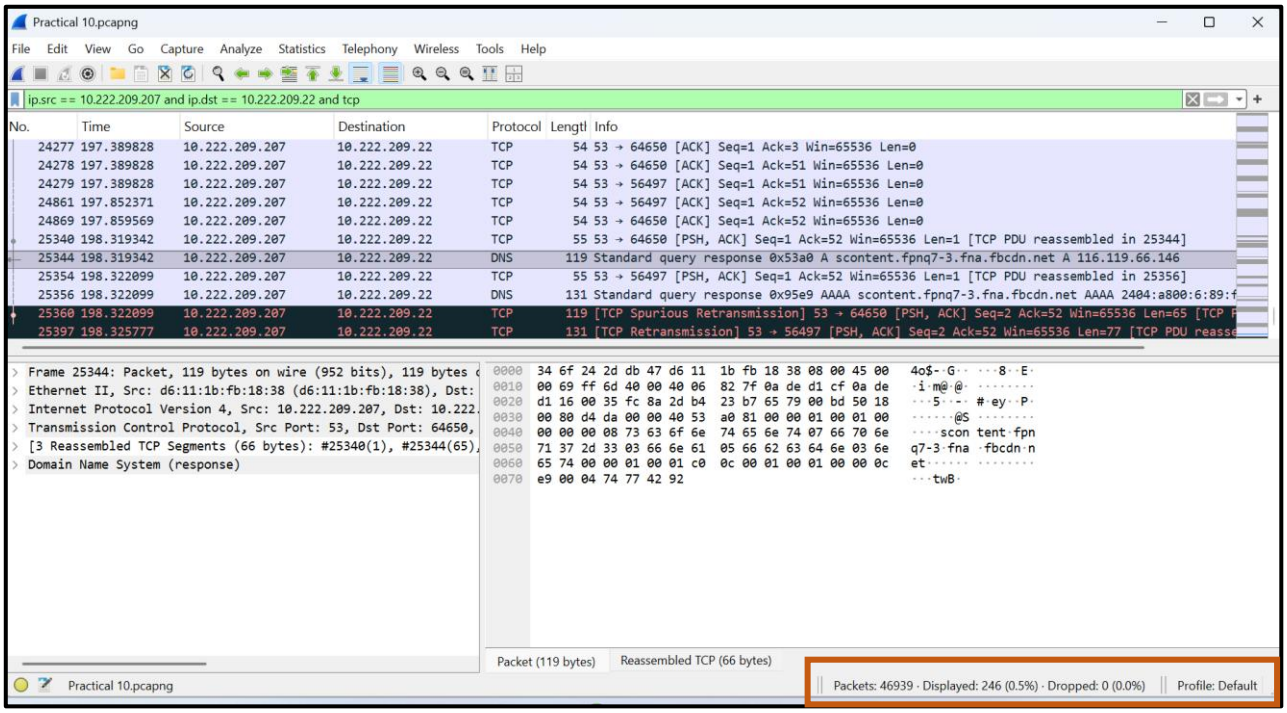
5. TCP RST Packets :



6. Count TCP Packets **Sent to** Facebook.



7. Count TCP Packets **Received from** Facebook.



****

Experiment No : 11

,, ,, ,,

Roll No : TE-43

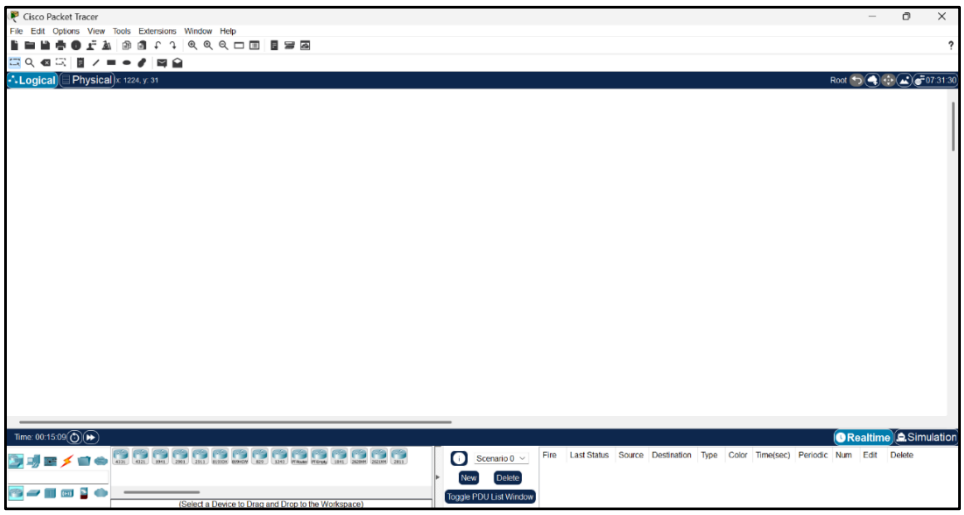
Name : Sanika Nilesh Patil

Title : Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer tool.

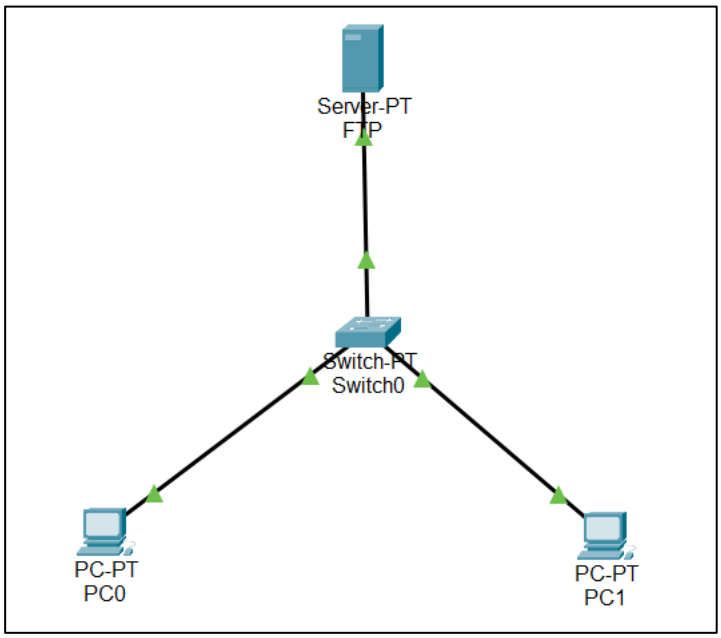
,, ,, ,,

FTP Protocol :

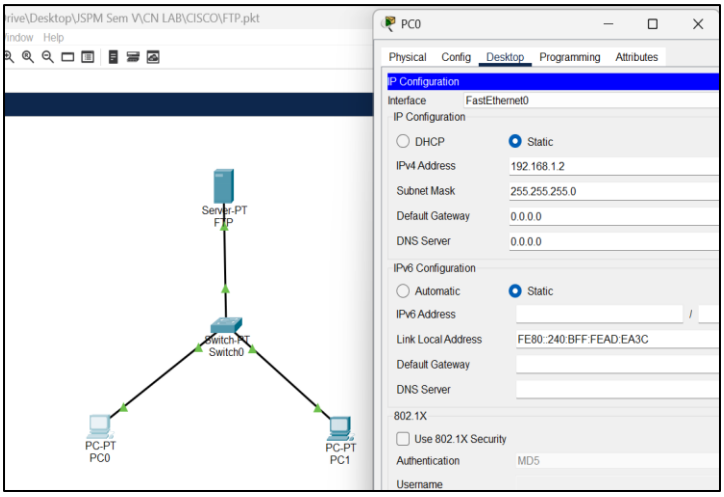
Step 1: Launch Packet Tracer.

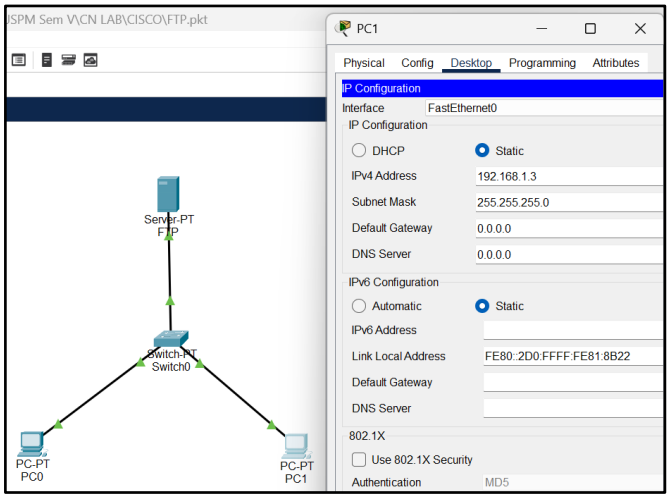


Step 2: Build the topology (1 Server, 1 Switch and 2 PCs) .



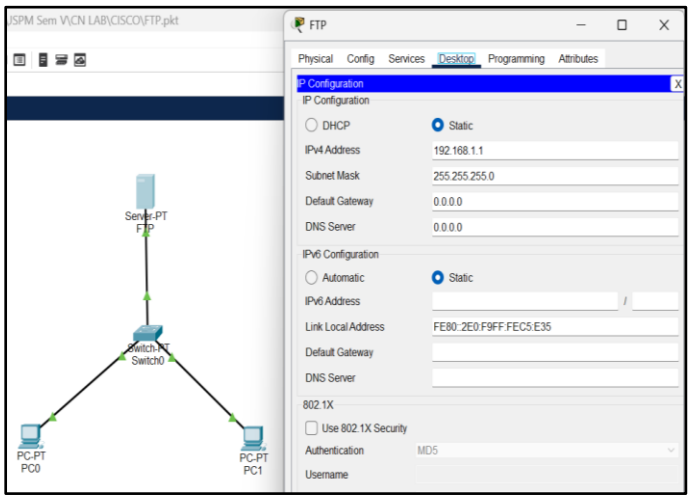
Step 3: Configure IP addresses and subnet masks on all PCs.

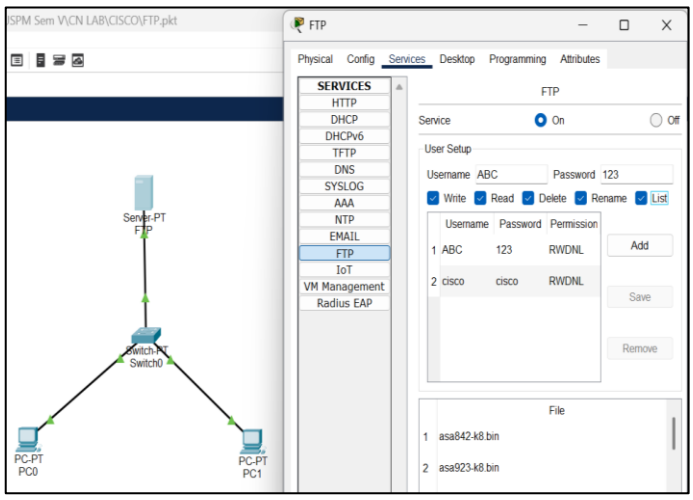




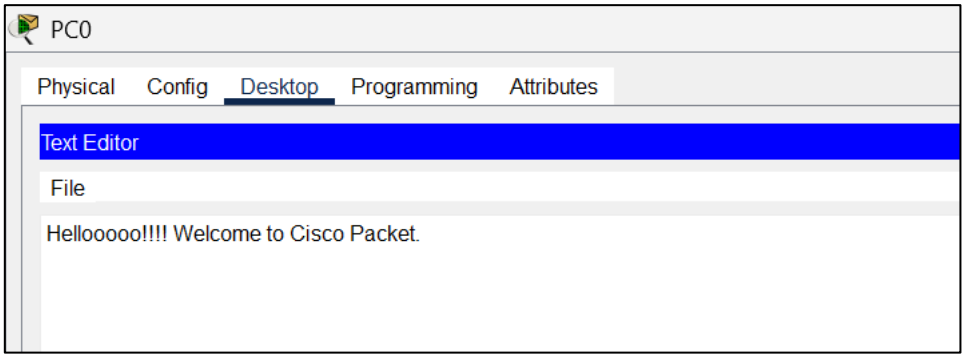
Step 4: Configure IP addresses and subnet masks on Server.

- Go to **Services** tab → select **FTP**.
- Turn FTP **ON** (enable the service).
- In the FTP settings you can **create users**. Add a user:
  - Username: ABC
  - Password: 123



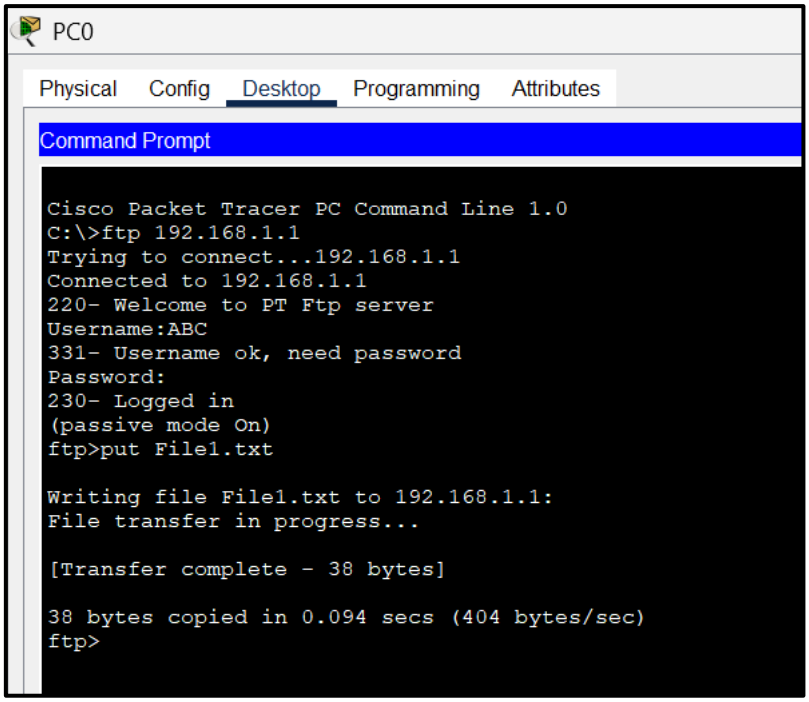


Step 5: Create text file on any PC.



Step 6 : Using FTP from PC.

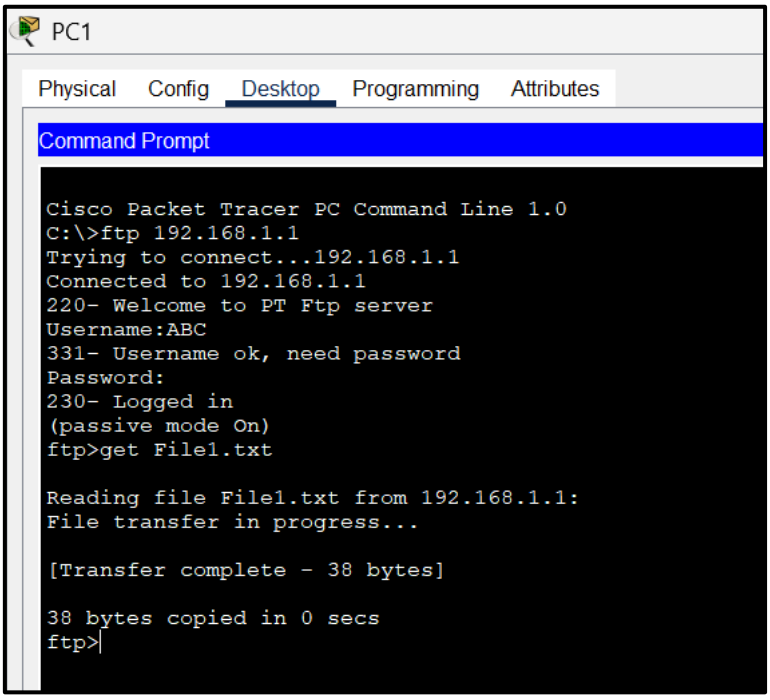
Open PC0 → Desktop → Command Prompt and run:



Step 7 : Verify FTP Connection from PC1

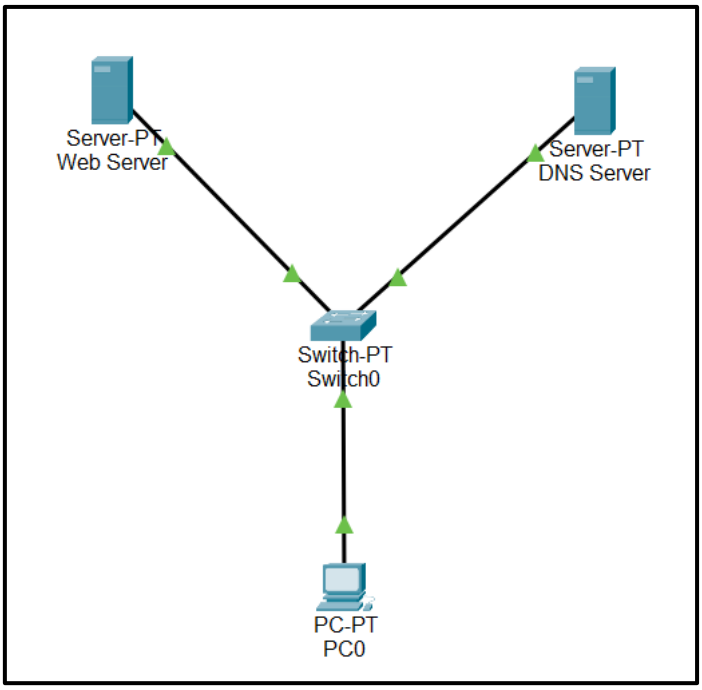
Use FTP commands from another PC to ensure the connection and file transfer are working correctly.

Open PC1 → Desktop → Command Prompt and run :



HTTP and HTTPS Protocol :

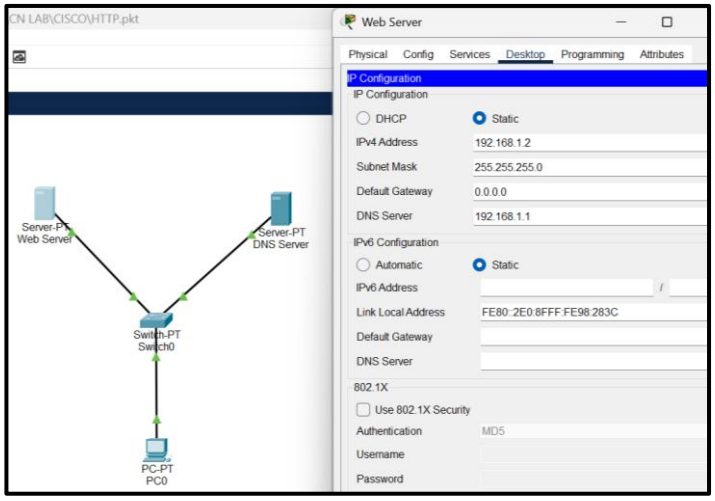
Step 1: Build the topology (1 Server, 1 Switch and 2 PCs) .

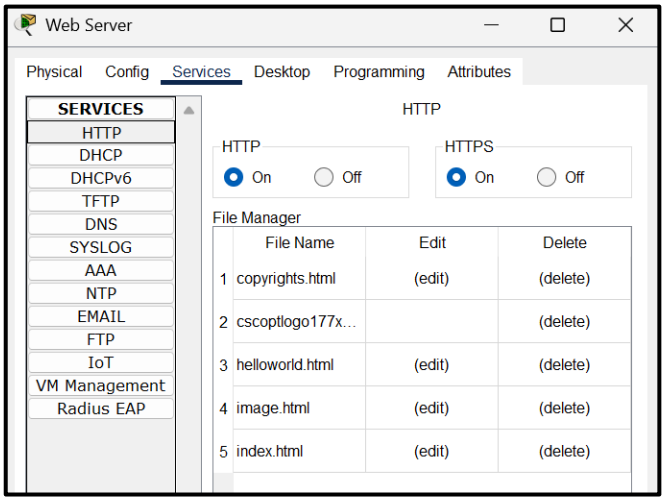


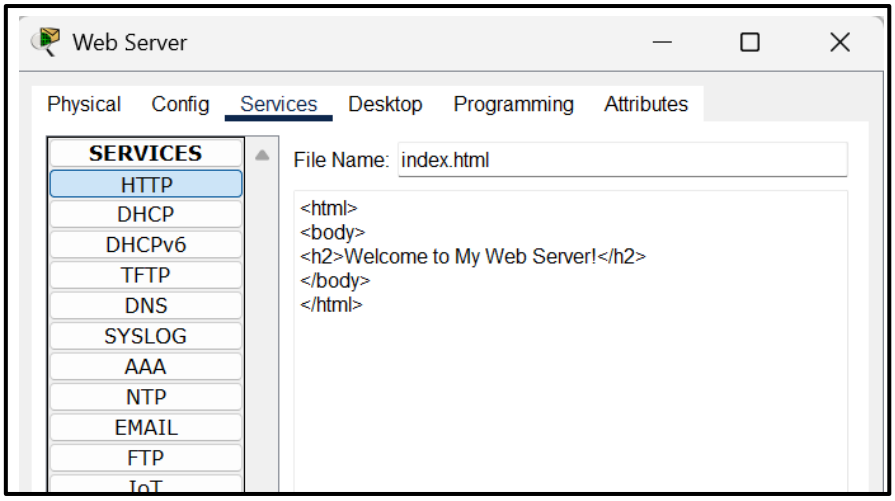
Step 2: Click Web Server → Desktop → IP Configuration

- IP Address: 192.168.1.2
- Subnet Mask: 255.255.255.0

1. Go to Services → HTTP → Turn HTTP ON
2. Go to Services → HTTPS → Turn HTTPS ON and
3. Edit index.html page content under HTTP service

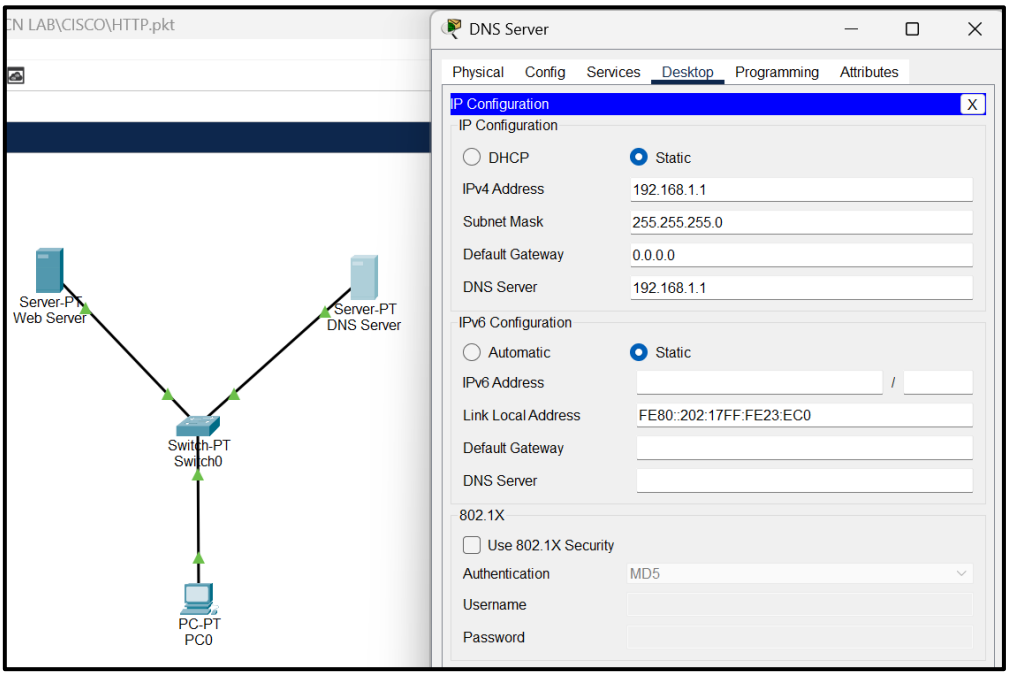


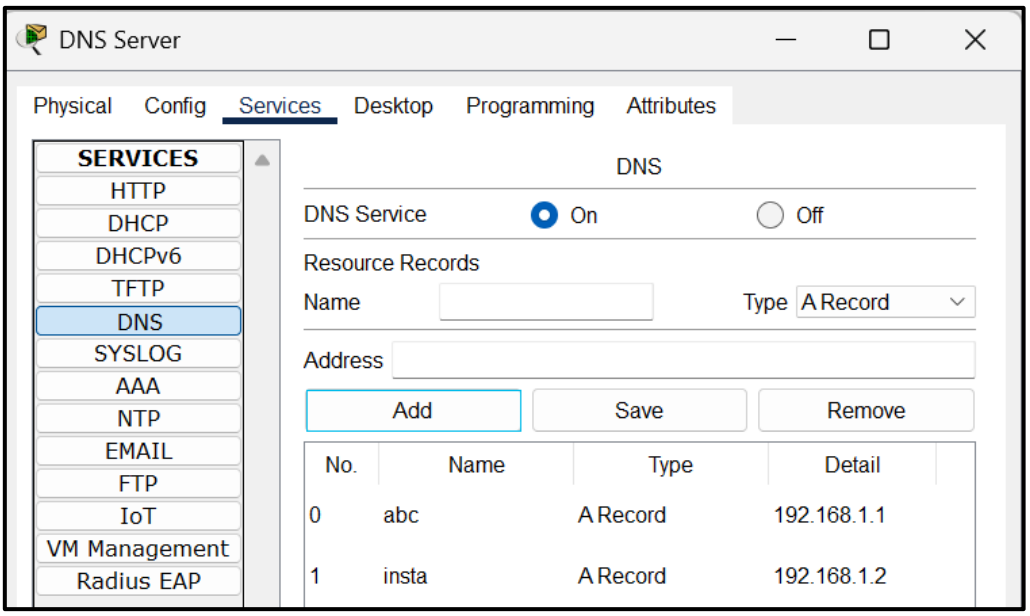




Step 4: Configure DNS Server

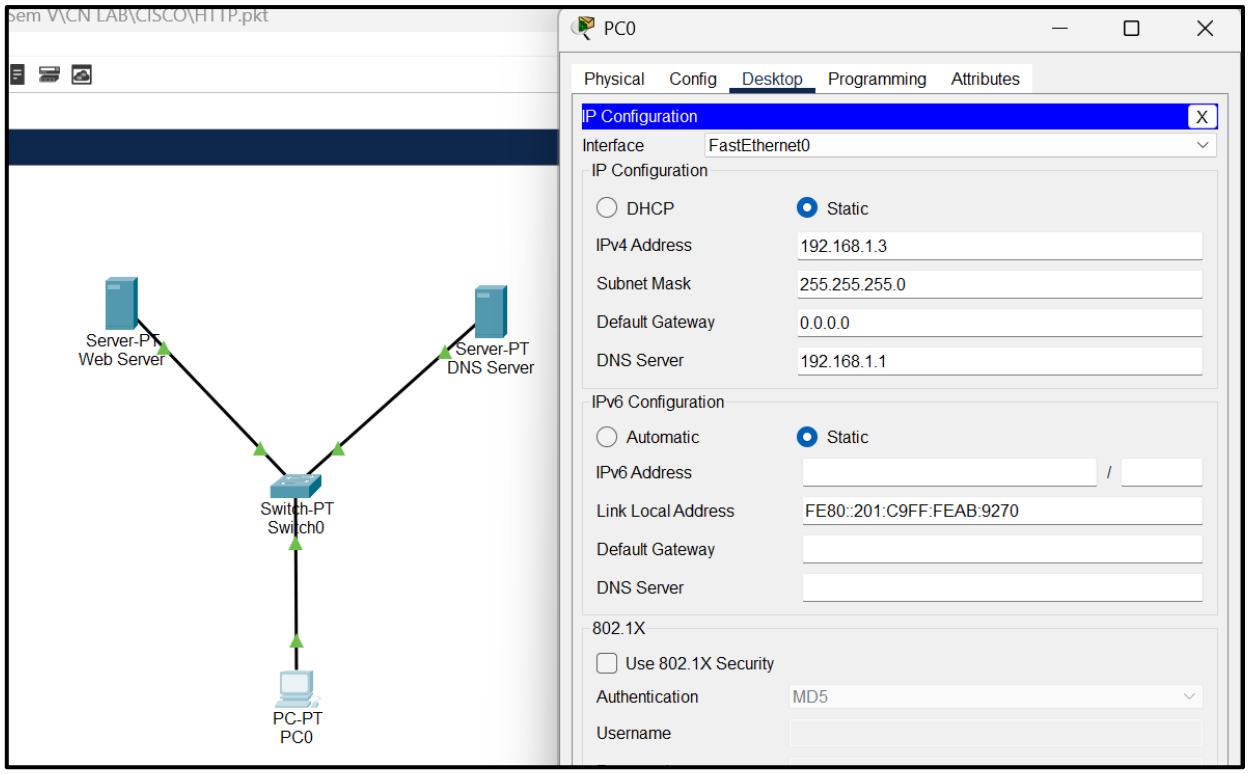
1. Click Server1 → Desktop → IP Configuration.
    - IP Address: 192.168.1.1
    - Subnet Mask: 255.255.255.0
2. Go to Services → DNS
    - Turn DNS ON
    - Under “Name” enter: Insta
    - Under “Address” enter: 192.168.1.2
    - Click Add





Step 5: Click PC0 → Desktop → IP Configuration

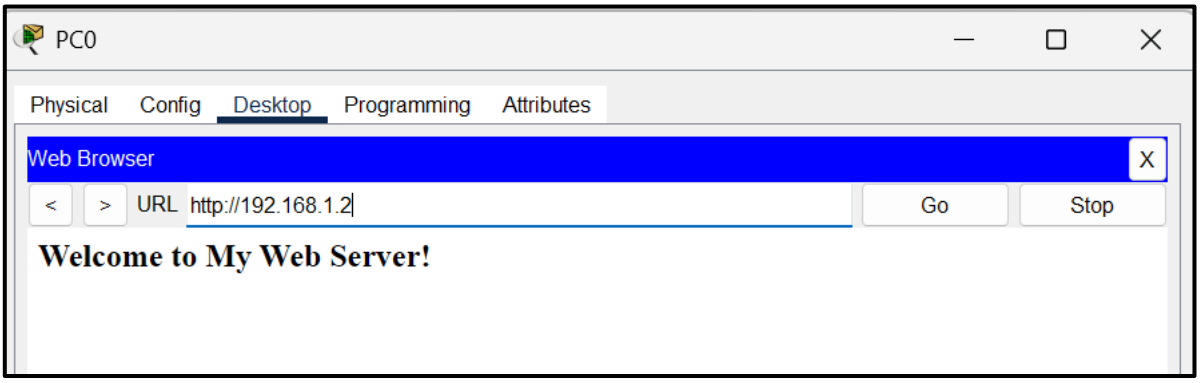
- IP Address: 192.168.1.3
- Subnet Mask: 255.255.255.0
- DNS Server: 192.168.1.1

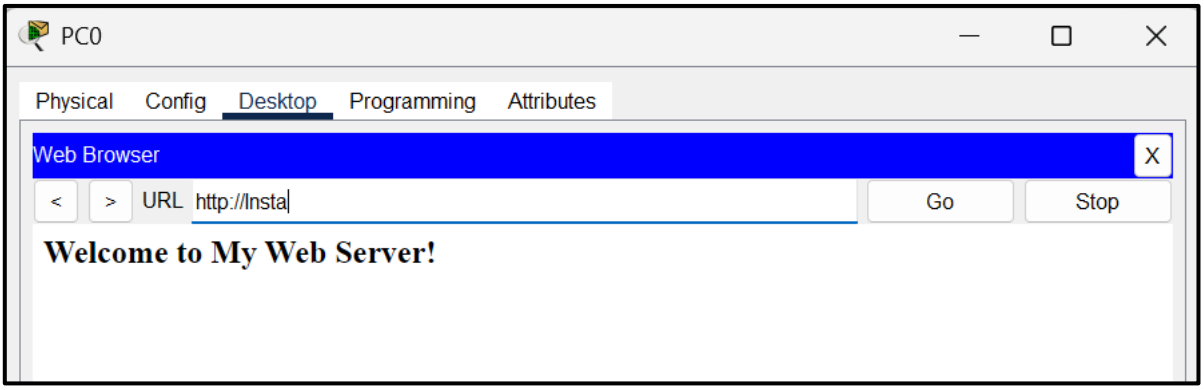


Step 6 : Verify HTTPS Connection from PC0

Use the Web Browser to test the secure connection.

1. On PC0 → Desktop → Web Browser
2. Type:





****

Experiment No : 12

,, ,, ,,

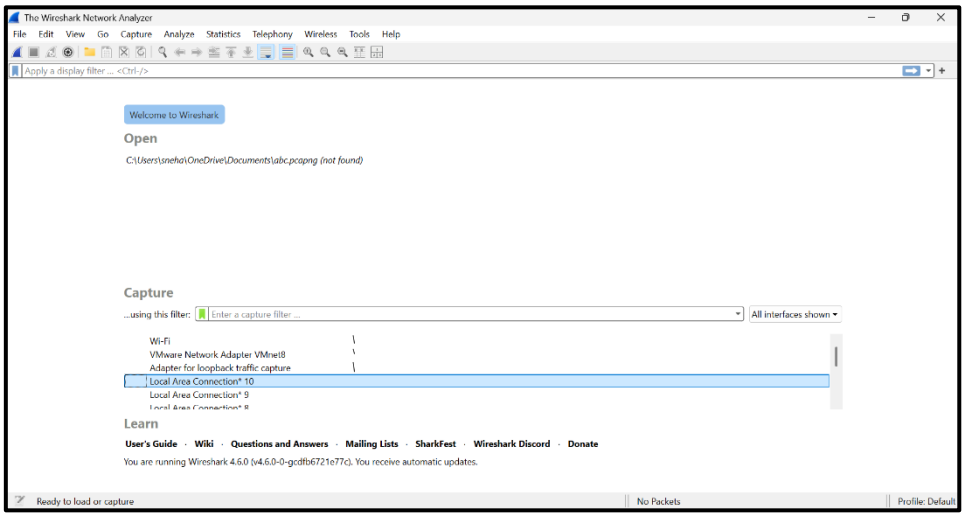
Roll No : TE-43

Name : Sanika Nilesh Patil

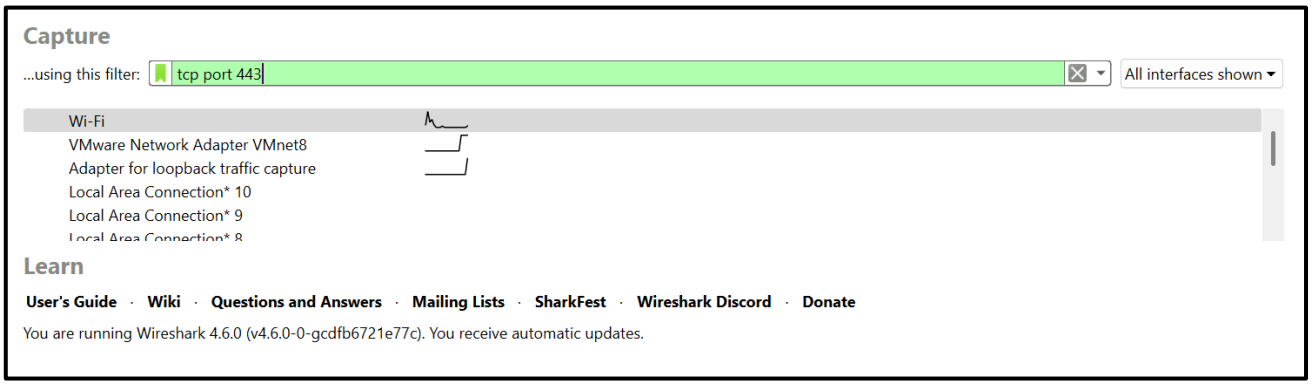
Title : To study the SSL (Secure Sockets Layer) protocol by capturing packets using Wireshark while visiting any SSL-secured website (e.g., online banking, e-commerce, or secure login pages).

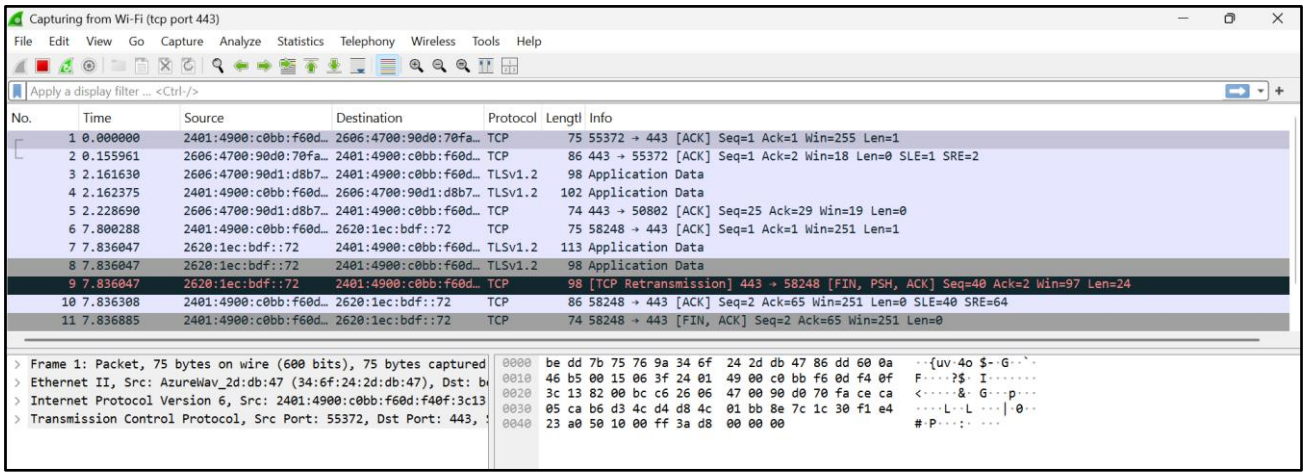
,, ,, ,,

Step 1: Launch Wireshark.



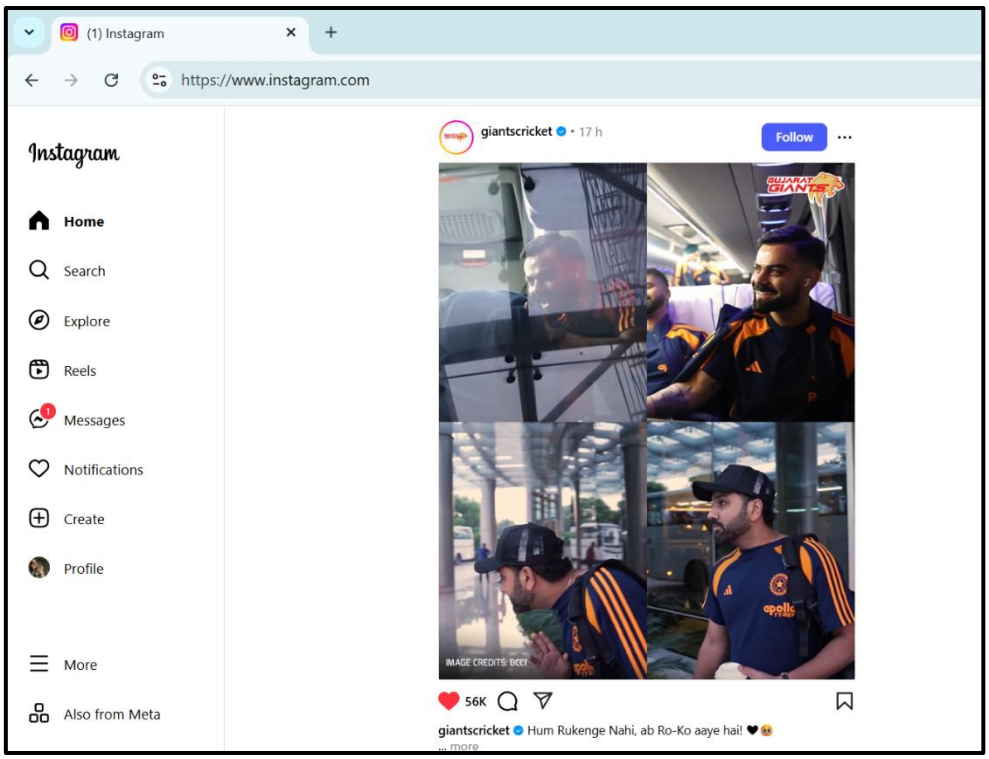
Step 2: Capture filter → TCP port 443.



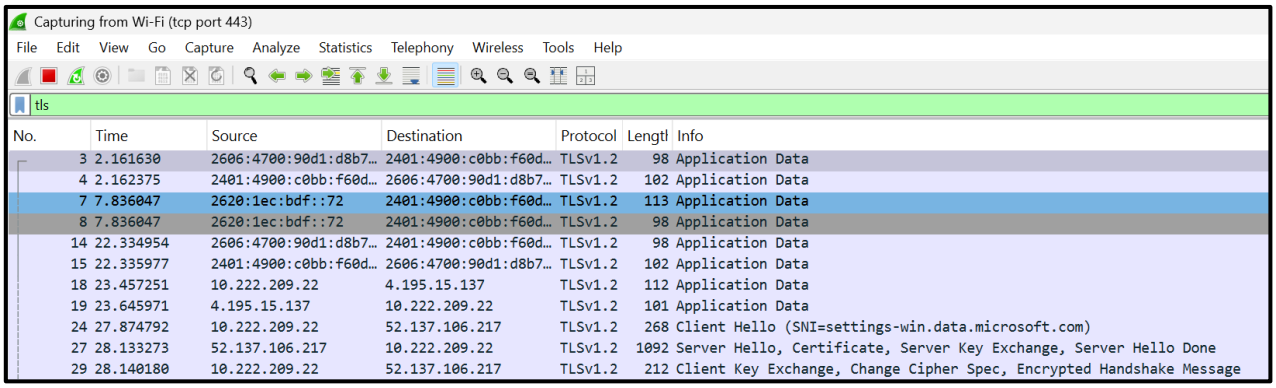


Step 3: Open web browser and visit site HTTPs.

Wait to load page and perform some activity on page.



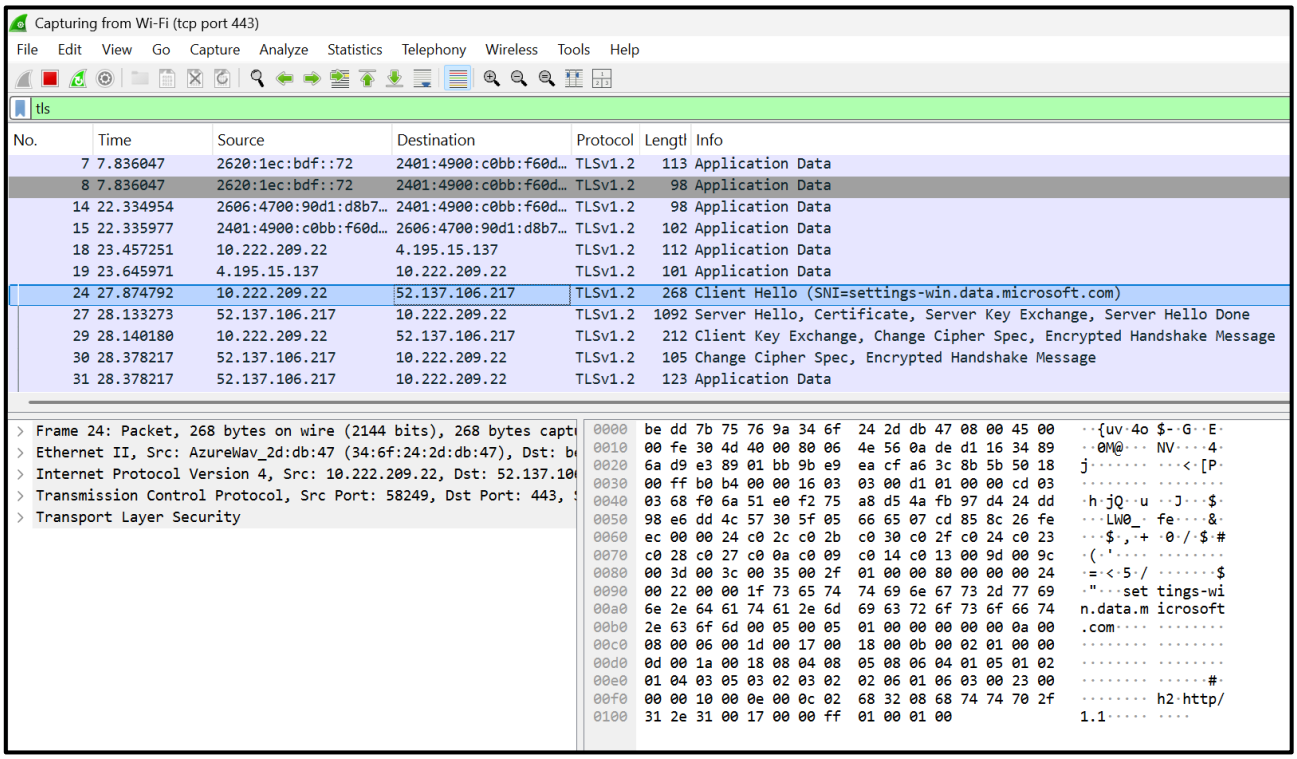
Step 4: Go to Wireshark and type **TLS** to view TLS/SSL packets.



Step 5: Observe Client Hello, Server Hello and Certificate message in packet list.

Click on packet to see details :

1. TLS Version
2. Cipher suits
3. Certificate Info



****

Experiment No : 14

,, ,, ,,

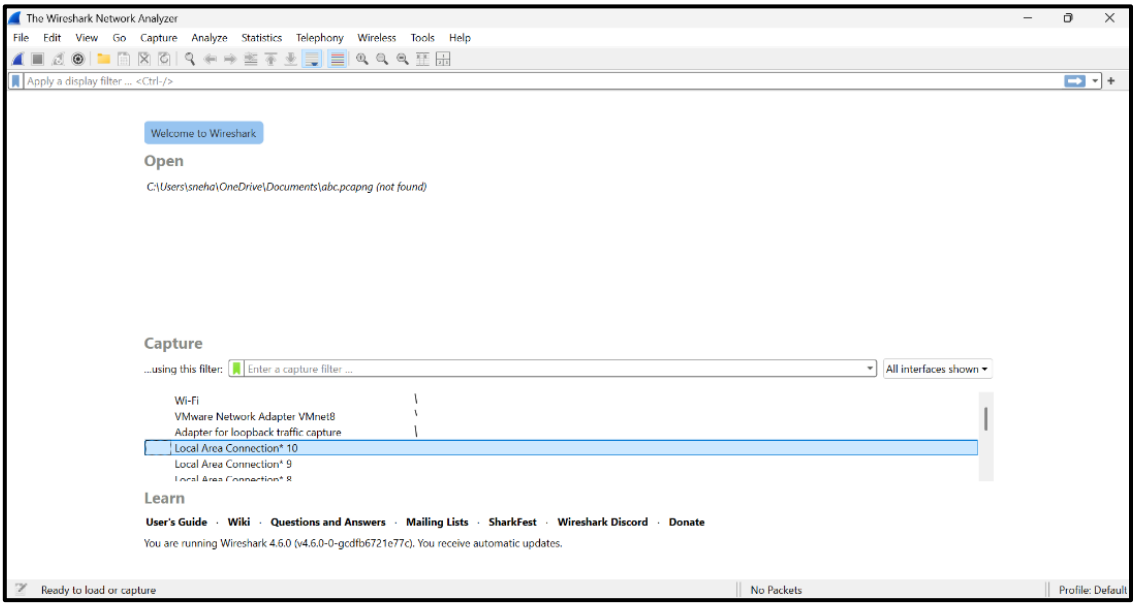
Roll No : TE-43

Name : Sanika Nilesh Patil

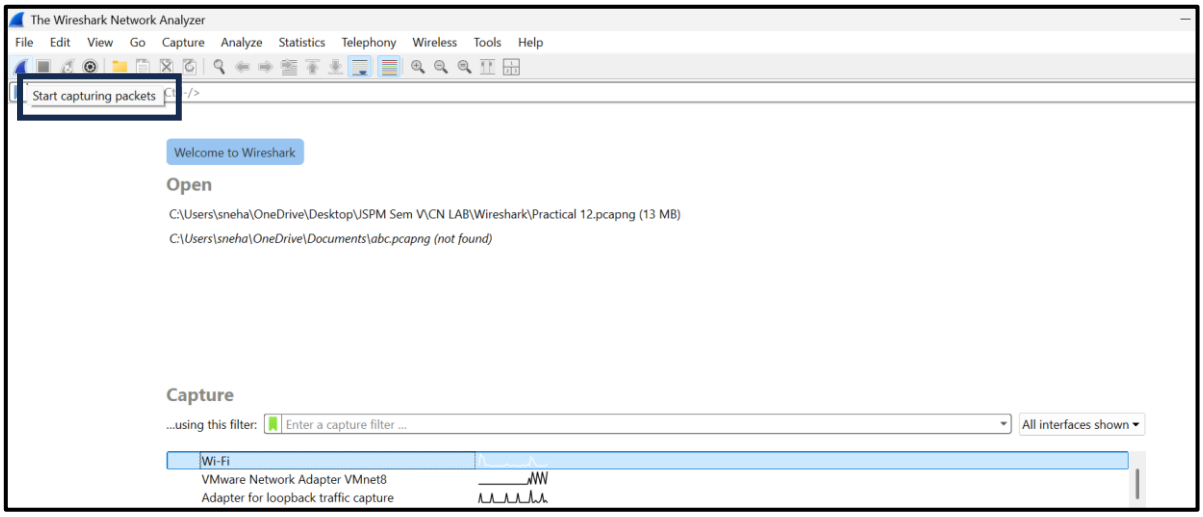
Title : To study the IPsec (Internet Protocol Security) protocol by capturing packets using Wireshark while observing ESP (Encapsulating Security Payload) and AH (Authentication Header) headers during secure communication.

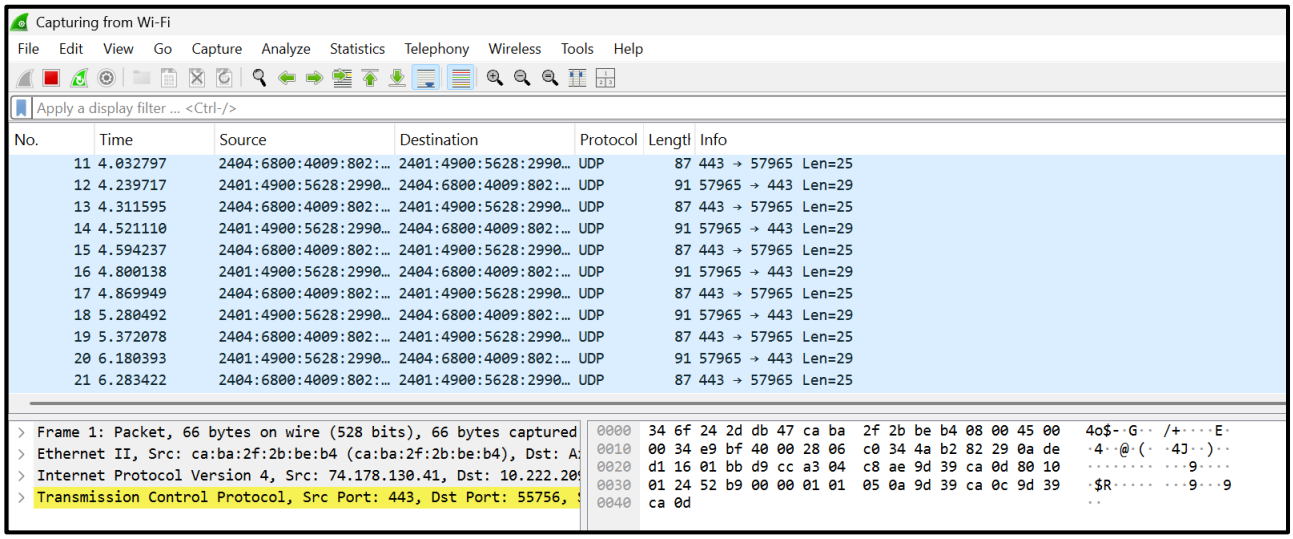
,, ,, ,,

Step 1: Launch Wireshark.



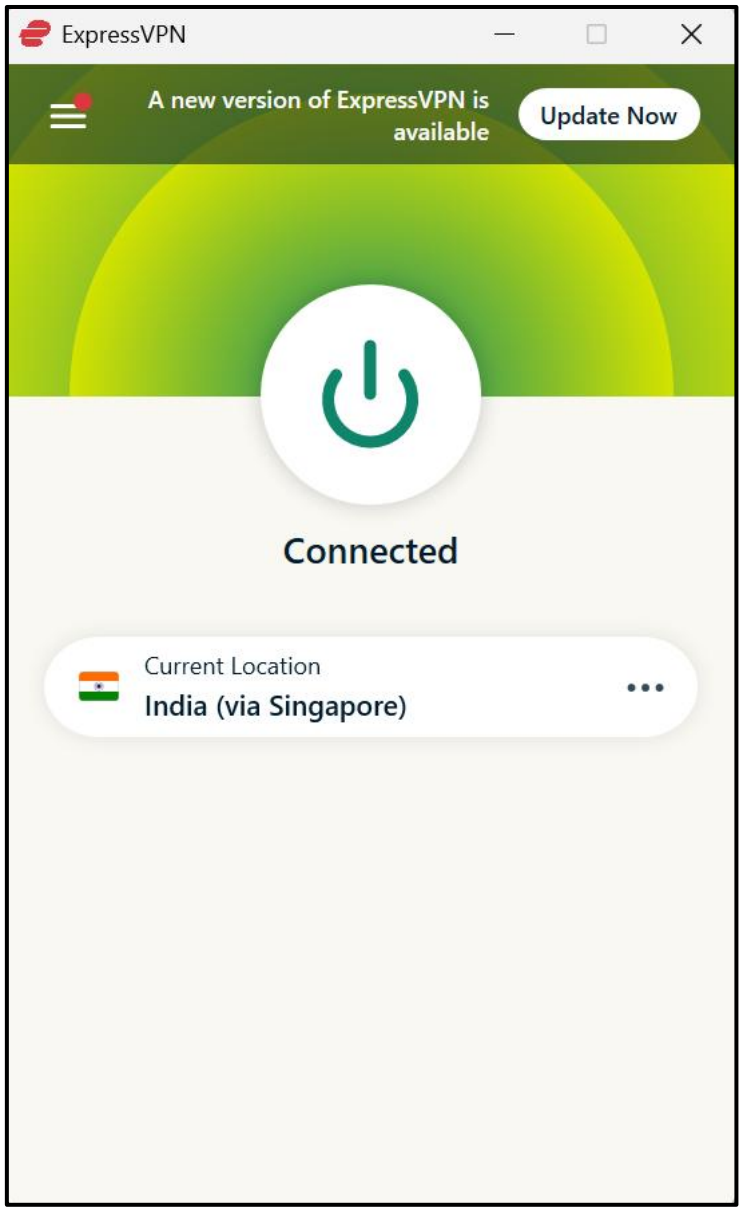
Step 2: Select network interface (Wi-Fi) and click on start capturing packets.





Step 3: Establish IPSec Connection (Connect to ExpressVPN).

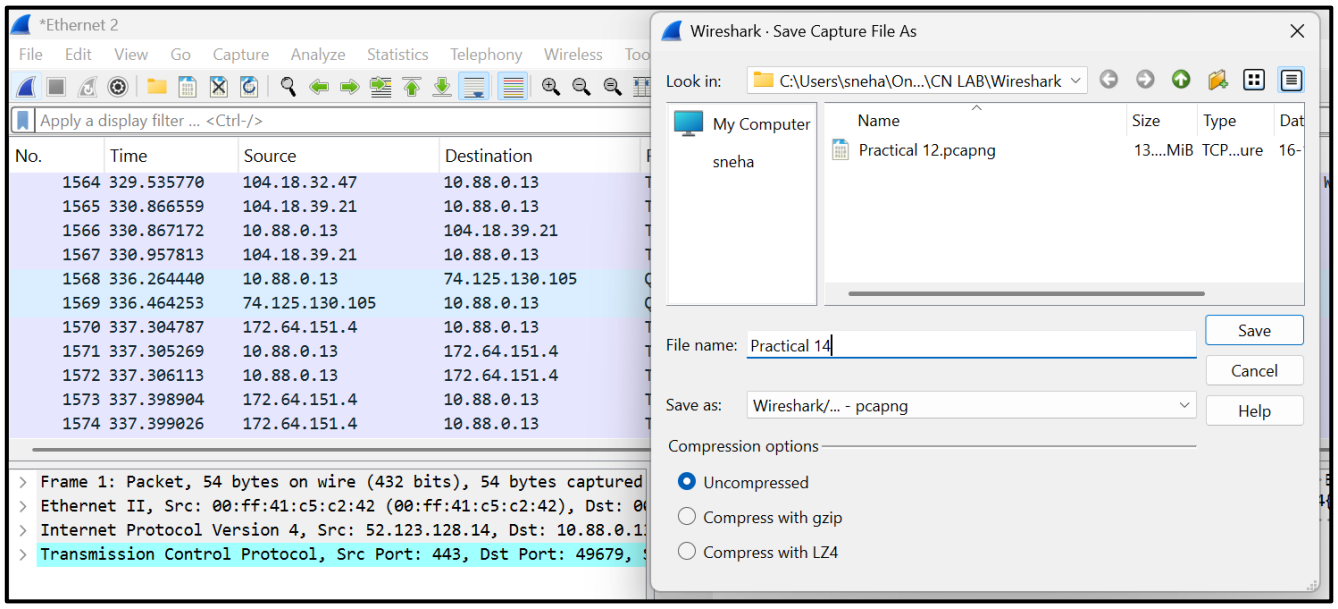
1. Open ExpressVPN and choose a server(any server location).
2. While the VPN is connecting , Wireshark will automatically capture all network packets.



Step 4: Stop capture packets.

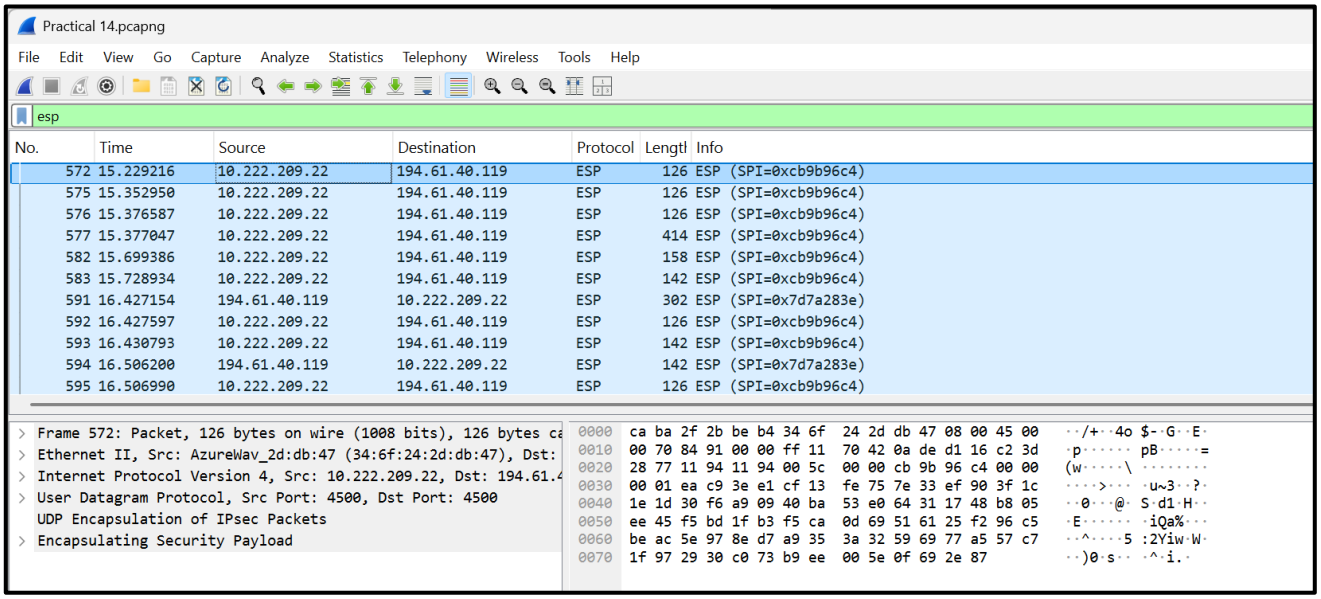
1. After 30–60 seconds of connection (or once traffic appears in Wireshark), click the Red Stop button.
2. Save the capture:

   File → Save As → practical_14.pcapng

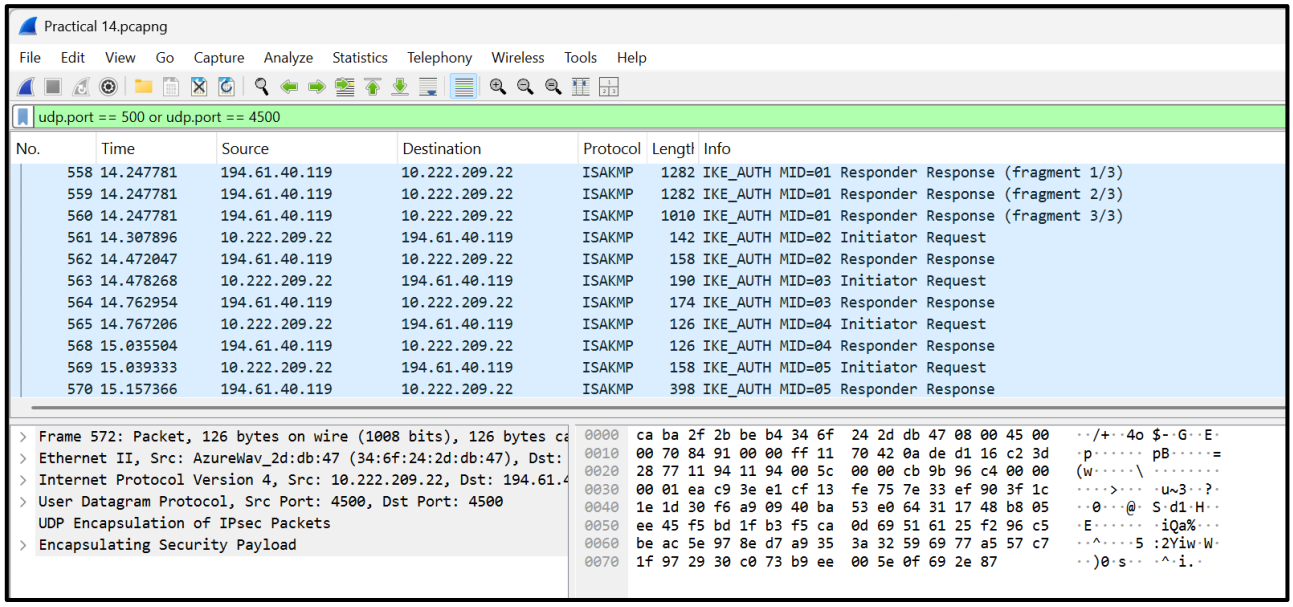


Step 5: Apply display filters in Wireshark.

1. For all IPSec traffic – esp or ah.



2. VPN negotiation packets : udp.port == 500 or udp,port == 4500



****

Experiment No : 15

,, ,, ,,

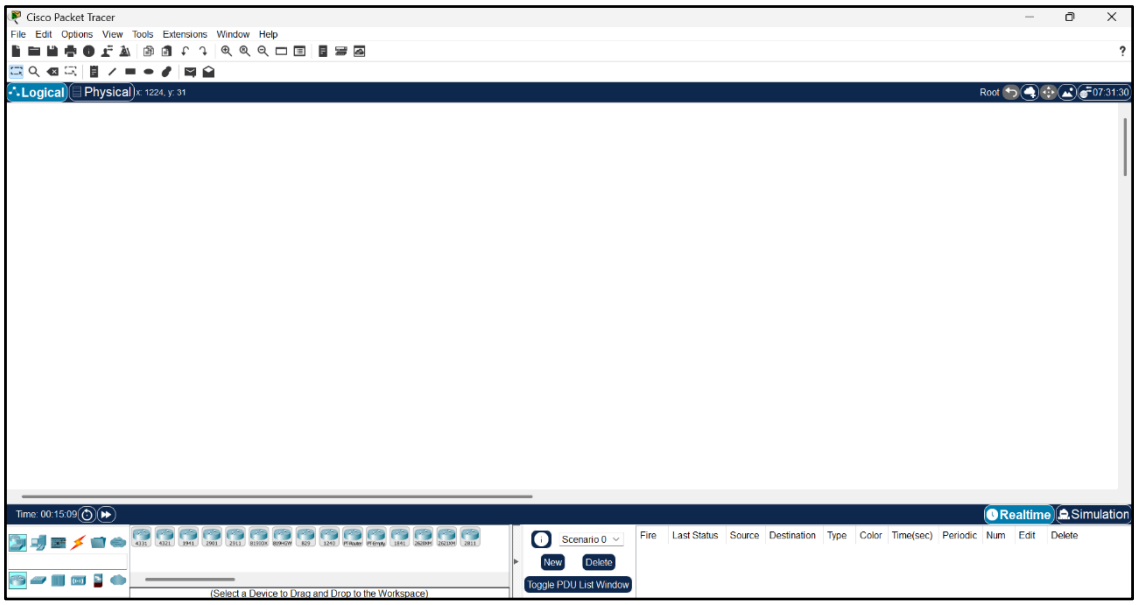
Roll No : TE-43

Name : Sanika Nilesh Patil

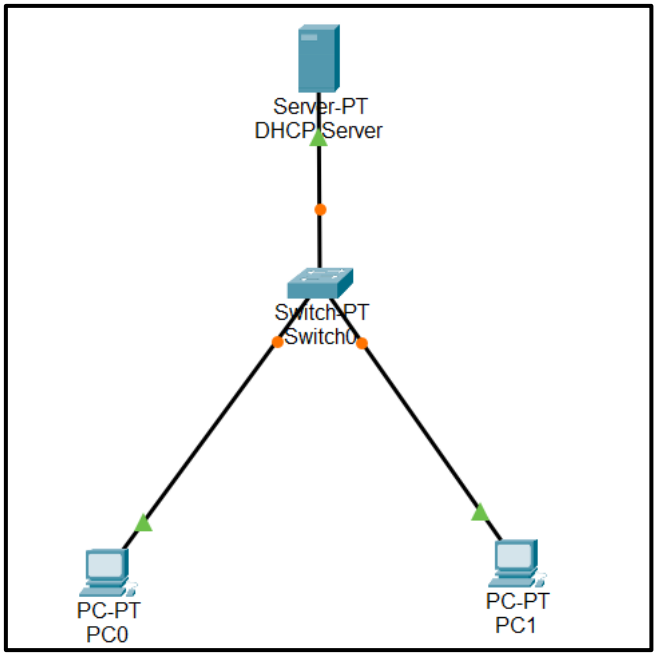
Title : To install and configure a DHCP (Dynamic Host Configuration Protocol) Server and assign IP addresses automatically to client machines connected in a network.

,, ,, ,,

Step 1: Launch Packet Tracer.

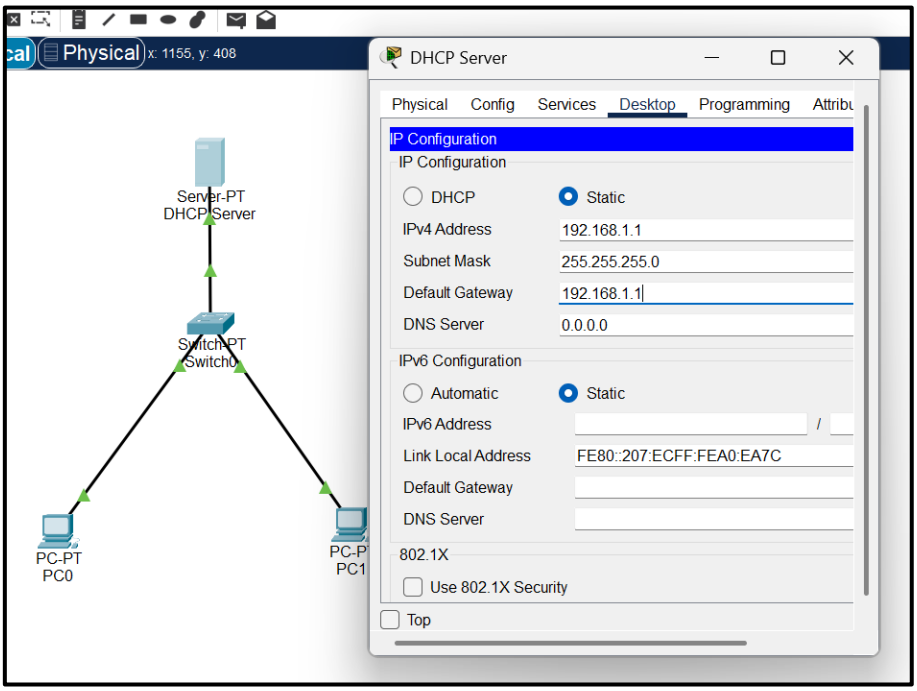


Step 2: Build the topology.

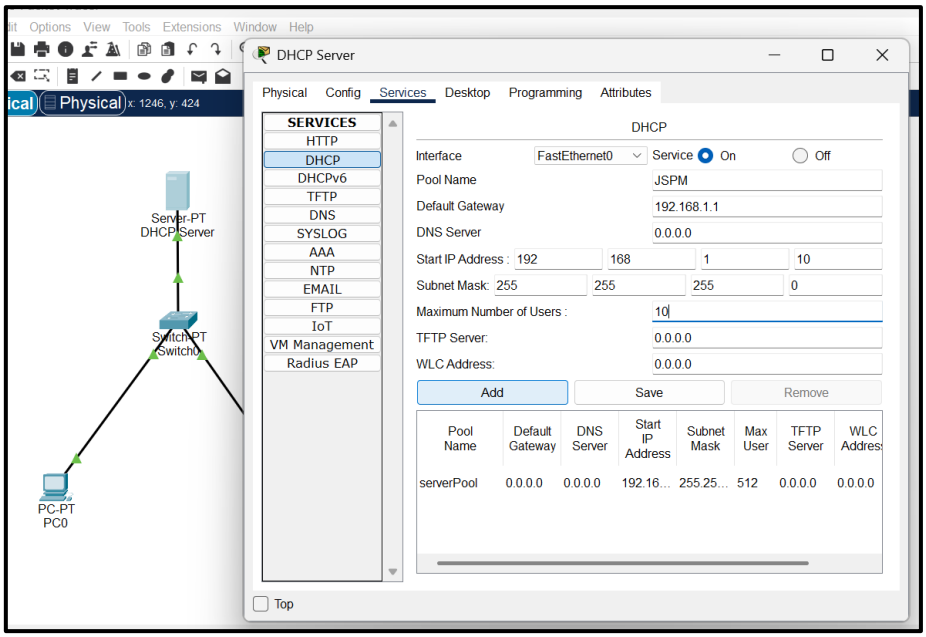


Step 3: Configure DHCP Server.

1. Assign IP address to DHCP server.

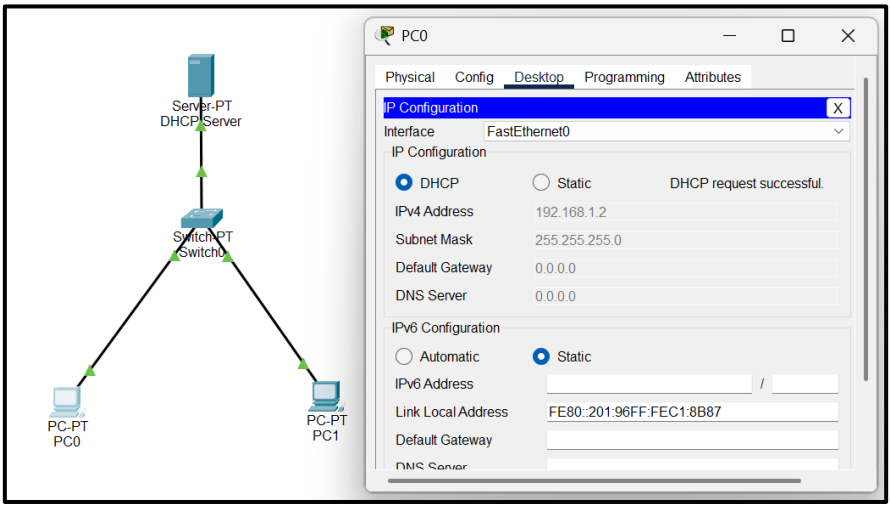


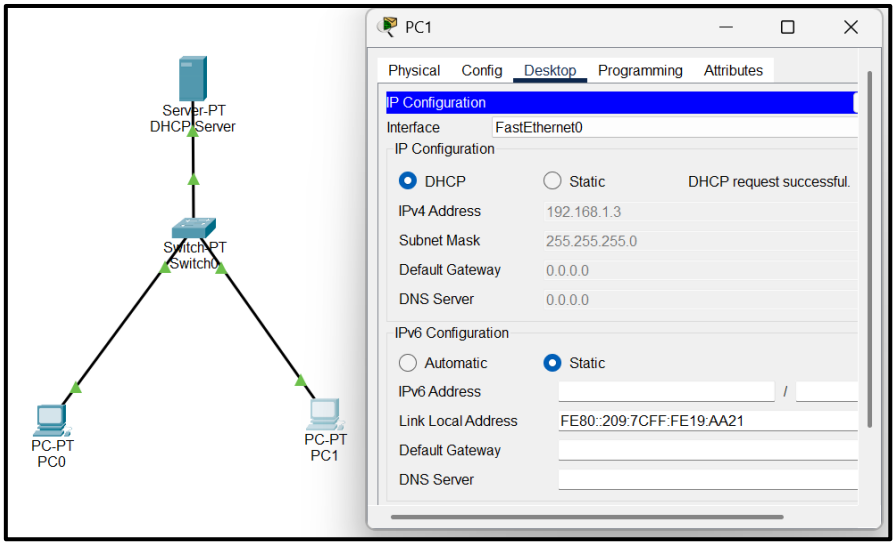
2. Activate DHCP service from service tab.
   - Pool name : JSPM
   - Default Gateway : 192.168.1.1
   - Start IP address : 192.168.1.10
   - Maximum Number of users : 10



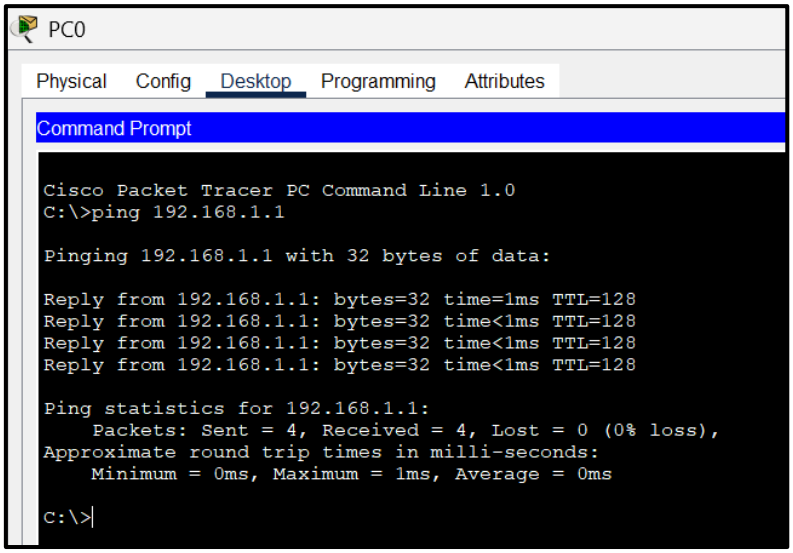
Step 4: Configure IP addresses on all PCs.

Click on PC → Desktop → IP configuration → DHCP





Step 5: Check connectivity using PING command.



****

Experiment No : 16

”” ”” ””

Roll No : TE-43

Name : Sanika Nilesh Patil

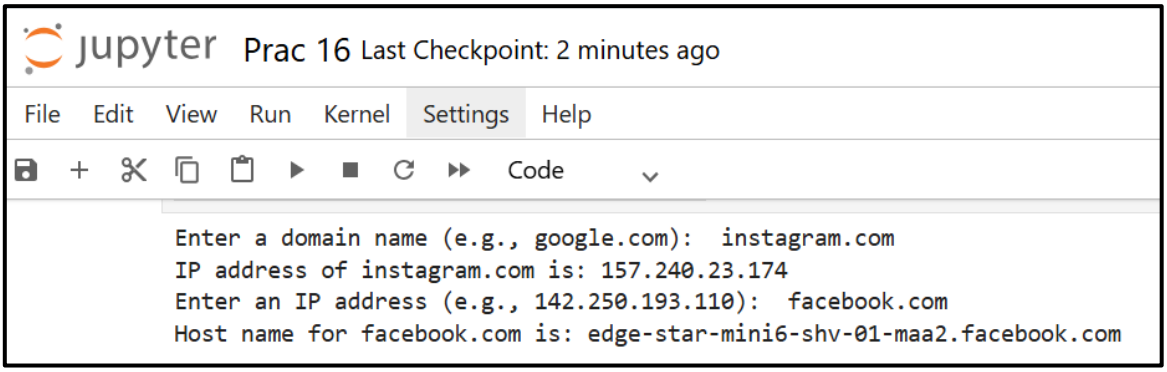
Title : Write a program for DNS lookup. Given an IP address input, it should return URL and vice versa.

”” ”” ””

- **Source Code :**

```
import socket

# Forward lookup (URL to IP)
url = input("Enter a domain name (e.g., google.com): ")
try:
    ip = socket.gethostbyname(url)
    print(f"IP address of {url} is: {ip}")
except socket.gaierror:
    print("Invalid domain name")

# Reverse lookup (IP to URL)
ip_input = input("Enter an IP address (e.g., 142.250.193.110): ")
try:
    host = socket.gethostbyaddr(ip_input)
    print(f"Host name for {ip_input} is: {host[0]}")
except socket.herror:
    print("Host not found for given IP")
```

- **Output :**



****