



SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

COMPUTER SCIENCE AND ENGINEERING

22CS005 – NETWORKS AND SECURITY LABORATORY

SEMESTER – VII

(2025-26 ODD SEM)



SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

COMPUTER SCIENCE AND ENGINEERING

22CS005 - NETWORKS AND SECURITY LABORATORY

CONTINUOUS ASSESSMENT RECORD

Submitted by

Name:

Reg. No :

Class: IV CSE -

Branch : CSE

BONAFIDE CERTIFICATE

This is to certify that this bonafide record work done by Mr./Ms.

..... (Register No.....) during the academic year 2025-2026.

Faculty In-Charge

Head of the Department

Submitted for the End Semester Practical Examination held on.....

INTERNAL EXAMINER

EXTERNAL EXAMINER



22CS005 – NETWORKS AND SECURITY LABORATORY (COMPONENT)

LIST OF EXPERIMENTS

| Exp. No. | Date | Name of the Experiment | Page No. |
|---------------------------------|-------------|---|---------------------|
| 1. | | | |
| 2. | | | |
| 3. | | | |
| 4. | | | |
| 5. | | | |
| 6. | | | |
| 7. | | | |
| 8. | | | |
| ADDITIONAL EXPERIMENT | | | |
| 9. | | Implement A Home or Small Business Network Using Wireless Technology and Facilitate with Internet | |
| Signature of the Faculty | | | |



SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade

Kuniamuthur, Coimbatore – 641008

Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

Department of Computer Science and Engineering

Rubrics for Evaluating Laboratory

Subject Code : 22CS005

Lab Name : NETWORKS AND SECURITY LABORATORY (Component)

| Criteria | Range of Marks | | | |
|---|----------------|-------|---------|---------------|
| | Excellent | Good | Average | Below Average |
| Aim and Theoretical Description (20 Marks) | 18-20 | 14-17 | 10-13 | 0-9 |
| Procedure & designing / Algorithm & Coding (30 Marks) | 27-30 | 21-26 | 15-20 | 0-14 |
| Configuration & troubleshooting/ Compilation and Debugging (20 Marks) | 18-20 | 14-17 | 10-13 | 0-9 |
| Simulation Results (20 Marks) | 18-20 | 14-17 | 10-13 | 0-9 |
| Documentation & Viva (10 Marks) | 9-10 | 7-8 | 5-6 | 0-4 |

| OVERALL MARKS | | | |
|---------------|-------|---------|---------------|
| 90-100 | 70-89 | 50-69 | 0-49 |
| Excellent | Good | Average | Below Average |



SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade

Kuniamuthur, Coimbatore – 641008

Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

Department of Computer Science and Engineering

Rubrics based Evaluation

Name of the Laboratory: 22CS005 - NETWORKS AND SECURITY LABORATORY

Reg No:

Name of the Student:

| Components | Exp No and Date | | | | | | | | | |
|---|--|-------|-------|-------|-------|-------|-------|----------------------|-------|--------|
| | Exp 1 | Exp 2 | Exp 3 | Exp 4 | Exp 5 | Exp 6 | Exp 7 | Exp 8 | Exp 9 | Exp 10 |
| Experiments | | | | | | | | | | |
| Lab Dates | | | | | | | | | | |
| Aim and Theoretical Description (20 Marks) | | | | | | | | | | |
| Procedure & designing / Algorithm & Coding (30 Marks) | | | | | | | | | | |
| Configuration & troubleshooting/ Compilation and Debugging (20 Marks) | | | | | | | | | | |
| Simulation Results (20 Marks) | | | | | | | | | | |
| Viva and Documentation (5+5 Marks) | | | | | | | | | | |
| Total | | | | | | | | | | |
| Faculty Signature | | | | | | | | | | |
| | Laboratory Experiments (Average) (100 Marks): | | | | | | | Faculty Sign: | | |

Ex No: 1

Date:

CONFIGURING AND ACCESSING A SWITCH IN PACKET TRACER

Aim:

To simulate simple wired LAN networks using hubs, switches and basic router configurations, establish and test for successful communication between host devices, using Cisco Packet Tracer software

Theory:

Creating a Local Area Network (LAN) can involve wired components.

Router: Central device to manage both wired and wireless connections.

- **Ethernet Switch** (optional): Expands the number of Ethernet ports if needed.
- Hubs can also be used.
- **Ethernet Cables:** For wired connections (Cat5e, Cat6, or higher for better speeds).
- **Wireless Access Point (WAP):** If the router doesn't have built-in Wi-Fi, or if you need to extend wireless coverage.
- **Network Devices:** PCs, laptops, smartphones, etc., to connect to the network.

1. Setting Up the Wired LAN

- **Position Your Router:** Place your router in a central location if possible. This will optimize the wireless coverage and minimize the length of cables for wired devices.
- **Connect to the Internet:** Connect the router to your modem (if separate) using an Ethernet cable. This provides internet access to the network.
- **Wired Device Connections:** Use Ethernet cables to connect your devices (e.g., computers, printers) to the router. If your router has limited Ethernet ports, connect an Ethernet switch to the router, then connect additional devices to the switch.
- **Configuration:** Access the router's web interface by entering the router's IP address in a browser (commonly 192.168.1.1 or 192.168.2.1). Follow the instructions to set up basic network settings like IP addressing (usually DHCP).

Procedure:

WIRED LAN

1. Launch Packet Tracer and Build the Topology:

1. Open Cisco Packet Tracer.

2. Select and drag end devices (like PCs and servers) to the workspace.
3. Select and drag a switch to the workspace.
4. Click on the "Connections" icon (lightning bolt) and choose a suitable cable type (usually copper straight-through).
5. Connect the PCs and server to the switch using the chosen cables.

2. Configure IP Addresses:

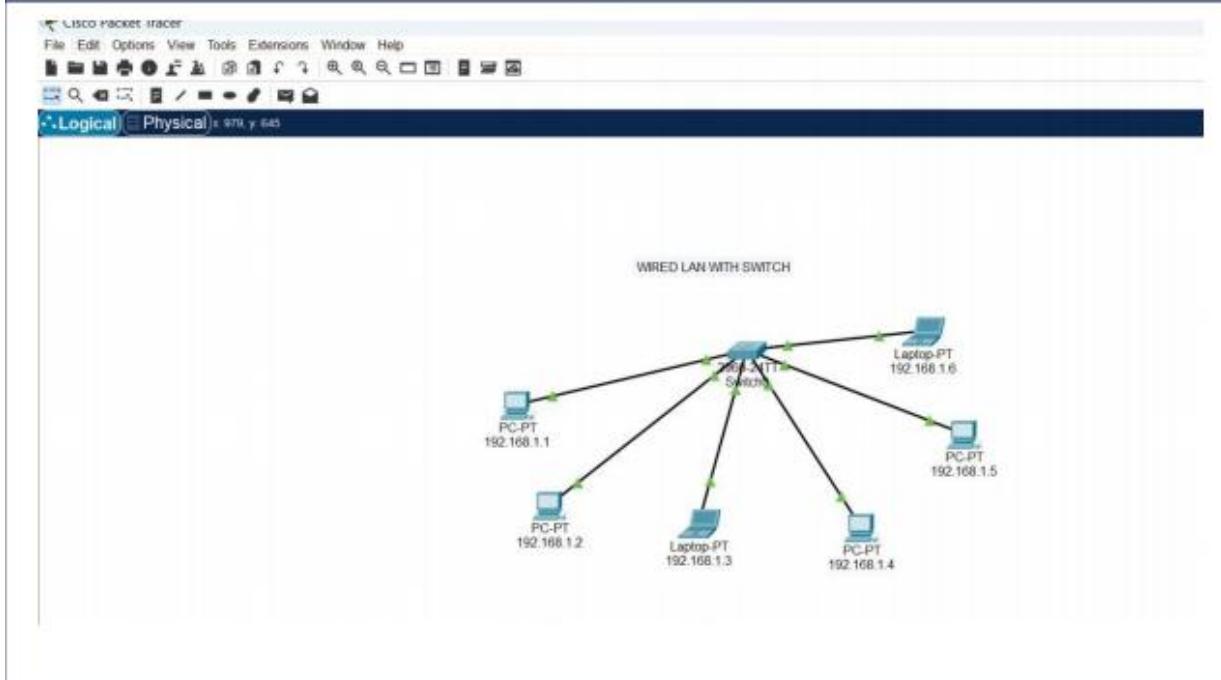
1. Click on a PC, go to the "Desktop" tab, and then "IP Configuration".
2. Assign a static IP address, subnet mask, and default gateway (if needed).
3. Repeat this process for each PC and server, ensuring each device has a unique IP address within the same network range.
4. For example, you might use:
 - PC1: 192.168.1.100, subnet mask 255.255.255.0, gateway 192.168.1.1
 - PC2: 192.168.1.101, subnet mask 255.255.255.0, gateway 192.168.1.1
 - Server: 192.168.1.102, subnet mask 255.255.255.0, gateway 192.168.1.1
5. The switch typically does not require an IP address for a simple LAN.

3. Test Connectivity:

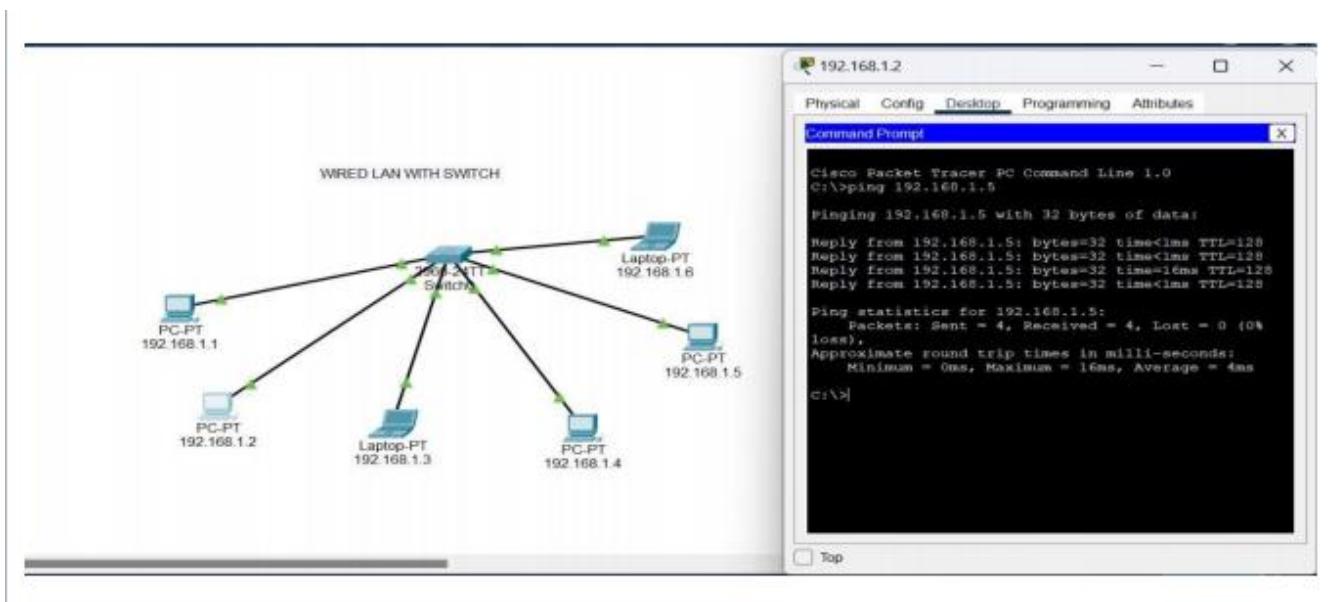
1. Click on a PC, go to the "Desktop" tab, and then "Command Prompt".
2. Use the ping command followed by the IP address of another device on the network to test communication.
3. For example, ping 192.168.1.101

MODEL OUTPUT:

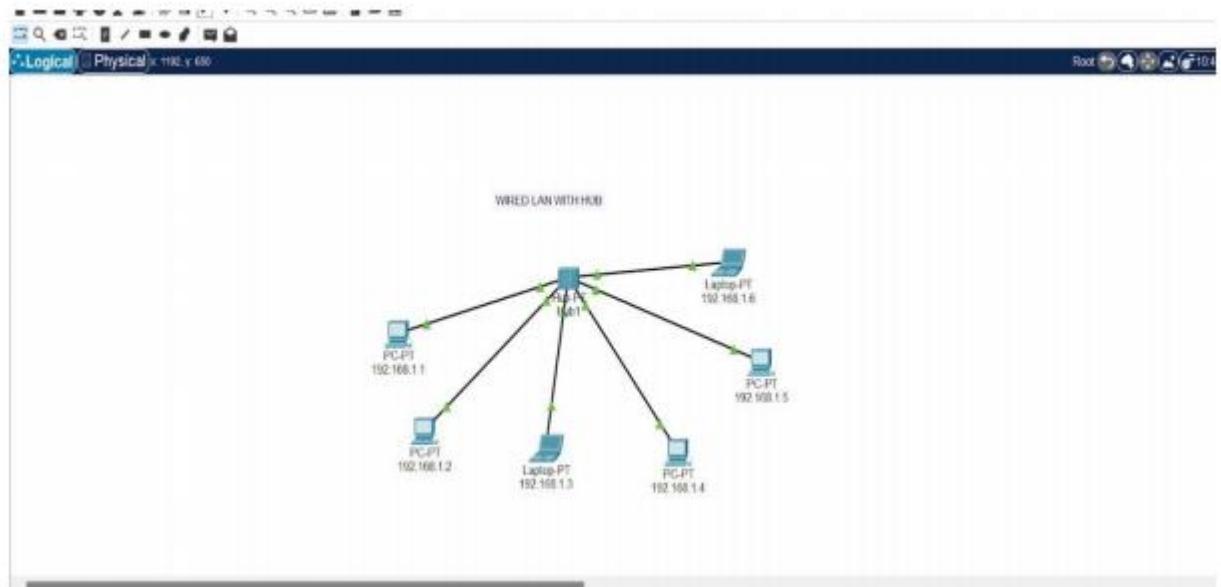
WIRED LAN USING SWITCH:



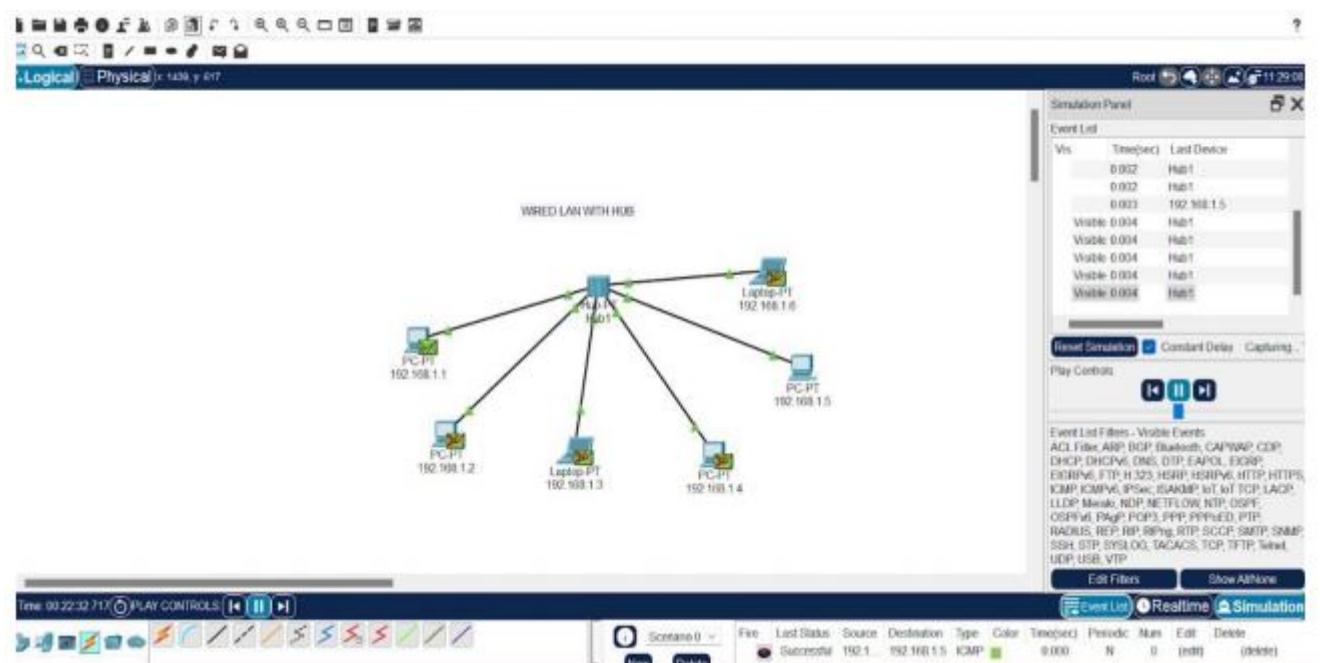
TEST THE CONNECTIVITY (Ping Command):



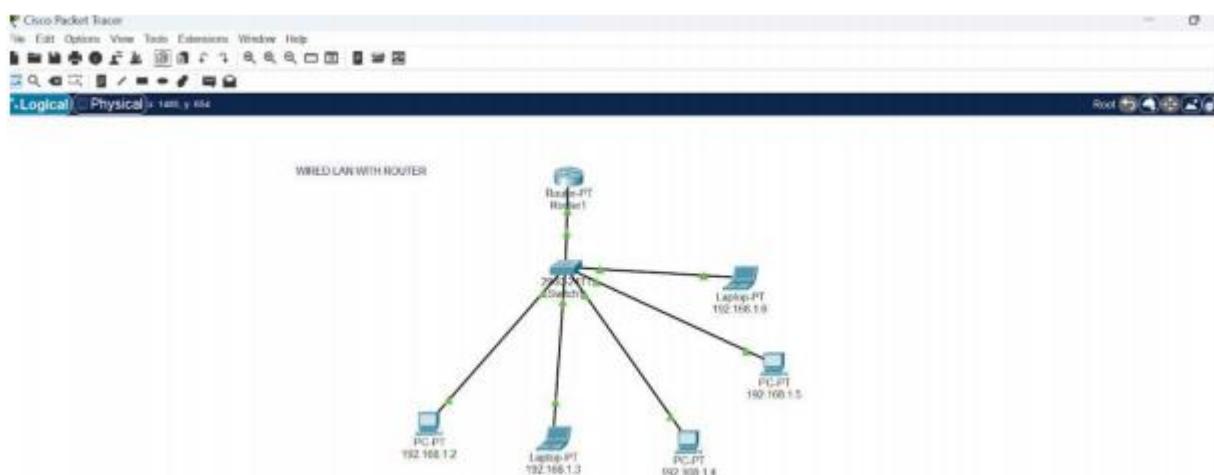
WIRED LAN USING HUB:



SIMULATION OF PACKET TRANSFER BETWEEN END USER:



WIRED LAN USING ROUTER:



CONFIGURATION OF ROUTER

Router0

Physical Config CLI Attributes

GLOBAL

- Settings
- Algorithm Settings
- ROUTING**
- Static
- RIP

INTERFACE

- FastEthernet0/0
- FastEthernet1/0
- Serial2/0
- Serial3/0
- FastEthernet4/0
- FastEthernet5/0

FastEthernet0/0

Port Status: On
Bandwidth: 100 Mbps 10 Mbps Auto
Duplex: Half Duplex Full Duplex Auto
MAC Address: 0060.5C4E.A178

IP Configuration

| | |
|--------------|-------------|
| IPv4 Address | 172.16.1.9 |
| Subnet Mask | 255.255.0.0 |

Tx Ring Limit: 10

Equivalent IOS Commands

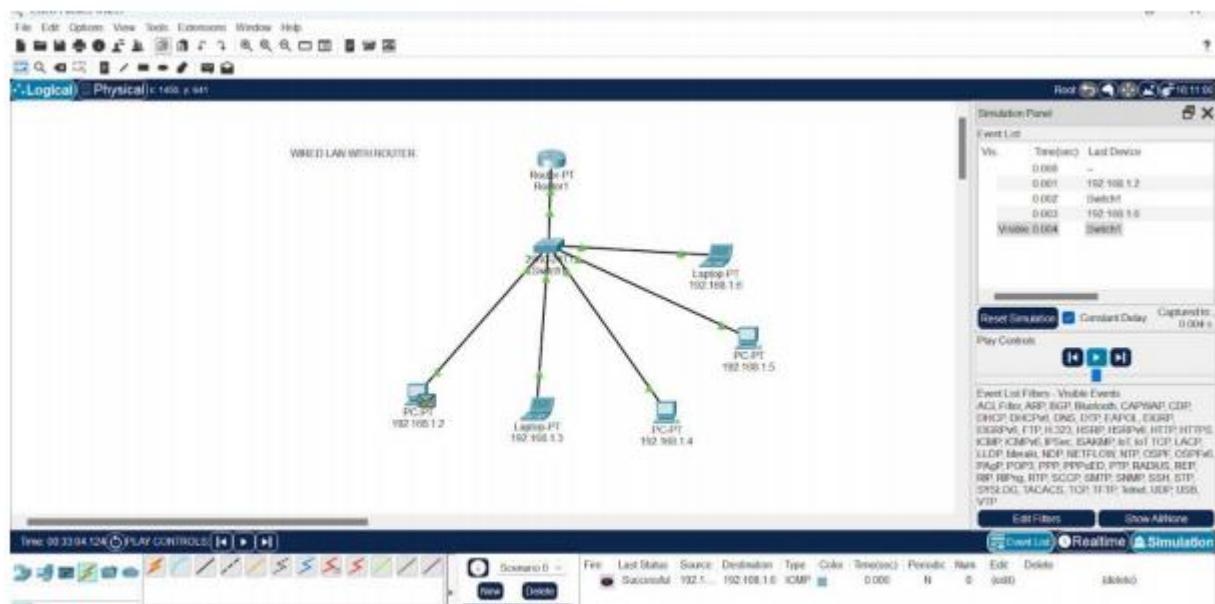
```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
ip address 172.16.1.9 255.255.0.0
Router(config-if)#ip address 172.16.1.9 255.255.0.0
Router(config-if)#

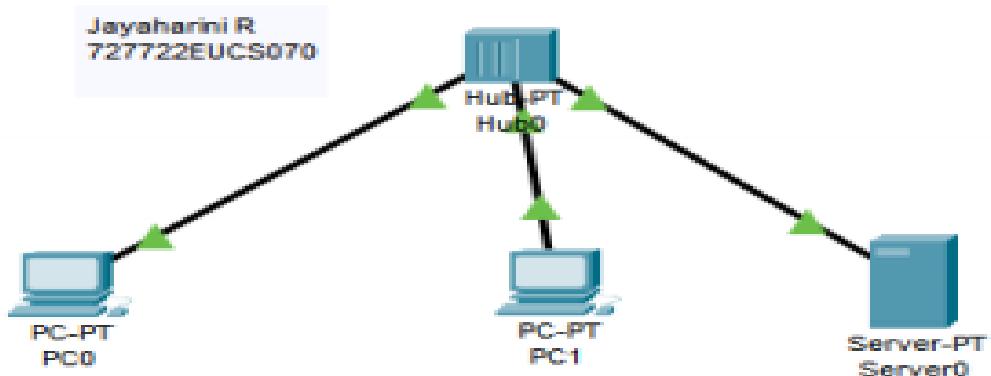
```

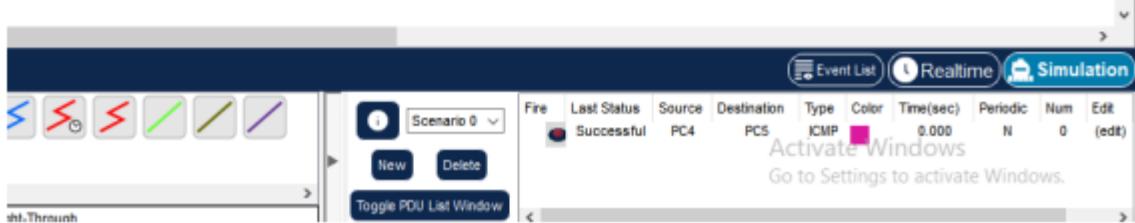
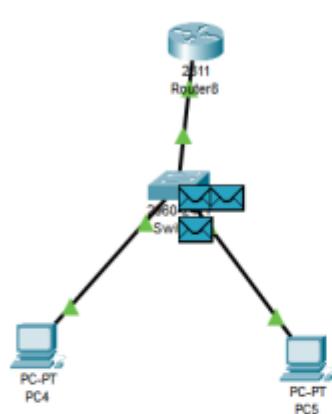
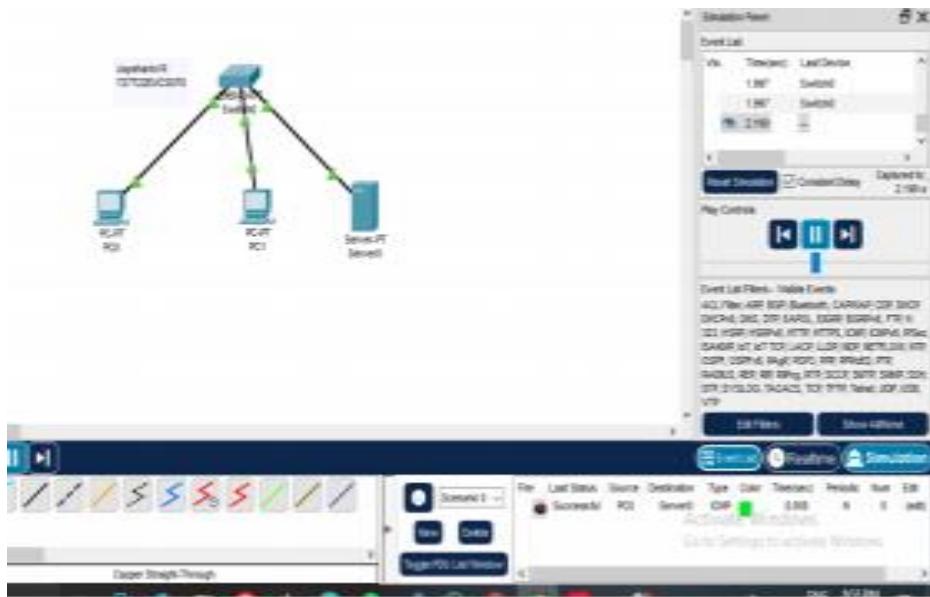
Top

SIMULATION OF PACKET TRANSFER THROUGH ROUTER:



EXPERIMENTAL OUTPUT:





Result:

Thus, simple wired LAN networks using hubs, switches and basic router configurations were implemented and communication between host devices were established and tested successfully.

Ex No: 2

Date:

CONFIGURE AND ANALYZE IPV4 AND IPV6 ADDRESSING SCHEMES AND SUBNETTING

AIM:

To build simple LANs, perform basic configurations for routers and switches, and implement IPv4 and IPv6 addressing schemes and Subnetting.

THEORY:

IPv4 (Internet Protocol version 4) is the most widely used version of the IP protocol, providing 32-bit addresses and supporting approximately 4.3 billion unique IP addresses. It uses a hierarchical addressing scheme consisting of a network address and a host address, separated by subnet masks to define network boundaries.

IPv6 (Internet Protocol version 6), on the other hand, was introduced to address the limitations of IPv4, offering a 128-bit address space, which provides a virtually limitless number of unique addresses. IPv6 also introduces improvements such as simplified header structures, improved security features, and better support for mobile networks. In this experiment, both addressing schemes are configured in a simple LAN, demonstrating how devices can communicate within the network and how these protocols coexist. IPv4 addresses are usually assigned using DHCP, while IPv6 uses **stateless address autoconfiguration (SLAAC)** or DHCPv6 for automatic address assignment. The setup allows for understanding how both protocols function, highlighting the importance of transitioning from IPv4 to IPv6 due to the growing demand for IP addresses in modern networks.

IPv4 (Internet Protocol Version 4)

- Format: IPv4 addresses are 32-bit numeric values, typically written in dotted-decimal notation (e.g., 192.168.1.1).
- Structure: It consists of four octets (8 bits each) separated by periods, with each octet ranging from 0 to 255.
- Address Space: IPv4 supports approximately 4.3 billion unique addresses.
- Example: 192.0.2.1
- Limitations: Due to rapid internet growth, IPv4 faces address exhaustion despite techniques like NAT (Network Address Translation).

IPv6 (Internet Protocol Version 6)

- Format: IPv6 addresses are 128-bit values, typically written in hexadecimal notation, separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

- Structure: It consists of eight groups of four hexadecimal digits, with leading zeros in groups often omitted for simplicity.
- Address Space: IPv6 provides a vast address space (approximately 340 undecillion addresses), effectively solving the exhaustion problem.
- Example: 2001:db8::ff00:42:8329 (using :: to compress consecutive zeros).
- Enhancements: IPv6 offers improved security, multicast addressing, simplified header structure, and better support for mobile devices.
- An IPv6 (normal) address has the format y:y:y:y:y:y:y:y, where y is called a *segment* and can be any hexadecimal value between 0 and FFFF. The segments are separated by colons, not periods. An IPv6 normal address must have eight segments; however, a short form notation can be used in the TS4500 management GUI for segments that are zero, or those that have leading zeros.
- The following are examples of valid IPv6 (normal) addresses:
2001:db8:3333:4444:5555:6666:7777:8888
- 2001:db8:3333:4444:CCCC:DDDD:EEEE:FFFF
- :: (implies all 8 segments are zero)
- 2001:db8:: (implies that the last six segments are zero)
- ::1234:5678 (implies that the first six segments are zero)
- 2001:db8::1234:5678 (implies that the middle four segments are zero)
- 2001:0db8:0001:0000:0000:0ab9:C0A8:0102 (This can be compressed to eliminate leading zeros, as follows: 2001:db8:1::ab9:C0A8:102)

PROCEDURE:

WIRED LAN

1. First, we will download Cisco Packet Tracer from netacad.com (latest version).
2. After downloading we will open it and now in this window, we see there are multiple small windows where we can select component and create our own particular computer network.
3. Select the components that are listed on the left bottom corner.
4. Select the 2950T switch and 2 routers from the components and place it on the white screen.
5. Place the PC's and laptops from the components and place it on the white screen.
6. Now select the wire from the connections and select copper straight through wire and connect fastethernet from PC to the switch.
7. Select serial connector for router to router connection.

CONFIGURING THE NETWORK

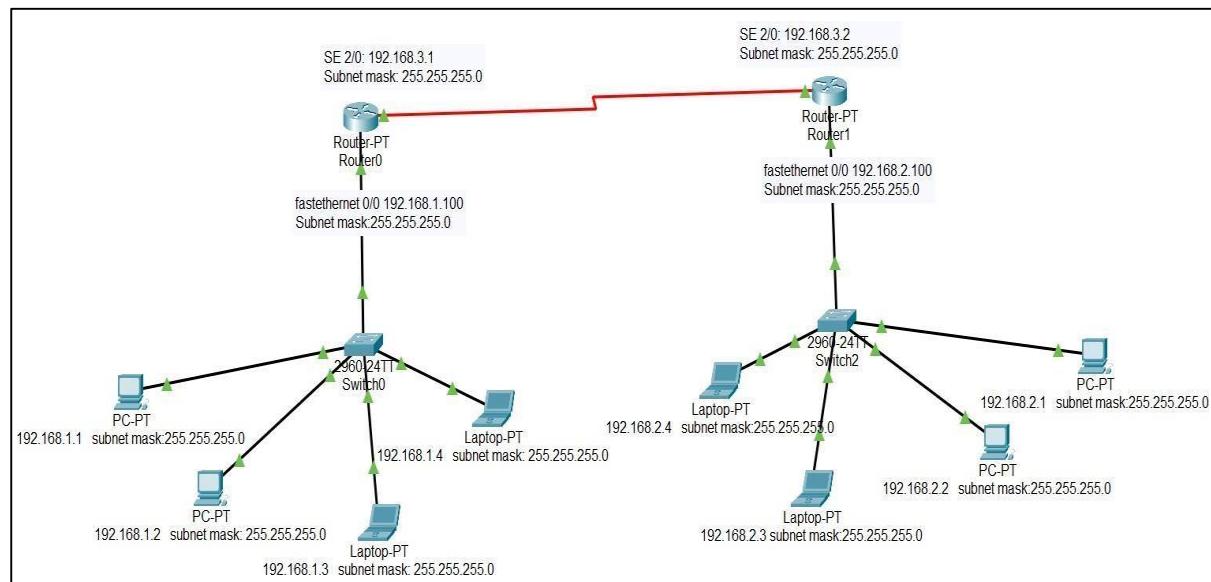
- Now assign ip address to each of the PC and laptops based on IPv4 or IPv6 formats.
- Under fastethernet tab when you double click on the PC you will able to see fastethernet and under that set IPv4 or IPv6 Address.

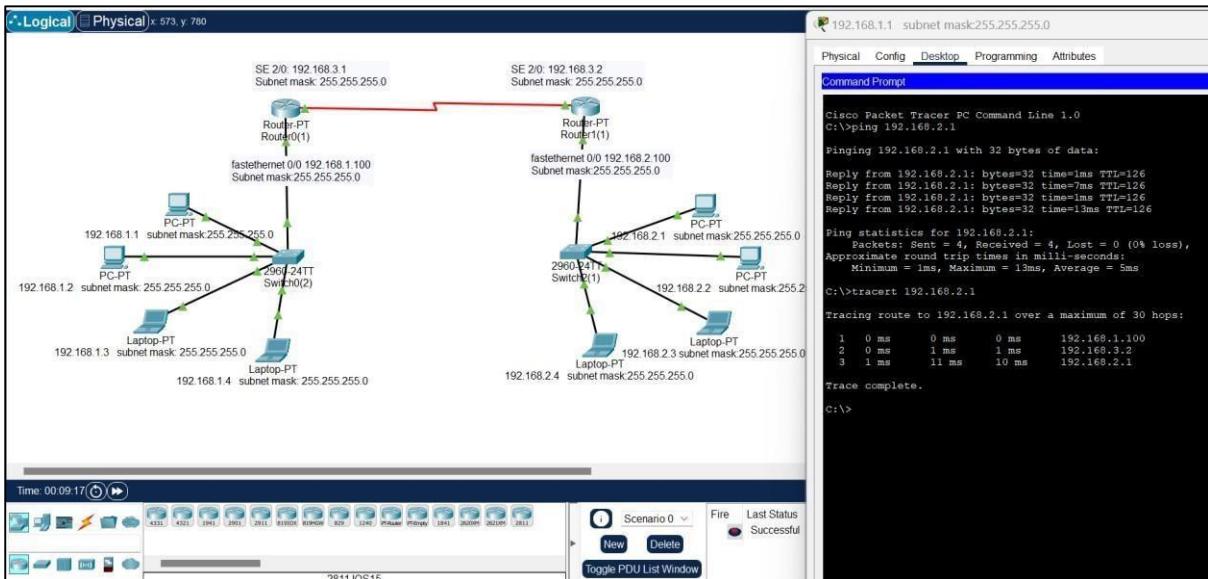
TESTING THE NETWORK

- Choose the device you want to test and double click on that and under desktop you will see the command prompt option
- Click on that and type the command ping “host ip”(the ip of any other device in the network).
- The data packets are successfully sent from the source to destination.

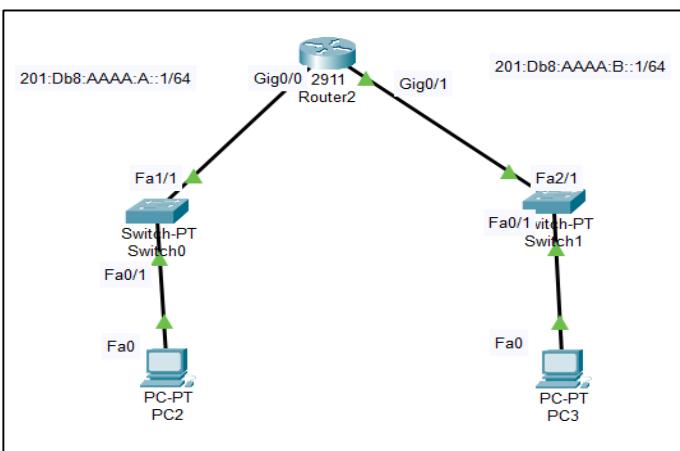
MODEL OUTPUT:

IPv4:





IPv6:



```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ipv6 unicast-routing
Router(config)#int Gig0/0
Router(config-if)#ipv6 address FE80::1 link
Router(config-if)#ipv6 address FE80::1 link-local
Router(config-if)#no shut

Router(config-if)#int Gig0/1
Router(config-if)#ipv6 address FE80::1 link-local
Router(config-if)#no shut

Router#en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int Gig0/0
Router(config-if)#ipv6 address 2001:DB8:AAAA:A::1/64
Router(config-if)#no shut
Router(config-if)#int Gig0/1
Router(config-if)#ipv6 address 2001:DB8:AAAA:B::1/64
Router(config-if)#no shut

```

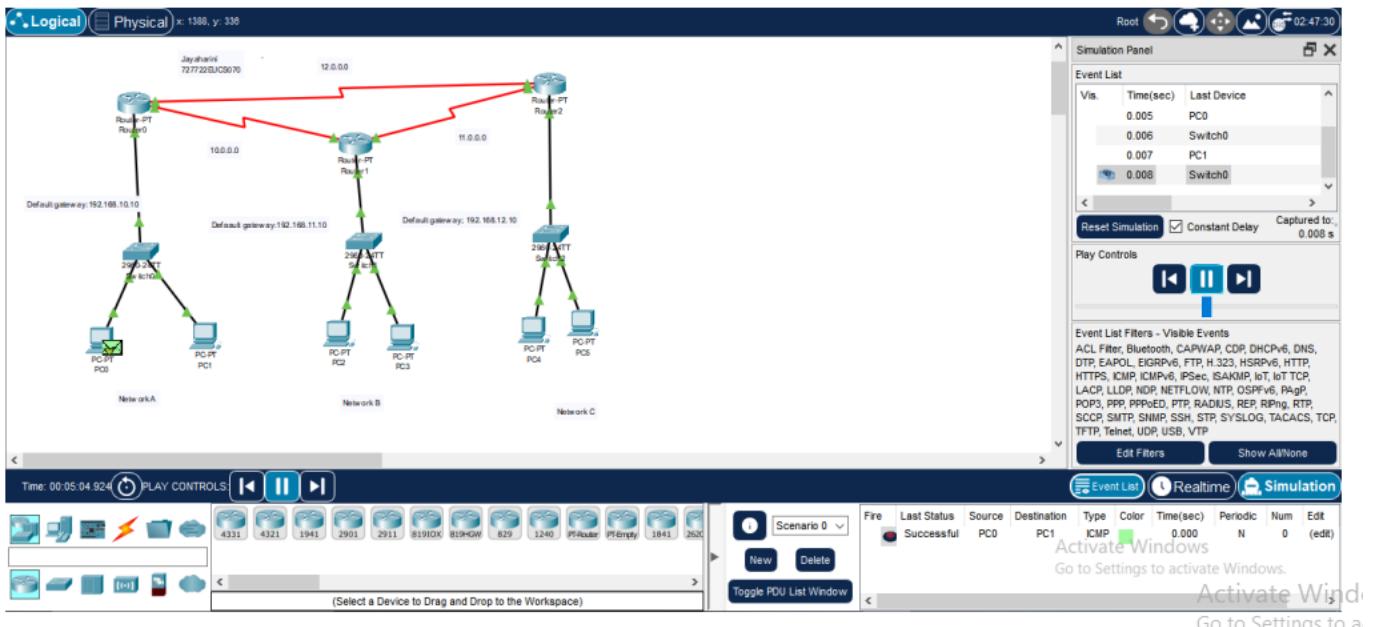
We have configured the router now change the settings of hosts in IPv6 configuration:
First, click on PC0 and go to desktop then IP configuration.

Now find the IPv6 configuration.

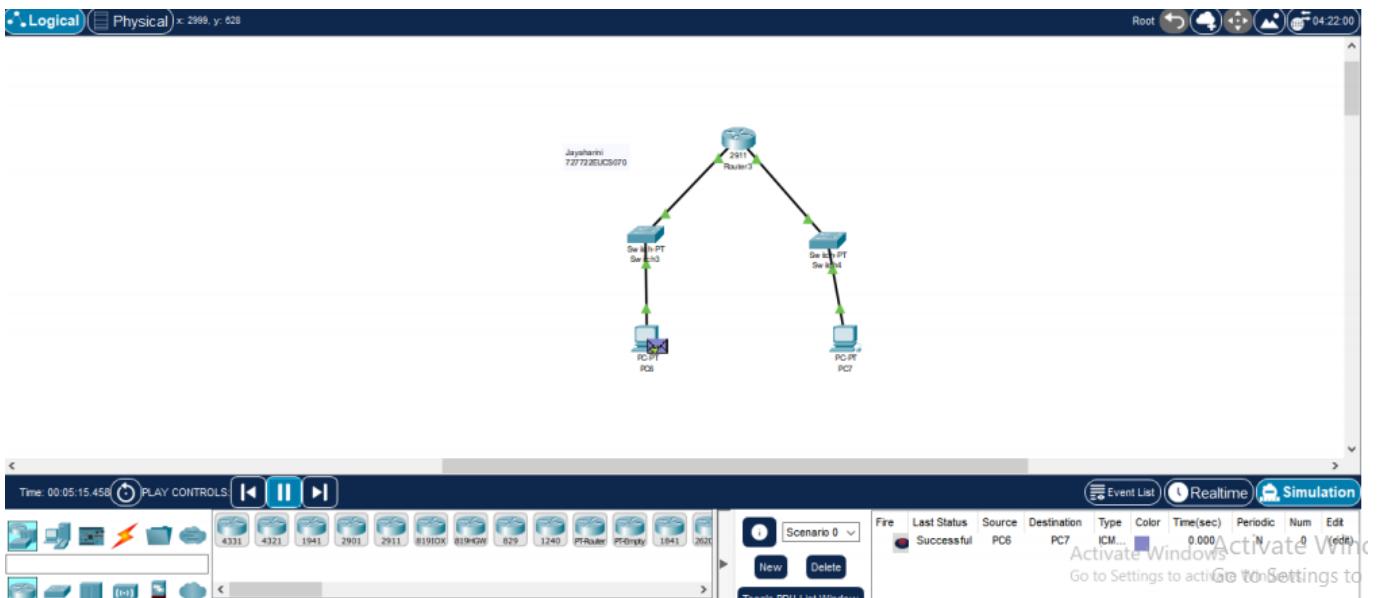
Change the settings from static to automatic and then after a few seconds. the IPv6 address and default

SIMULATION OUTPUT:

IPv4:



IPv6:



RESULT:

Thus, simple LAN networks, were implemented, where all the nodes were configured with both IPv4 and IPv6 addressing and communication between nodes was tested, in the Cisco Packet Tracer Simulation environment.

EX.NO:03

Date :

VLAN Configuration

Aim:

To configure Virtual Local Area Networks (VLANs) on switches and wireless access points in order to logically segment a network into smaller broadcast domains. This improves network performance, enhances security, simplifies management, and allows efficient utilization of network resources by grouping devices based on function, department, or application rather than physical location.

Theory: VLAN Configuration:

A Virtual Local Area Network (VLAN) allows logical segmentation of a network into different broadcast domains, regardless of physical location. Devices in the same VLAN can communicate directly, while devices in different VLANs require a Layer 3 device (Router or Layer 3 switch) for communication.

Router-on-a-Stick is a method of inter-VLAN routing in which a single router interface is divided into multiple sub-interfaces, each assigned to a VLAN. This allows communication between VLANs while keeping them logically separated.

Required Equipment:

1. Switch (Managed): To configure and manage VLANs.
2. Router (or Layer 3 Switch): For inter-VLAN routing if communication is needed between VLANs.
3. End Devices: PCs, laptops, servers, etc.
4. Ethernet Cables (Cat5e/Cat6): To connect devices.
5. Packet Tracer (Software): For simulation of VLAN configuration.

Procedure: VLAN Configuration in Cisco Packet Tracer

1. Build the Topology

1. Open Cisco Packet Tracer.
2. Drag and drop devices: PCs, a switch, and (if needed) a router for inter-VLAN routing.

3. Connect PCs to the switch using copper straight-through cables.

2. Create VLANs on the Switch

1. Click on the switch → go to CLI tab.
2. Enter global configuration mode and create VLANs:
3. Switch> enable
4. Switch# configure terminal
5. Switch(config)# vlan 10
6. Switch(config-vlan)# name Sales
7. Switch(config-vlan)# exit
8. Switch(config)# vlan 20
9. Switch(config-vlan)# name HR
10. Switch(config-vlan)# exit

3. Assign Ports to VLANs

1. Assign specific switch ports to VLANs (e.g., FastEthernet 0/1 to VLAN 10, FastEthernet 0/2 to VLAN 20):
2. Switch(config)# interface fastEthernet 0/1
3. Switch(config-if)# switchport mode access
4. Switch(config-if)# switchport access vlan 10
5. Switch(config-if)# exit
6. Switch(config)# interface fastEthernet 0/2
7. Switch(config-if)# switchport mode access
8. Switch(config-if)# switchport access vlan 20
9. Switch(config-if)# exit

4. Configure IP Addresses on PCs

1. Click on PC1 → Desktop → IP Configuration.

- Example for VLAN 10 (Sales):
 - IP: 192.168.10.2
 - Subnet: 255.255.255.0
 - Gateway: 192.168.10.1

2. On PC2 (VLAN 20 - HR):

- IP: 192.168.20.2
- Subnet: 255.255.255.0
- Gateway: 192.168.20.1

3. Repeat for other PCs according to their VLAN.

5. Configure Router-on-a-Stick (Inter-VLAN Routing)

1. Router> enable
2. Router# configure terminal
3. Router(config)# interface gigabitEthernet 0/0
4. Router(config-if)# no shutdown
5. Router(config)# interface gigabitEthernet 0/0.10
6. Router(config-subif)# encapsulation dot1Q 10
7. Router(config-subif)# ip address 192.168.10.1 255.255.255.0
8. Router(config-subif)# exit
9. Router(config)# interface gigabitEthernet 0/0.20
10. Router(config-subif)# encapsulation dot1Q 20
11. Router(config-subif)# ip address 192.168.20.1 255.255.255.0
12. Router(config-subif)# exit

6. Test Connectivity

1. Within the same VLAN:

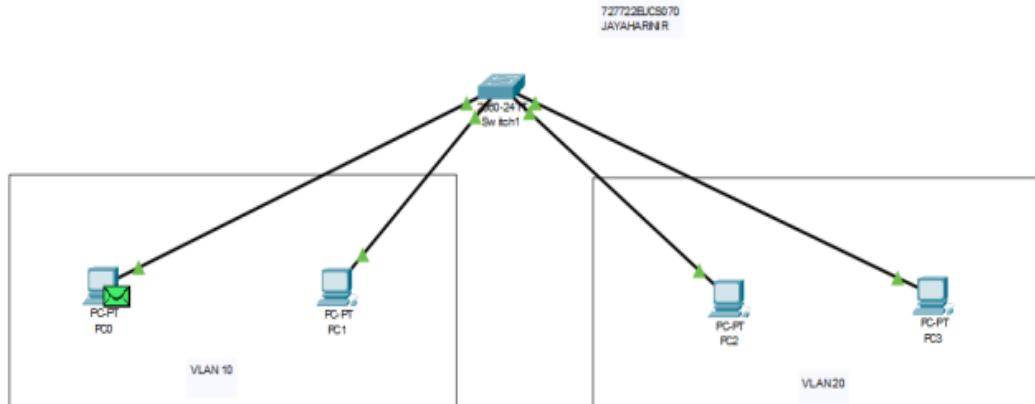
- Open Command Prompt on PC1 and ping another PC in VLAN 10 (e.g., PC3).
- Ping should succeed.

2. Between different VLANs:

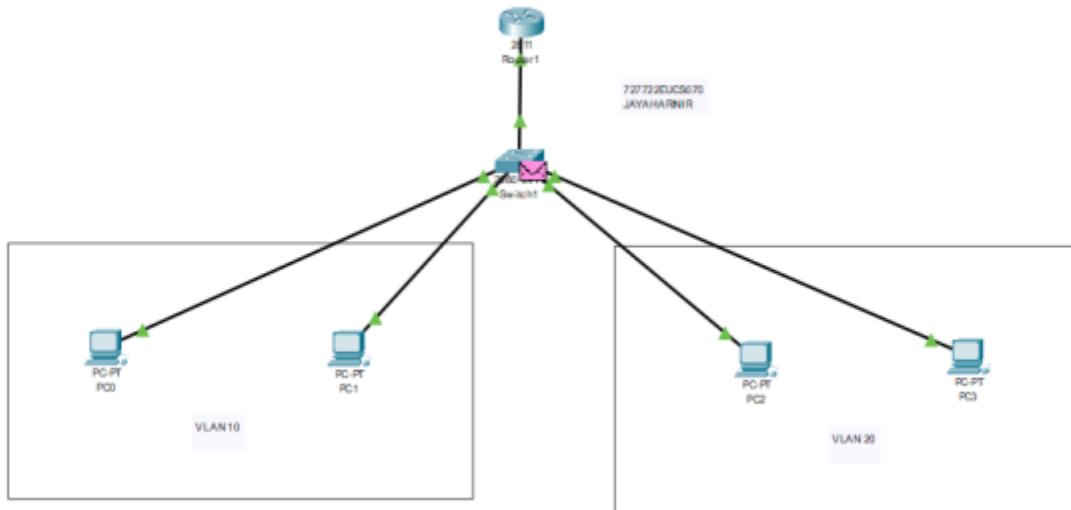
- If no router is configured, ping will fail (VLAN isolation).
- To allow inter-VLAN communication, configure a Router-on-a-Stick or a Layer 3 switch.

MODEL OUTPUT:

VLAN USING SWITCH ALONE:



VLAN CONIGURATION USING SWITCH AND ROUTER:



RESULTS:

VLANs were successfully configured, and with the help of the router, devices in different VLANs were able to communicate with each other.

EX.NO:04

Date :

WLAN Configuration

Aim:

To configure a Wireless Local Area Network (WLAN) using Access Points and Switch in Cisco Packet Tracer, enabling wireless devices (Laptop, Smartphone, Tablet) to connect and communicate with each other and with a server.

Theory: WLAN Configuration:

A **Wireless Local Area Network (WLAN)** allows devices to connect without physical cables using wireless signals. In this topology:

1. **Access Points (APs):** Provide wireless connectivity for end devices (Laptop, Smartphone, Tablet).
2. **Switch:** Connects all APs and the Server, forming the backbone of the network.
3. **Server:** Provides services like DHCP, HTTP, or DNS for wireless clients.
4. **End Devices:** Connect wirelessly to APs via SSID.

WLANs improve mobility, flexibility, and scalability compared to wired LANs.

Required Equipment:

1. **Switch (2960):** Central device connecting APs and server.
2. **Access Points (APs):** Provide wireless connectivity to end devices.
3. **Server:** Provides network services.
4. **End Devices:** Laptop, Smartphone, Tablet (with wireless adapters).
5. **Ethernet Cables (Cat5e/Cat6):** To connect APs and server to the switch.
6. **Cisco Packet Tracer Software.**

Procedure: WLAN Configuration in Cisco Packet Tracer

1. Build the Topology

1. Place a Switch, Server, Access Points (APs), and Wireless End Devices (Laptop, Smartphone, Tablet).
2. Connect APs to the Switch using copper straight-through cables.
3. Connect Server to the Switch using a copper straight-through cable.

2. Configure the Access Points

1. Click on Access Point0 → Config tab.
 - a) Set SSID: Campus-WLAN
 - b) Enable DHCP (optional) or leave to server/router.
 - c) Set Security: WPA2-PSK with password (e.g., 12345).
2. Repeat the same for AP1 and AP2 (using same SSID if you want a unified WLAN).

3. Configure the Server

1. Click on Server0 → go to Config tab → select Services.
2. Enable DHCP service and set IP pool:
 - a) Default Gateway: 192.168.1.1
 - b) Subnet Mask: 255.255.255.0
 - c) Start IP: 192.168.1.10
 - d) Maximum Users: 50
3. Set the Server's static IP as 192.168.1.1 with subnet 255.255.255.0.

4. Configure Wireless End Devices

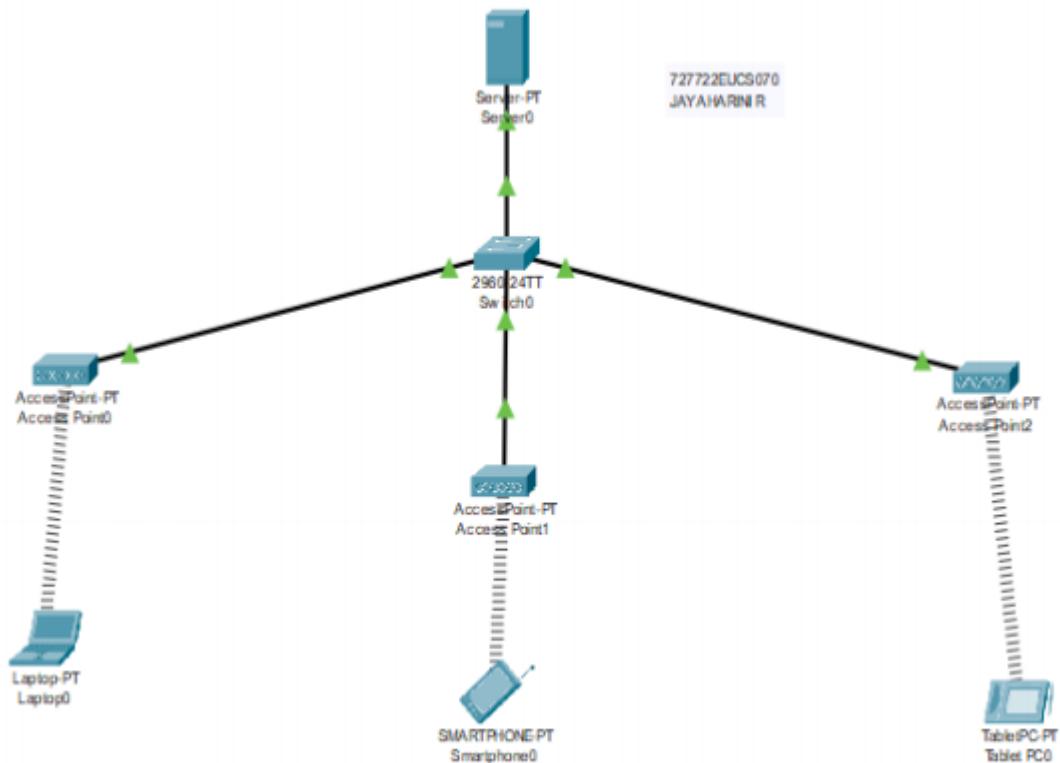
1. On Laptop0 → Desktop → PC Wireless → Connect to Campus-WLAN SSID → enter WPA2 password.
2. On Smartphone0 and Tablet0 → enable wireless adapter → connect to Campus-WLAN.
3. Devices will automatically obtain IPs from the server via DHCP.

5. Test Connectivity

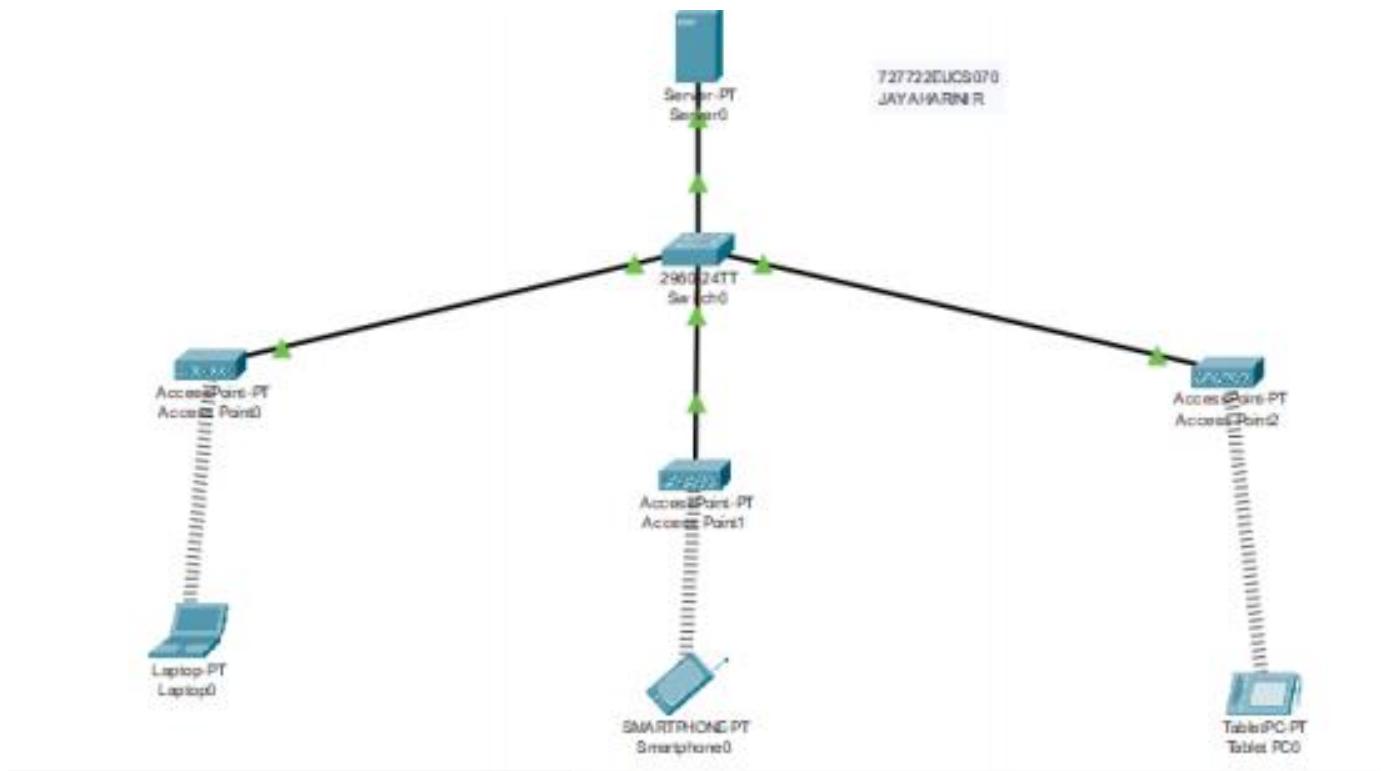
1. On Laptop0 → Command Prompt → ping 192.168.1.1 (Server IP) → Reply should be received.
2. On Smartphone0 → ping Tablet0's IP → Reply should be received.
3. Communication between all WLAN devices and the server should be successful.

MODEL OUTPUT:

WLAN USING SWITCH AND SERVER :



WLAN:



RESULTS:

The WLAN was successfully configured using access points and a switch. Wireless devices (Laptop, Smartphone, Tablet) were able to connect to the WLAN using the configured SSID and security key. Communication between wireless devices and the server was tested successfully using ping, verifying proper WLAN connectivity.

EX.NO:05(A)

Date:

DHCP Relay Configuration | IP Helper Address

Aim:

To configure DHCP relay (IP Helper Address) on a router so that hosts in different VLANs/subnets can obtain IP addresses dynamically from a centralized DHCP server.

Theory:

- DHCP uses **broadcast messages** to discover a DHCP server (Discover/Offer/Request/Ack).
- Broadcasts do **not cross routers**, so hosts in different VLANs cannot reach a central DHCP server directly.
- To solve this, we configure the router interface with **IP helper-address <DHCP-server-IP>**.
- The router will **relay** DHCP broadcast requests from clients to the DHCP server as **unicast** messages.
- The DHCP server then responds, and the router forwards the reply back to the clients.

Required Equipment:

1. Cisco Router (2811)
2. Cisco Switch (2960)
3. DHCP Server (Packet Tracer Server)
4. End Devices – PCs / IP Phones
5. Copper Straight-through cables
6. Cisco Packet Tracer software

Procedure:

Step 1: Build the Topology

- Connect **Router ↔ Switch ↔ PCs/Phones**.
- Connect **Server** to the switch (this will act as DHCP server).
- Assign PCs/Phones to VLANs (optional, if using VLAN-based DHCP).

Step 2: Configure the DHCP Server

On Server1:

1. Go to **Desktop** → **IP Configuration**
 - o IP: 192.168.10.10
 - o Subnet: 255.255.255.0
 - o Gateway: 192.168.10.1
 2. Go to **Services** → **DHCP**
 - o Enable DHCP
 - o Pool Name: VLAN10
 - o Default Gateway: 192.168.10.1
 - o Start IP: 192.168.10.100
 - o Subnet Mask: 255.255.255.0
 - o DNS Server: (optional)
- Step 3: Configure Router Interfaces**
- Example for two VLANs:
- ```
Router> enable
Router# configure terminal
!
! Configure VLAN 10 sub-interface
Router(config)# interface g0/0.10
Router(config-subif)# encapsulation dot1Q 10
Router(config-subif)# ip address 192.168.10.1 255.255.255.0
Router(config-subif)# ip helper-address 192.168.10.10
!
! Configure VLAN 20 sub-interface
Router(config)# interface g0/0.20
Router(config-subif)# encapsulation dot1Q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
Router(config-subif)# ip helper-address 192.168.10.10
!
Router(config)# exit
```

```
Router(config)# interface g0/0
```

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

- ip helper-address 192.168.10.10 → tells the router to forward DHCP requests to the DHCP server.

#### **Step 4: Configure Switch Ports**

```
Switch> enable
```

```
Switch# configure terminal
```

```
Switch(config)# interface fa0/1
```

```
Switch(config-if)# switchport mode access
```

```
Switch(config-if)# switchport access vlan 10
```

```
!
```

```
Switch(config)# interface fa0/2
```

```
Switch(config-if)# switchport mode access
```

```
Switch(config-if)# switchport access vlan 20
```

#### **Step 5: Configure Clients (PCs / IP Phones)**

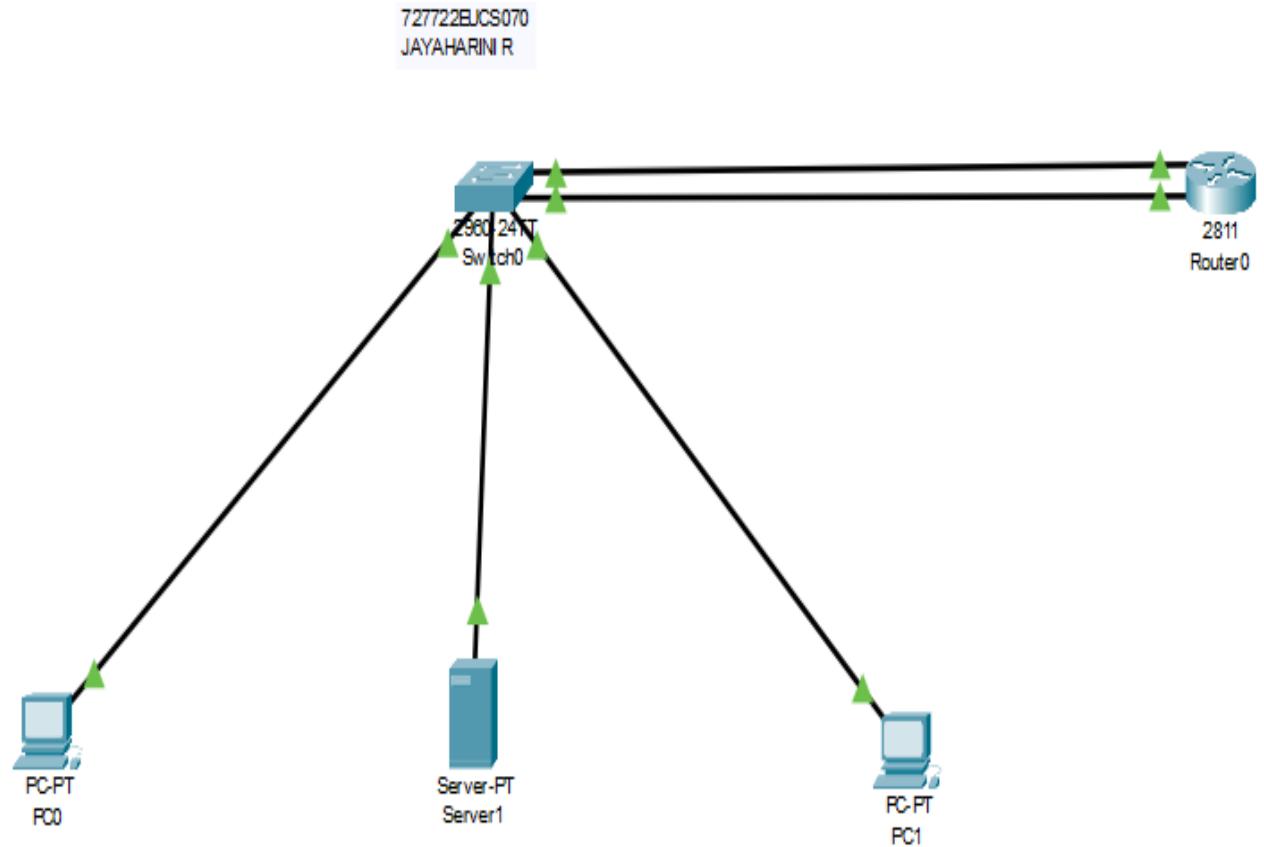
- On PC0 / PC1 / IP Phone:
  - Go to Desktop → IP Configuration
  - Select **DHCP**
  - The device should automatically receive IP, Subnet, Gateway from the DHCP server.

#### **Testing:**

1. On PC0 (VLAN 10), check ipconfig → should get an IP from 192.168.10.100+ range.
2. On PC1 (VLAN 20), check → should get an IP from 192.168.20.100+.
3. Ping the router gateway and server → should succeed.
4. Cross-VLAN communication works if Router-on-a-Stick is enabled.

## Model Output:

### DHCP Relay Configuration | IP Helper Address



## Result:

DHCP relay was successfully configured using **IP Helper Address**, enabling clients in different VLANs/subnets to obtain IP addresses dynamically from a centralized DHCP server.

**EX.NO:05(B)**

**Date :**

## **Configuring VoIP Phones using Cisco Packet Tracer (IP Telephony)**

**Aim:**

To configure IP Telephony in Cisco Packet Tracer using VoIP phones and a Call Manager router. The objective is to enable voice communication between IP phones connected in a LAN.

**Theory:**

Voice over Internet Protocol (VoIP) allows transmission of voice data over IP networks. In Cisco Packet Tracer, VoIP can be simulated using IP phones connected to a switch and configured through a router with Call Manager Express (CME) features enabled.

- **Call Manager Express (CME):** Runs on a Cisco router and provides call processing to register and manage IP phones.
- **DHCP Service:** The router can act as a DHCP server to dynamically assign IP addresses to the IP phones.
- **Telephony-service:** A special service that assigns extension numbers to phones.
- **Dialing:** Once IP phones are registered, users can dial extension numbers to communicate.

**Required Equipment:**

1. Cisco Router (2811/2911 with Telephony support)
2. Cisco Switch (2960/2950)
3. IP Phones (e.g., 7960/7970 series)
4. PCs (optional for management)
5. Ethernet Cables (Copper Straight-Through)
6. Cisco Packet Tracer Software

**Procedure:**

### **Step 1: Build the Topology**

1. Drag and drop one **Router (2811)**, one **Switch (2960)**, and two **IP Phones (7960)**.
2. Connect the devices using Copper Straight-Through cables.

### **Step 2: Configure the Router for Telephony Service**

1. Go to Router → CLI.

2. Enable the telephony service:
3. Router> enable
4. Router# configure terminal
5. Router(config)# telephony-service
6. Router(config-telephony)# max-dn 5
7. Router(config-telephony)# max-ephones 5
8. Router(config-telephony)# ip source-address 192.168.1.1 port 2000
9. Router(config-telephony)# auto assign 1 to 5
10. Router(config-telephony)# exit

### **Step 3: Configure Ephone-dn (Extension Numbers)**

```
Router(config)# ephone-dn 1
Router(config-ephone-dn)# number 101
Router(config-ephone-dn)# exit
```

```
Router(config)# ephone-dn 2
Router(config-ephone-dn)# number 102
Router(config-ephone-dn)# exit
```

### **Step 4: Configure IP Addresses and DHCP on Router**

```
Router(config)# interface fastEthernet 0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
```

```
Router(config)# ip dhcp pool VOIP
Router(dhcp-config)# network 192.168.1.0 255.255.255.0
Router(dhcp-config)# default-router 192.168.1.1
Router(dhcp-config)# option 150 ip 192.168.1.1
Router(dhcp-config)# exit
```

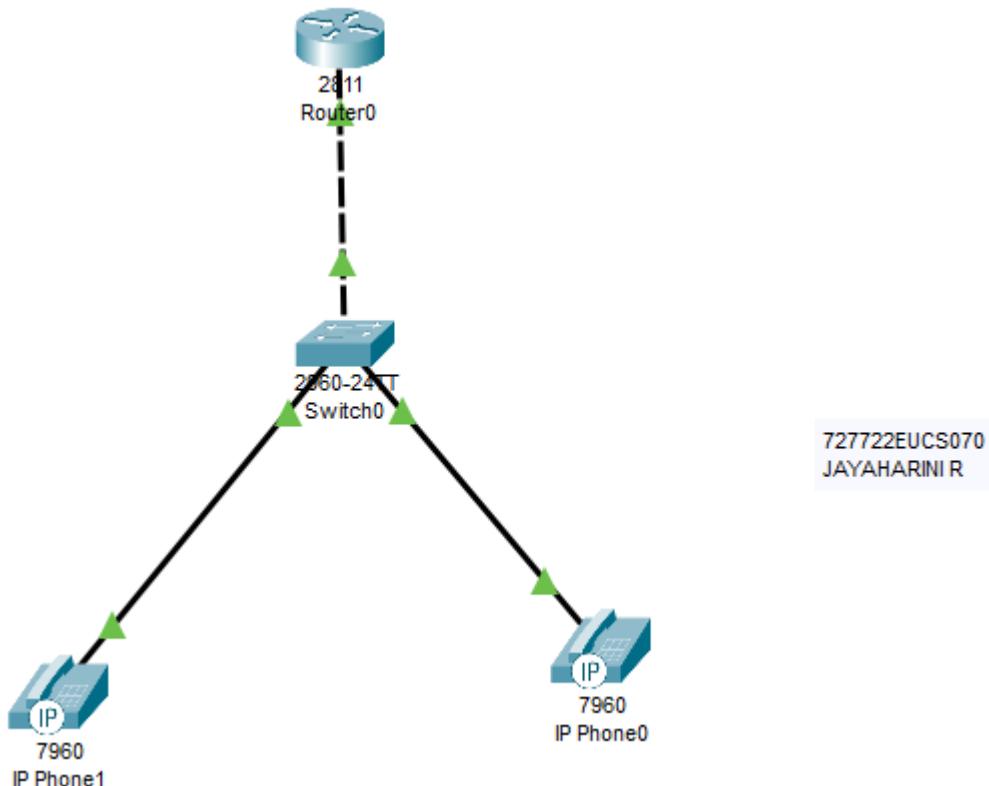
## Step 5: Configure IP Phones

1. Connect IP Phones to Switch ports.
2. By default, they will request an IP address via DHCP.
3. Phones should automatically register with the router and display assigned extensions (101, 102).

## Step 6: Test Calling Between Phones

1. Pick up **IP Phone0** and dial 102.
2. **IP Phone1** should ring.
3. Similarly, dial 101 from IP Phone1 to call IP Phone0.

## Model Output:



## Configuring VoIP Phones

## Result:

VoIP Phones were successfully configured using Cisco Packet Tracer. The router acted as a Call Manager Express, and the IP phones registered with assigned extensions, enabling voice communication between them.

## **EX.NO:06**

### **Date:**

## **Secure WLAN Setup using 802.1X Authentication with RADIUS**

### **Aim:**

To configure a secure Wireless LAN (WLAN) using **802.1X authentication** with a **RADIUS server**, ensuring only authenticated users can connect to the wireless network.

### **Theory:**

- **802.1X** is a port-based access control protocol that provides authentication before allowing devices onto the network.
- **RADIUS (Remote Authentication Dial-In User Service)** is used to centralize authentication, authorization, and accounting.
- Process Flow:
  1. The **client (Laptop, Smartphone, Tablet)** requests access through the Access Point (AP).
  2. The **AP (Authenticator)** forwards credentials to the **RADIUS server**.
  3. The **RADIUS server (Authentication Server)** verifies credentials from its database.
  4. If valid, the client is granted access to the WLAN; otherwise, denied.

This ensures only authorized users can connect, providing better **security** compared to WPA2-PSK.

### **Required Equipment:**

1. Cisco Switch (2960)
2. Wireless Access Points
3. End Devices – Laptop, Smartphone, Tablet
4. Server (Configured as RADIUS server)
5. Copper Straight-through cables
6. Cisco Packet Tracer software

### **Procedure:**

#### **Step 1: Build the Topology**

- Connect **Access Points** → **Switch** → **RADIUS Server**.
- Connect **Laptop, Smartphone, Tablet** wirelessly to APs.
- Configure the **Server0** as RADIUS server.

### **Step 2: Configure the RADIUS Server (Server0)**

1. Go to **Services** → **AAA**
  - Enable AAA.
  - Configure a RADIUS username and password.
  - Example:
    - Username: student
    - Password: cisco123
2. Enable RADIUS Authentication service.

### **Step 3: Configure the Access Points**

On each **Access Point**:

- Go to **Config** → **Wireless Settings**
  - SSID: SecureWiFi
  - Security: **WPA2-Enterprise** (802.1X)
  - RADIUS Server: 192.168.10.10 (Server0's IP)
  - Shared Secret: radius123

### **Step 4: Configure the Switch (if required for VLAN assignment)**

Switch> enable

Switch# configure terminal

Switch(config)# interface fa0/1

Switch(config-if)# switchport mode access

Switch(config-if)# switchport access vlan 10

Switch(config)# interface fa0/2

Switch(config-if)# switchport mode access

Switch(config-if)# switchport access vlan 20

### **Step 5: Configure Client Devices (Laptop, Smartphone, Tablet)**

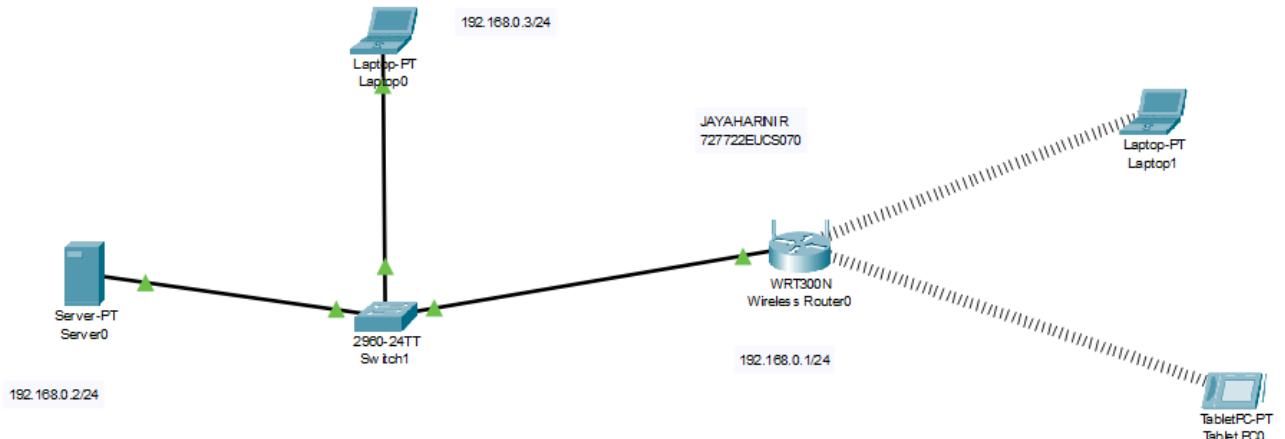
- Go to **Desktop** → **Wireless Settings**

- Select SSID: SecureWiFi
- Authentication: **WPA2-Enterprise**
- Enter Username: student
- Enter Password: cisco123

### Testing:

1. On **Laptop, Smartphone, Tablet** → connect to SecureWiFi.
2. The AP forwards authentication request to the **RADIUS server**.
3. If username/password matches, device gets IP and network access.
4. Try accessing/pinging the server → should succeed.
5. If wrong credentials are entered → access is denied.

### MODEL OUTPUT:



### Result:

The WLAN was successfully secured using **802.1X authentication with RADIUS**, ensuring only authenticated clients could access the network.

**EX.NO:07**

**Date :**

## **OSPF Configuration**

**Aim:**

To configure and analyze the performance of the OSPF routing protocol in Cisco Packet Tracer using a multi-router topology, enabling communication between two networks (192.168.1.0 and 155.165.1.0) through dynamic routing.

### **Theory: OSPF**

OSPF is a link-state Interior Gateway Protocol (IGP) that uses Dijkstra's Shortest Path First (SPF) algorithm to compute the best routes. It exchanges Link-State Advertisements (LSAs) to build a complete network topology, ensuring quick convergence and scalability.

### **Key Points:**

1. **Dynamic Routing:** Automatically updates routing tables if topology changes.
2. **Metric:** Selects paths based on cost (inversely proportional to bandwidth).
3. **Areas:** Supports hierarchical routing with Area 0 as the backbone.
4. **Fast Convergence:** Ensures reliable and quick recovery after failures.

### **Required Equipment:**

1. 3 Routers (Router0, Router1, Router2).
2. 2 Switches (Switch0, Switch1).
3. 2 PCs (PC0, PC1).
4. Serial/FastEthernet connections between routers.
5. Cisco Packet Tracer Software.

## **Procedure: OSPF Configuration in Cisco Packet Tracer**

### **1. Build the Topology**

- Connect Router0 ↔ Router2 ↔ Router1 using Serial links.
- Add a direct link between Router0 and Router1.
- Connect PC0 to Router0 via Switch0 (Network A: 192.168.1.0).
- Connect PC1 to Router1 via Switch1 (Network B: 155.165.1.0).

### **2. Assign IP Addresses**

- **Network A:** PC0 = 192.168.1.2 /24, Gateway (Router0) = 192.168.1.1.
- **Network B:** PC1 = 155.165.1.2 /24, Gateway (Router1) = 155.165.1.1.
- Inter-router links:
  - Router0–Router2: 10.0.0.0/30.
  - Router2–Router1: 30.0.0.0/30.
  - Router0–Router1: 20.0.0.0/30.

### **3. Configure OSPF on Routers**

On each router, enable OSPF process ID 1 and advertise connected networks.

Example for Router0:

```
Router0(config)# router ospf 1
```

```
Router0(config-router)# network 192.168.1.0 0.0.0.255 area 0
```

```
Router0(config-router)# network 10.0.0.0 0.0.0.3 area 0
```

```
Router0(config-router)# network 20.0.0.0 0.0.0.3 area 0
```

Similarly configure Router1 and Router2 for their networks.

### **4. Verify OSPF**

- Check neighbor relationships:

```
Router0# show ip ospf neighbor
```

- Check routing table:

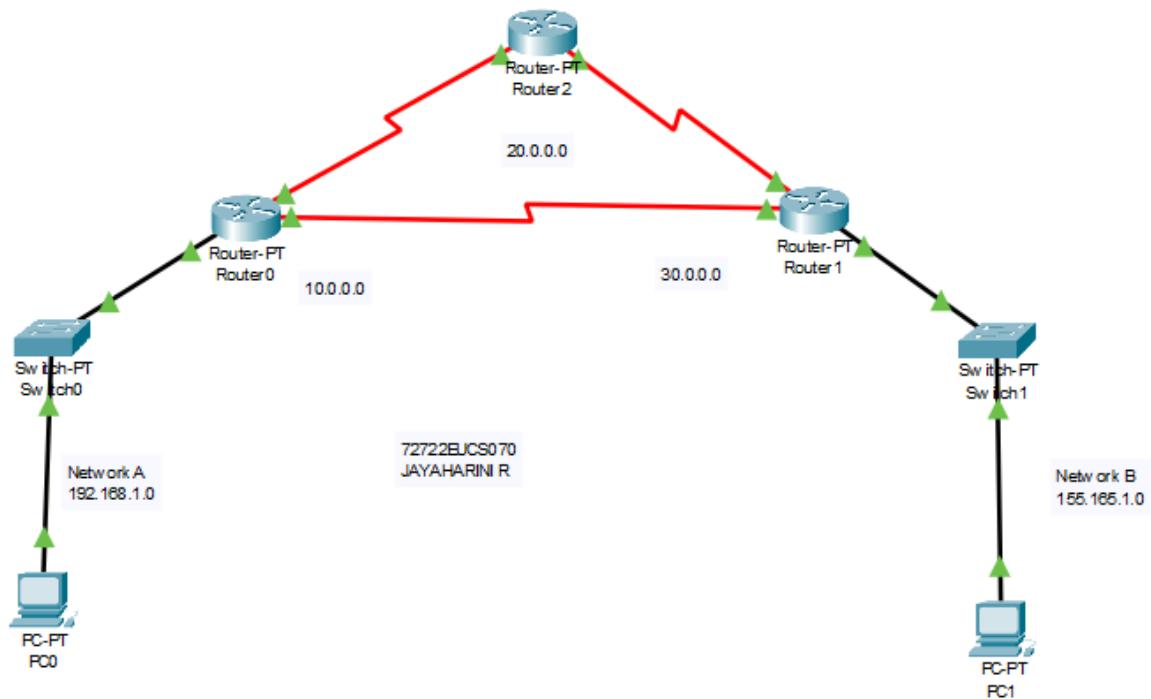
```
Router1# show ip route
```

- Routes learned via OSPF are marked with O.

## 5. Test Connectivity

- From PC0 → ping PC1 (192.168.1.2 → 155.165.1.2).
- Verify replies are received.
- Check alternate paths by shutting down one router link and observing OSPF rerouting.

## MODEL OUTPUT:



## Result:

The OSPF routing protocol was successfully configured and simulated. Routers dynamically exchanged link-state information, built routing tables, and ensured communication between Network A (192.168.1.0) and Network B (155.165.1.0). The simulation also confirmed OSPF's ability to reroute traffic through alternate paths, verifying its efficiency and reliability.

**EX.NO:08**

**Date :**

## **WAN Configuration**

**Aim:**

To simulate and configure a Wide Area Network (WAN) using Cisco Packet Tracer by interconnecting multiple LANs through routers, and verify communication between end devices across geographically distributed networks.

### **Theory: WAN (Wide Area Network)**

A **Wide Area Network (WAN)** interconnects multiple Local Area Networks (LANs) across large geographical distances using routers and communication links such as leased lines, DSL, or serial links.

### **Key Features of WANs:**

1. **Geographic Coverage:** Connects networks across cities, states, or even globally.
2. **Routers as Backbone:** Routers provide routing between LANs in different locations.
3. **WAN Links:** Serial connections, Frame Relay, MPLS, or modern broadband are used.
4. **Protocols:** Static Routing, Dynamic Routing (RIP, OSPF, EIGRP, BGP) can be used for WAN communication.
5. **Applications:** Used by enterprises, ISPs, and institutions for inter-branch connectivity.

### **Required Equipment:**

1. **Routers (e.g., Cisco 2911 / 2811):** Connect LANs over WAN.
2. **Switches:** Provide LAN connectivity in each branch.
3. **End Devices (PCs):** For communication testing.

4. **Serial DCE/DTE Connections or Copper Cables:** To simulate WAN links.
5. **Cisco Packet Tracer Software.**

## **Procedure: WAN Configuration in Cisco Packet Tracer**

### **1. Build the Topology**

- Place 2 routers (Router0 and Router1).
- Connect them using a **Serial DCE/DTE link** to simulate a WAN.
- Attach a switch and PC to each router to form two separate LANs:
  - **LAN A:** Network 192.168.1.0/24.
  - **LAN B:** Network 192.168.2.0/24.

### **2. Assign IP Addresses**

- Router0 (LAN A interface): 192.168.1.1/24.
- PC0 (LAN A): 192.168.1.2/24, Gateway = 192.168.1.1.
- Router1 (LAN B interface): 192.168.2.1/24.
- PC1 (LAN B): 192.168.2.2/24, Gateway = 192.168.2.1.
- Serial link (WAN):
  - Router0 Serial0/0/0 = 10.0.0.1/30.
  - Router1 Serial0/0/0 = 10.0.0.2/30.

### **3. Configure Routing (Static Routing Example)**

On Router0:

```
Router0(config)# ip route 192.168.2.0 255.255.255.0 10.0.0.2
```

On Router1:

```
Router1(config)# ip route 192.168.1.0 255.255.255.0 10.0.0.1
```

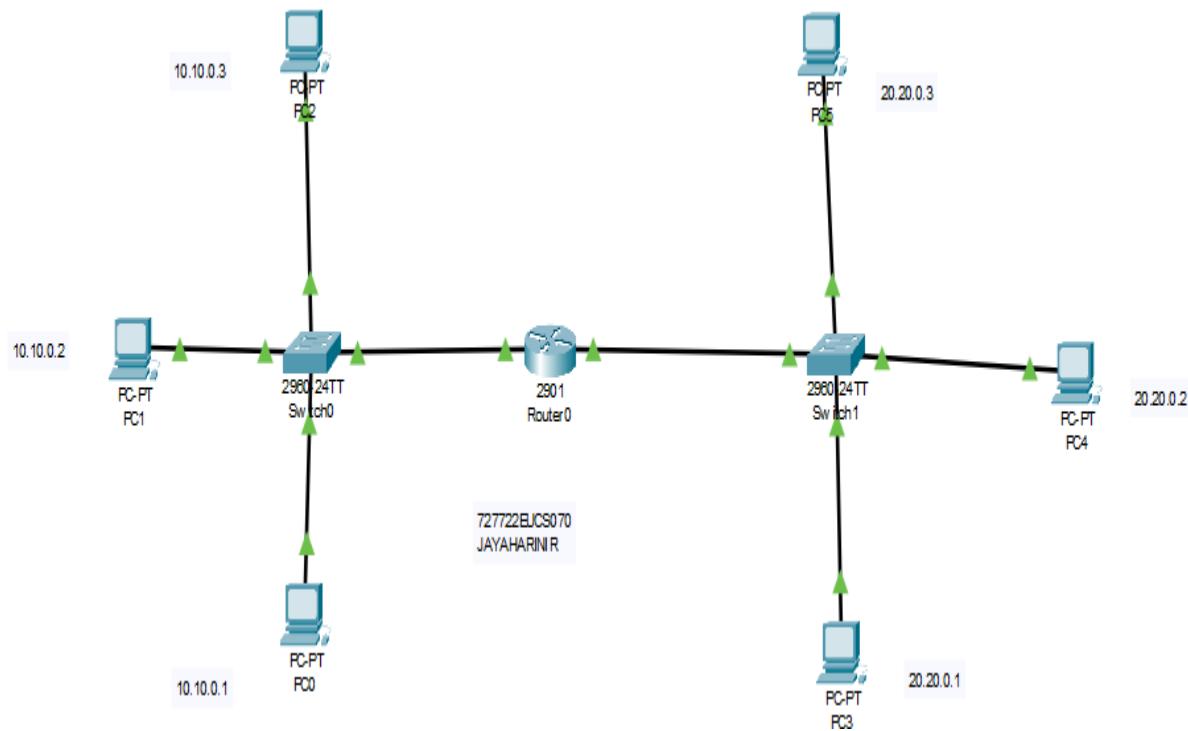
### **4. Verify WAN Link**

- Use show ip route to check static routes.
- Use ping between routers to ensure WAN serial connectivity.

### **5. Test End Device Communication**

- From PC0 (192.168.1.2), ping PC1 (192.168.2.2).
- Verify successful replies.

## Model Output:



## Results:

A Wide Area Network (WAN) was successfully simulated using Cisco Packet Tracer. Routers interconnected two LANs via a serial WAN link, and static routing enabled communication between PCs in different networks. The simulation verified WAN configuration and demonstrated inter-branch connectivity.

## **EX.NO:09**

**Date :**

### **HTTP Traffic Analysis Using Wireshark**

**Aim:**

To capture and analyze HTTP request and response messages using Wireshark, and to study the behavior of the HTTP protocol in client-server communication.

**Theory:**

The **Hypertext Transfer Protocol (HTTP)** is the foundation of web communication and functions as a **request–response protocol** between clients (such as web browsers) and servers. It operates at the **application layer** of the TCP/IP model and typically uses **port 80** for unsecured connections and **port 443** for secured connections (HTTPS).

When a user types a URL or clicks a link, the browser sends an **HTTP request** (usually a **GET** or **POST**) to the web server. The server processes this request and replies with an **HTTP response**, which may include a status code (e.g., **200 OK**, **404 Not Found**) and the requested resource (such as an HTML file, image, or JSON data).

Some key aspects of HTTP:

- **Methods:** GET, POST, HEAD, PUT, DELETE, OPTIONS.
- **Request Fields:** Host, User-Agent, Accept, Cookie.
- **Response Fields:** Status Code, Server, Content-Type, Content-Length.
- **Statelessness:** Each HTTP request is independent; the protocol does not remember previous interactions unless cookies/sessions are used.

Wireshark allows packet capture and provides detailed protocol dissection. By filtering http traffic, we can examine the **structure of messages**, headers, payloads, and even performance metrics like round-trip time.

## **Procedure:**

### **1. Start Wireshark Capture:**

- Launch Wireshark and select your active network interface (e.g., Wi-Fi).
- Begin the packet capture.

### **2. Filter for HTTP:**

- In the Wireshark filter bar, type:
- http

This displays only HTTP packets.

### **3. Generate HTTP Traffic:**

- Open a browser and visit:
- <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>
- This action generates HTTP GET requests and responses.

### **4. Stop Capture:**

- Return to Wireshark and stop the capture after the page has fully loaded.

### **5. Analyze HTTP Request:**

- Select the **HTTP GET** packet.
- Expand the “Hypertext Transfer Protocol” section.
- Note fields such as *Method*, *Host*, *User-Agent*, and *Request URI*.

### **6. Analyze HTTP Response:**

- Locate the **HTTP 200 OK** response from the server.
- Examine the *Status Code*, *Content-Type*, *Content-Length*, and other headers.

### **7. Optional Checks:**

- Right-click a request packet → “Follow → HTTP Stream” to view the full conversation.
- Save the capture for documentation.

## **MODEL OUTPUT :**

## Find the GET request

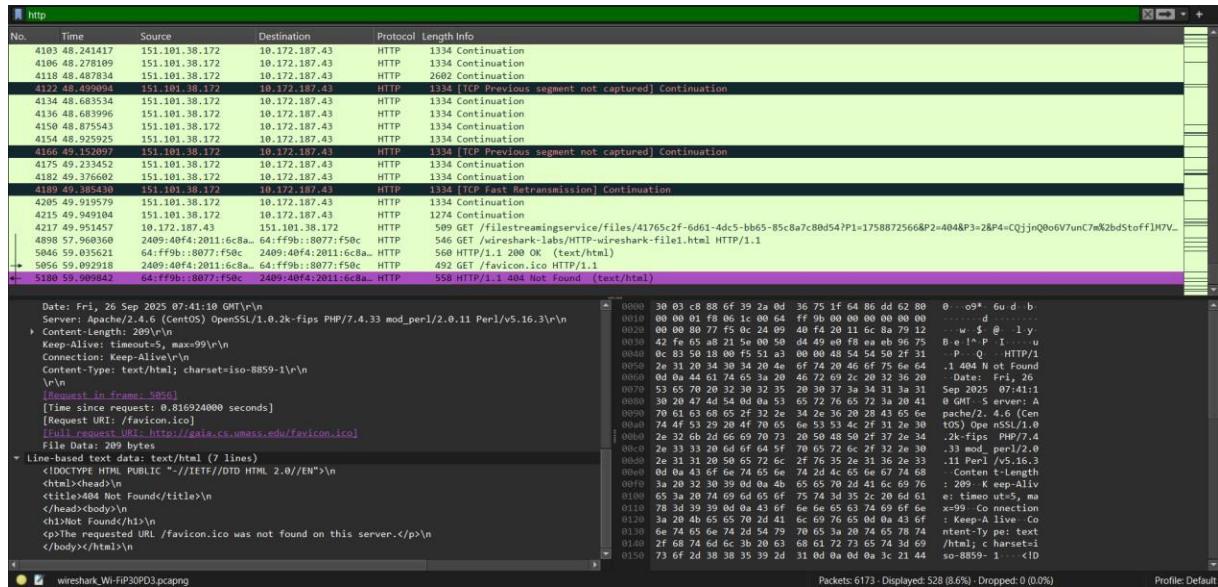
| No.                                                                                                 | Time               | Source                | Destination | Protocol | Length Info                                      |
|-----------------------------------------------------------------------------------------------------|--------------------|-----------------------|-------------|----------|--------------------------------------------------|
| 4103 48.241417                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4106 48.278109                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4118 48.487834                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 2602     | Continuation                                     |
| 4122 48.499994                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | [TCP Previous segment not captured] Continuation |
| 4124 48.683534                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4136 48.700001                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4150 48.875543                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4154 48.929525                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4166 49.152097                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | [TCP Previous segment not captured] Continuation |
| 4175 49.233452                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4182 49.376602                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4189 49.385430                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | [TCP Fast Retransmission] Continuation           |
| 4205 49.919579                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4215 49.949104                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1274     | Continuation                                     |
| 4242 49.951457                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4260 49.960732                                                                                      | 151.101.38.172     | 10.172.187.43         | HTTP        | 1334     | Continuation                                     |
| 4264 50.936261                                                                                      | 64:ff:fb:8077:f50c | 2409:40f4:2011:6c8a.. | HTTP        | 560      | HTTP/1.1 200 OK [text/html]                      |
| 5046 59.932918                                                                                      | 64:ff:fb:8077:f50c | 2409:40f4:2011:6c8a.. | HTTP        | 492      | GET /favicon.ico HTTP/1.1                        |
| 5180 59.909842                                                                                      | 64:ff:fb:8077:f50c | 2409:40f4:2011:6c8a.. | HTTP        | 558      | HTTP/1.1 404 Not Found [text/html]               |
| [Calculated window size: 65280]                                                                     |                    |                       |             |          |                                                  |
| [Window size scaling factor: 256]                                                                   |                    |                       |             |          |                                                  |
| Checksums: 0x2a40 [unverified]                                                                      |                    |                       |             |          |                                                  |
| [Checksum Status: Unverified]                                                                       |                    |                       |             |          |                                                  |
| Urgent Pointer: 0                                                                                   |                    |                       |             |          |                                                  |
| > [Timestamps]                                                                                      |                    |                       |             |          |                                                  |
| > [SEQ/ACK analysis]                                                                                |                    |                       |             |          |                                                  |
| TCP payload (472 bytes)                                                                             |                    |                       |             |          |                                                  |
| > HyperText Transfer Protocol                                                                       |                    |                       |             |          |                                                  |
| > GET /wreshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n                                         |                    |                       |             |          |                                                  |
| Host: galia.cs.umass.edu\r\n                                                                        |                    |                       |             |          |                                                  |
| Connection: keep-alive\r\n                                                                          |                    |                       |             |          |                                                  |
| Upgrade-Insecure-Requests: 1\r\n                                                                    |                    |                       |             |          |                                                  |
| User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome |                    |                       |             |          |                                                  |
| Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*; |                    |                       |             |          |                                                  |
| Accept-Encoding: gzip, deflate\r\n                                                                  |                    |                       |             |          |                                                  |
| Accept-Language: en-US,en;q=0.9\r\n\r\n                                                             |                    |                       |             |          |                                                  |
| [Response in frame: 5046]                                                                           |                    |                       |             |          |                                                  |
| [HTTP request URL: http://galia.cs.umass.edu/wreshark-labs/HTTP-wireshark-file1.html]               |                    |                       |             |          |                                                  |
| [0000 2a 0d 36 75 1f 64 90 03 c8 88 fd 39 86 dd 60 05 .. w d] .. -.-.~                              |                    |                       |             |          |                                                  |
| [0010 98 a2 01 ec 06 3f 99 40 f4 20 11 6c 8a 79 12 .. ? \$ @ 1 y                                    |                    |                       |             |          |                                                  |
| [0020 42 fe 65 a8 21 5e 06 64 ff 9b 00 00 00 00 00 B e] .. \u2014 d                                 |                    |                       |             |          |                                                  |
| [0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .. w - I P ..                                 |                    |                       |             |          |                                                  |
| [0040 e9 05 58 18 00 f5 2a 40 00 00 00 00 00 00 00 00 .. w - I P ..                                 |                    |                       |             |          |                                                  |
| [0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .. GET /w ..                                  |                    |                       |             |          |                                                  |
| [0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .. index.html HTTP/1.1 ..                     |                    |                       |             |          |                                                  |
| [0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .. IP-wires hark-fil ..                       |                    |                       |             |          |                                                  |
| [0080 65 31 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 .. e1.html HTTP/1.1 ..                        |                    |                       |             |          |                                                  |
| [0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .. Host: galia.cs ..                          |                    |                       |             |          |                                                  |
| [0090 75 6d 61 73 73 2a 65 64 75 0d 0a 43 6f 6e 65 umass.edu Conne ..                               |                    |                       |             |          |                                                  |
| [00a0 65 6d 61 73 6e 3a 26 6b 65 65 70 2d 61 6c 69 .. ction: k-eap ..                               |                    |                       |             |          |                                                  |
| [00b0 65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 .. Upgr ..                                          |                    |                       |             |          |                                                  |
| [00c0 65 6d 61 73 6e 3a 26 6b 65 65 70 2d 61 6c 69 .. Insec ..                                      |                    |                       |             |          |                                                  |
| [00d0 00 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f .. User-Ag ..                                    |                    |                       |             |          |                                                  |
| [00e0 66 6c 61 6f 25 3e 20 28 57 69 6e 64 6f 77 illa/5.0 (Window ..                                 |                    |                       |             |          |                                                  |
| [00f0 73 20 4e 58 20 30 2e 3b 20 57 69 6e 36 34 .. Moz ..                                           |                    |                       |             |          |                                                  |
| [0100 3b 20 78 36 34 29 20 41 70 70 6c 65 57 65 62 .. X-Apple ..                                    |                    |                       |             |          |                                                  |
| [0110 69 74 2f 35 33 37 2e 33 20 28 4b 48 54 4d .. X-Apple ..                                       |                    |                       |             |          |                                                  |
| [0120 29 30 35 34 29 20 41 70 70 6c 65 57 65 62 .. X-Apple ..                                       |                    |                       |             |          |                                                  |
| [0130 65 6d 65 6f 25 31 34 20 28 57 69 6e 64 6f .. X-Apple ..                                       |                    |                       |             |          |                                                  |
| [0140 61 66 61 70 62 20 35 33 37 2e 33 36 0d 0a 41 63 .. romu/14.0 ..                               |                    |                       |             |          |                                                  |
| [0140 65 65 70 74 3a 24 74 65 78 74 2f 68 74 6d 6c 2c .. 0.5 ..                                     |                    |                       |             |          |                                                  |
| [0150 65 65 70 74 3a 24 74 65 78 74 2f 68 74 6d 6c 2c .. cept: te/xhtml, ..                         |                    |                       |             |          |                                                  |

## Find the HTTP 200 OK response

```
Date: Fri, 26 Sep 2025 07:41:09 GMT\r\n\r\nServer: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\nLast-Modified: Fri, 26 Sep 2025 05:59:02 GMT\r\n\r\nEtag: \"80-63fauf8891d8\"\\r\\n\r\nAccept-Ranges: bytes\\r\\n\r\nContent-Length: 128\\r\\n\r\nKeep-Alive: timeout=5, max=100\\r\\n\r\nConnection: Keep-Alive\\r\\n\r\nContent-Type: text/html; charset=UTF-8\\r\\n\r\n\r\n[Request in frame: 4808]\r\n[Time since request: 1.057261000 seconds]\r\n[Request URL: /wireshark-labs/HTTP-wireshark-file1.html]\r\n[Full request URL: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]\r\nFile Data: 128 bytes\r\n+ Line-based text data: text/html (4 lines)\r\n html\\r\\n Congratulations. You've downloaded the file \\n\r\n http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html\\n\r\n
```

Packets: 6173 - Displayed: 528 (8.6%) - Dropped: 0 (0.0%) Profile: Default

## The 404 Not Found



## Result:

The experiment successfully demonstrated the working of the HTTP protocol using Wireshark. HTTP request messages such as **GET** were captured along with their corresponding **200 OK responses**. Detailed header information including *Host*, *User-Agent*, *Content-Type*, and *Status Codes* was analyzed. The lab helped in understanding the **stateless nature of HTTP** and the structure of web communication between client and server.

## **EX.NO:10**

**Date :**

### **5G Packet Capture and Analysis Using Wireshark**

**Aim:**

To capture and analyze 5G signaling and user-plane packets using Wireshark, and to study the functioning of 5G core protocols such as NAS (Non-Access Stratum) and NGAP (Next Generation Application Protocol).

**Theory:**

5G is the **fifth generation of mobile networks**, designed to deliver high bandwidth, ultra-low latency, and massive device connectivity. Unlike previous generations, 5G is built on a **service-based architecture (SBA)** and supports technologies like **network slicing, beamforming, and ultra-reliable low latency communication (URLLC)**.

In the 5G core, communication involves:

- **NAS-5GS:** Handles mobility management and session management messages between the User Equipment (UE) and the core network. Examples: *Registration Request, Authentication, Security Mode Command*.
- **NGAP:** A protocol between the gNB (5G base station) and AMF (Access and Mobility Management Function). It manages signaling for registration, session setup, and mobility.
- **PDCP/RLC Layers:** Used in user-plane data transmission.

Wireshark supports dissection of 5G protocols. With either real captures (from testbeds like **srsRAN/OpenAirInterface**) or public **sample pcaps**, students can study how devices connect, authenticate, and establish sessions in a 5G environment.

**Procedure:**

#### **1. Obtain 5G Capture File:**

- Download a sample capture file such as `5g-nr-registration.pcap` from Wireshark's official sample repository.
- Alternatively, use a 5G testbed to generate real packets.

#### **2. Open in Wireshark:**

- Launch Wireshark → File → Open → Select the .pcap file.

### 3. Apply Protocol Filters:

- To analyze signaling traffic, use filters like:
  - ngap || nas-5gs
- To view data-plane traffic, use filters like:
  - pdcp || gtp

### 4. Analyze Registration Procedure:

- Locate the **NAS Registration Request** message from UE to AMF.
- Observe *5G Mobile Identity (SUPI)*, *requested NSSAI*, and security details.
- Find the **Registration Accept** message and examine confirmation fields.

### 5. Examine Session Setup:

- Identify NGAP messages related to **PDU Session Establishment**.
- Check session type (IPv4/IPv6), QoS parameters, and resource allocation.

### 6. Study Data Packets (Optional):

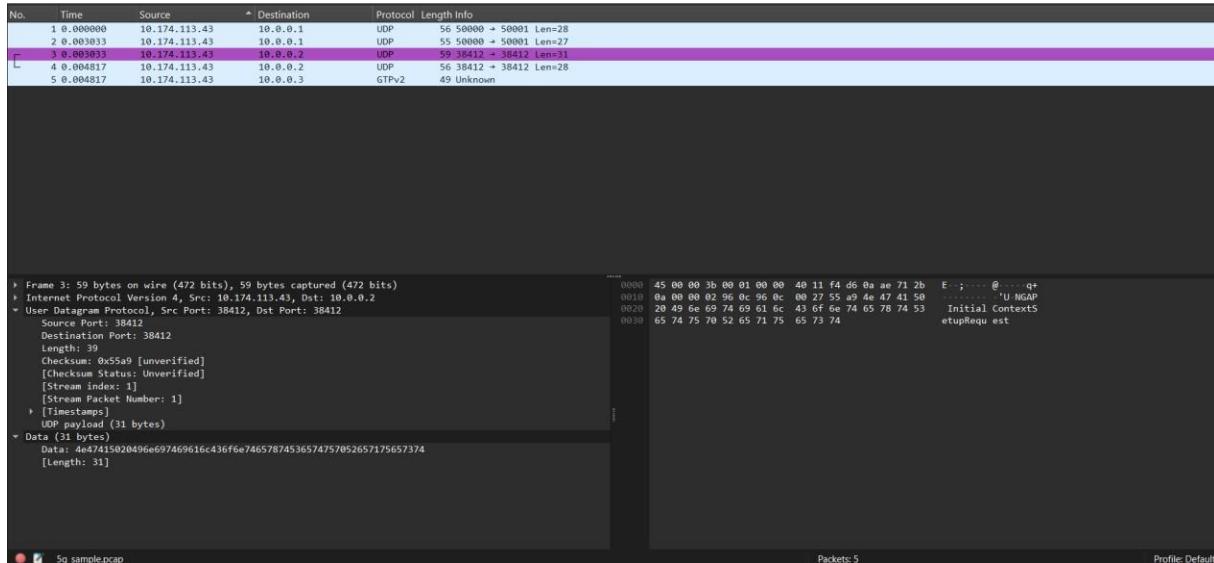
- Trace user-plane traffic tunneled over GTP/PDCP.
- Inspect throughput and latency if available.

### 7. Document Outputs:

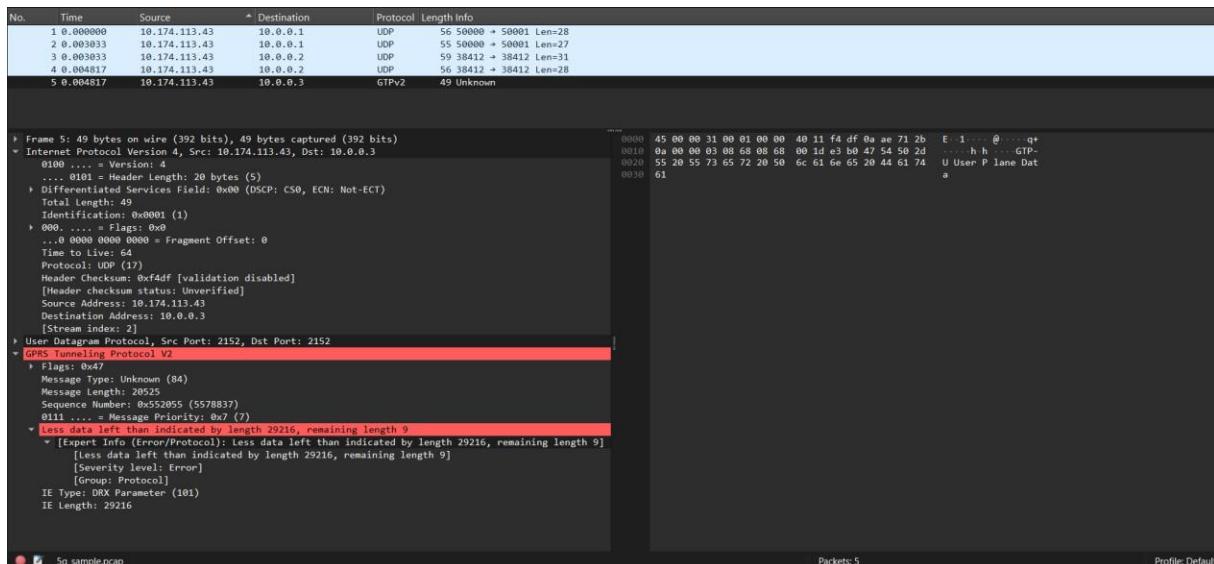
- Take screenshots of NAS Registration Request, Registration Accept, and NGAP session setup.

## MODEL OUTPUT :

### UDP Packet Analysis



### GTPv2 Packet Analysis



## Result:

The experiment successfully demonstrated the **capture and analysis of 5G packets using Wireshark**. Key 5G protocols including **NAS-5GS** and **NGAP** were examined. The Registration Request and Accept messages provided insights into how a UE attaches to the 5G core. Session establishment procedures were also studied, showing how data connectivity is enabled in 5G networks. This lab enhanced understanding of **5G architecture, signaling flow, and protocol stack analysis**.