CSA0704-Computer Networks for Data Communication

LAB MANUAL

LIST OF EXPERIMENTS

Sl. No	Experiment	
1.	Configuration of Network Devices using Packet Tracer tools (Hub,Switch, Ethernet, Broadcast).	
2.	Design and Configuration of Star Topologies using Packet Tracer.	
3.	Design and Configuration of BUS Topologies using Packet Tracer.	
4.	Design and Configuration of RING Topologies using Packet Tracer.	
5.	Design and Configuration of Mesh Topologies using Packet Tracer.	
6.	Design and Configuration of Tree Topologies using Packet Tracer.	
7.	Design and Configuration of Hybrid Topologies using Packet Tracer.	
8.	Data Link Layer Traffic Simulation using Packet Tracer Analysis of ARP.	
9.	Data Link Layer Traffic Simulation using Packet Tracer Analysis of LLDP.	
10.	Data Link Layer Traffic Simulation using Packet Tracer Analysis of CSMA/CD & CSMA/CA.	
11.	Designing two different network with Static Routing techniques using Packet Tracer.	
12.	Designing two different networks with Dynamic Routing techniques (RIP & OSPF) using Packet Tracer	
13.	Design the Functionalities and Exploration of TCP using Packet Tracer.	
14.	Design the Functionalities and Exploration of UDP using Packet Tracer.	
15.	Design the network model for Subnetting – Class C Addressing using Packet Tracer.	
16.	Simulating X, Y, Z Company Network Design and simulate using Packet Tracer.	
17.	Configuration of DHCP (dynamic host configuration protocol) in packet Tracer.	
18.	Configuration of firewall in packet tracer.	
19.	Make a Computer Lab to transfer a message from one node to another to design and simulate using Cisco Packet Tracer.	
20.	Simulate a Multimedia Network in Cisco Packet Tracer.	
21.	IoT based smart home applications.	
22.	Implementation of IoT based smart gardening.	
23.	Implementation of IoT devices in networking.	
24.	IOT Based Smart building using WPA Security & Radius Server.	
25.	Transport layer protocol header analysis using Wire shark- TCP	

26.	Transport layer protocol header analysis using Wire shark- UDP.	
27.	Network layer protocol header analysis using Wire shark – SMTP	
28.	Network layer protocol header analysis using Wire shark –ICMP.	
29.	Network layer protocol header analysis using Wire shark – ARP	
30.	Network layer protocol header analysis using Wire shark – HTTP.	
31.	Identify and monitor the IP, network address, Trace the router information, how to take remote system and check the node connection in network	
32.	Demonstration of PING operation using ICMP in Wireshark	
33.	Implementation of Bit stuffing mechanism using C.	
34.	Implementation of server – client using TCP socket programming.	
35.	Implementation of server – client using UDP socket programming.	

Date:

EXPERIMENT-1

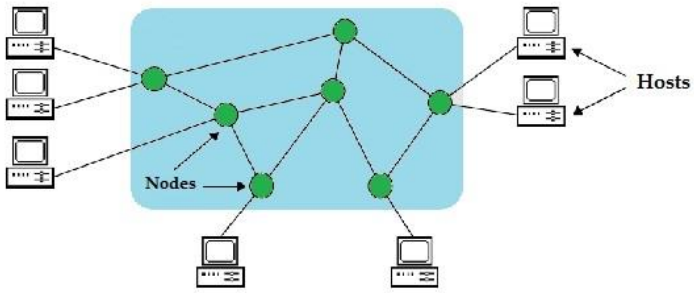
CONFIGURATION OF NETWORK COMPONENTS

Aim: To Study the following Network Devices in Detail

- PC
- Server
- Repeater
- Hub
- Switch
- Bridge
- Router
- Gate Way
- Transmission medium

Apparatus (Software): CISCO Packet tracer.

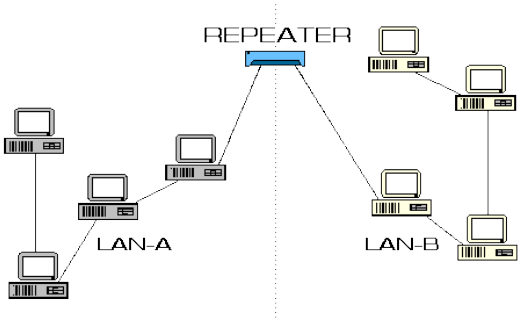
1. Node: In a communications ***network***, a ***network node*** is a connection point that can receive, create, store or send data along distributed ***network*** routes.



Computer Network

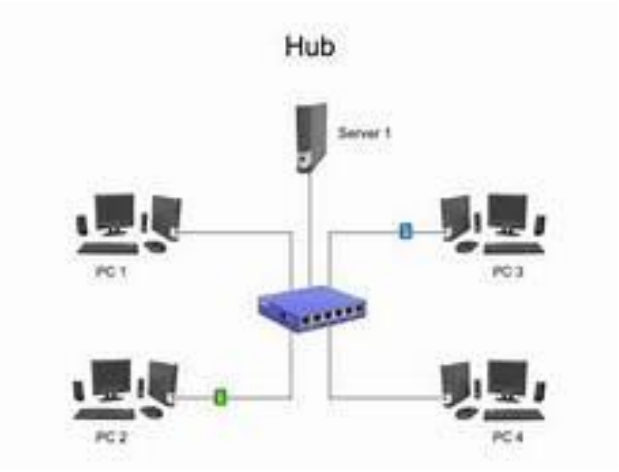
2. **Repeater:** Functioning at Physical Layer.

A **repeater** is an electronic device that receives a signal and retransmits it at a higher level and/or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances.

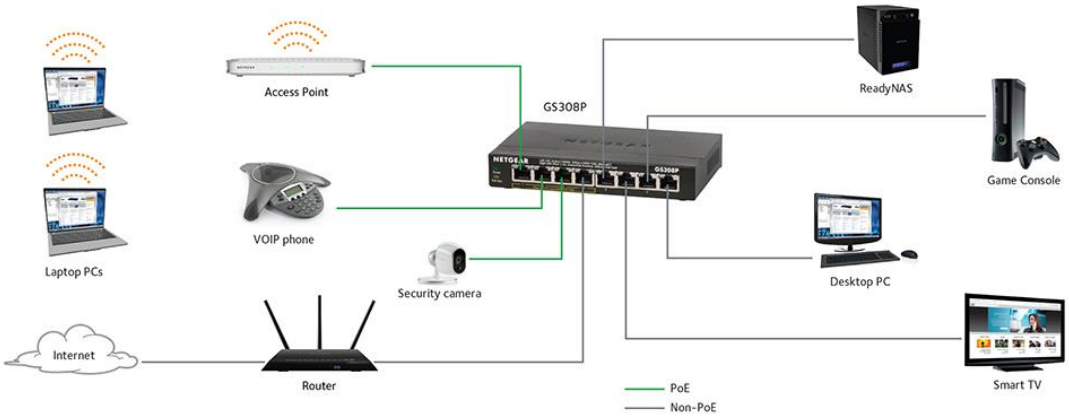


3. Hub: Ethernet hub, **active hub**, **network hub**, **repeater hub**

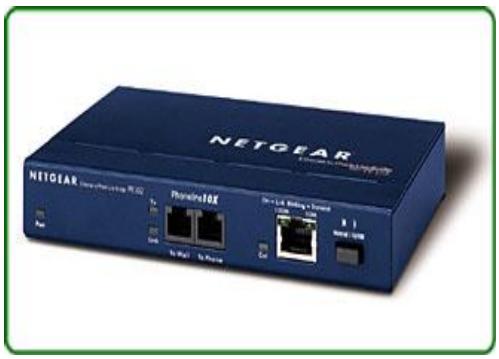
Hub or concentrator is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and making them act as a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. The device is a form of multiport repeater. Repeater hubs also participate in collision detection, forwarding a jam signal to all ports if it detects a collision.

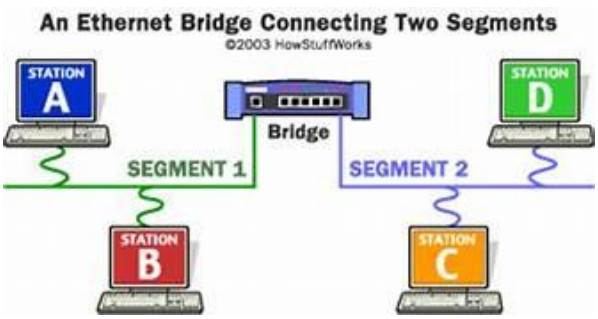


4. **Switch:** A **network switch** or **switching hub** is a computer networking device that connects network segments. The term commonly refers to a network bridge that processes and routes data at the data link layer (layer 2) of the OSI model. Switches that additionally process data at the network layer (layer 3 and above) are often referred to as Layer 3 switches or multilayer switches.



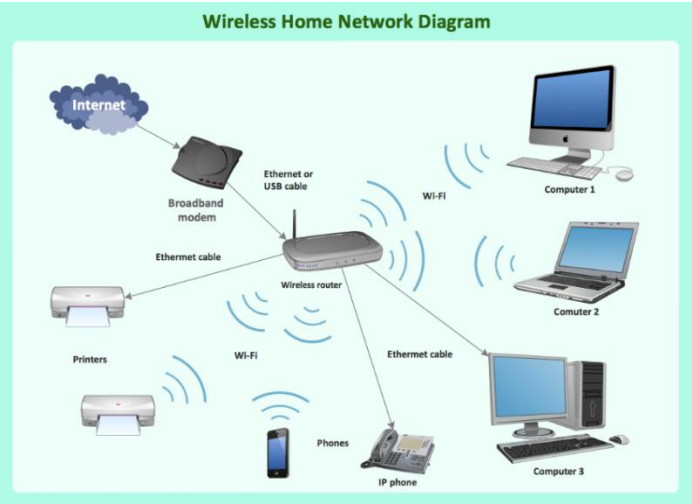
5. **Bridge:** A **network bridge** connects multiple network segments at the data link layer (Layer 2) of the OSI model. In Ethernet networks, the term bridge formally means a device that behaves according to the IEEE 802.1D standard. A bridge and switch are very much alike; a switch being a bridge with numerous ports. Switch or Layer 2 switch is often used interchangeably with bridge. Bridges can analyze incoming data packets to determine if the bridge is able to send the given packet to another segment of the network.



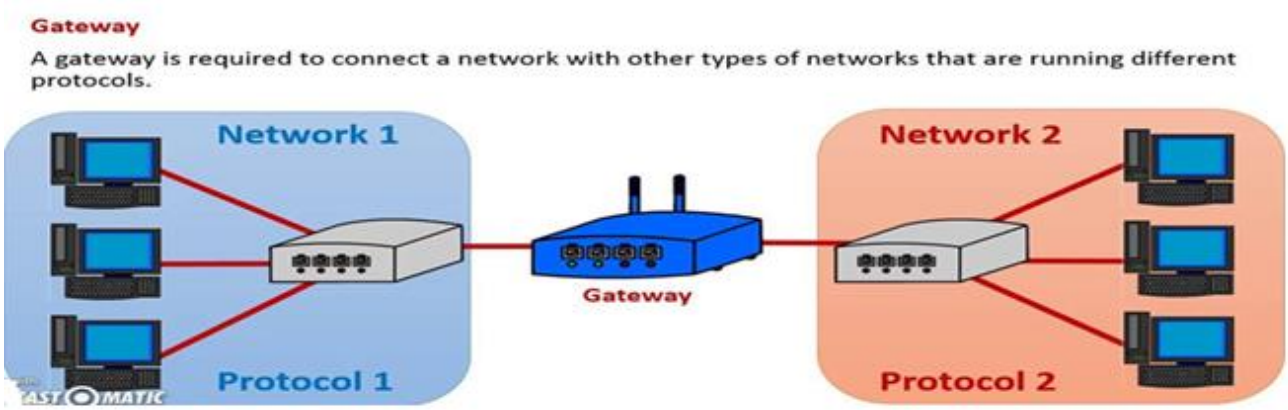


6. **Router:** A **router** is an electronic device that interconnects two or more computer

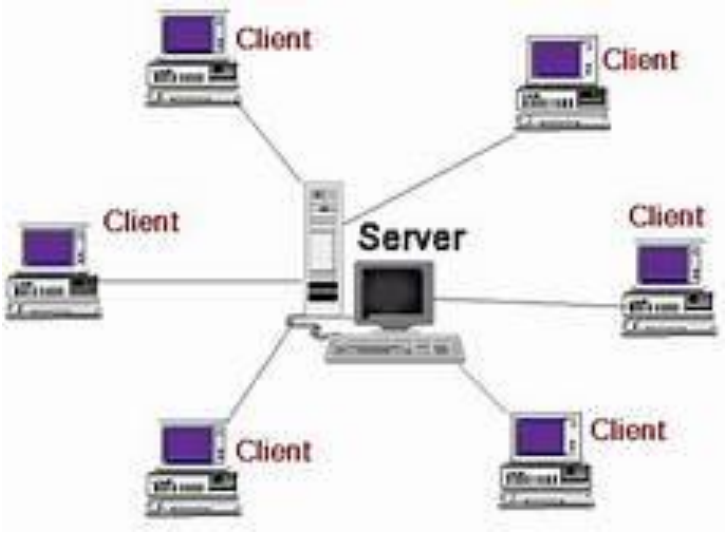
networks, and selectively interchanges packets of data between them. Each data packet contains address information that a router can use to determine if the source and destination are on the same network, or if the data packet must be transferred from one network to another. The multiple routers are used in a large collection of interconnected networks, the routers exchange information about target system addresses, so that each router can build up a table showing the preferred paths between any two systems on the interconnected networks.



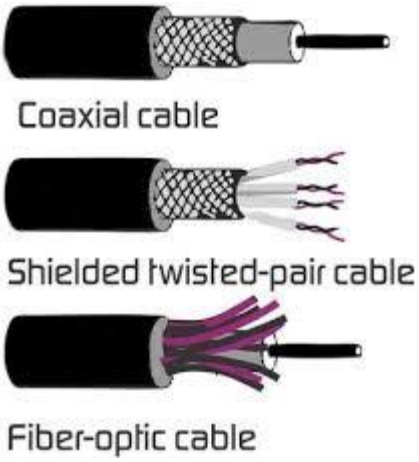
7. **Gate Way:** In a communication network, a network node equipped for interfacing with another network that uses different protocols. A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It also requires the establishment of mutually acceptable administrative procedures between both networks.
    - A protocol translation/mapping gateway interconnects networks with different network protocol technologies by performing the required protocol conversions.

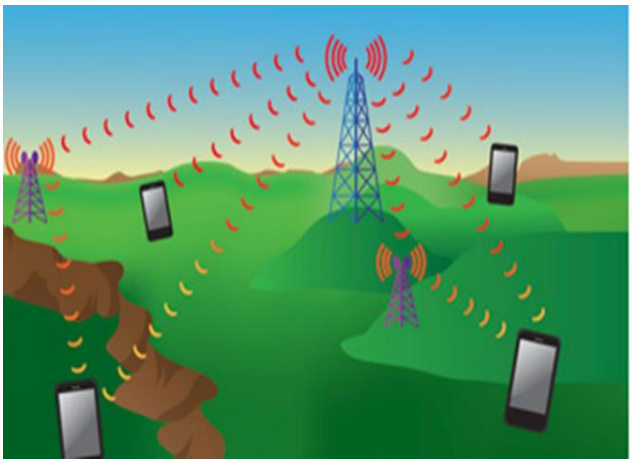


8. Server: A server is a type of computer or device on a network that manages network resources. Servers are often dedicated, meaning that they perform no other tasks besides their server tasks. On multiprocessing operating systems, however, a single computer can execute several programs at once. A server in this case could refer to the program that is managing resources rather than the entire computer.



9. **Transmission media**: The medium through which the signals travel from one device to another. These are classified as guided and unguided. Guided media are those that provide a conduit from one device to another. Eg. Twisted pair, coaxial cable etc. Unguided media transport signals without using physical cables. Eg. Air.





Result: Thus the network components are studied in detail.

Date:

EXPERIMENT-2

IMPLEMENTATION OF STAR TOPOLOGY USING PACKET TRACER

Aim: To Implement a star topology using packet tracer and hence to transmit data between the devices connected using star topology.

Software/Apparatus required: Packet Tracer/End devices, bridge, connectors.

Steps for building topology:

Step 1: Start Packet Tracer

Step 2: Choosing Devices and Connections

Step 3: Building the Topology – Adding Hosts

Single click on the **End Devices**.

Single click on the **Generic** host.

Move the cursor into topology area.

Single click in the topology area and it copies the device.

Step 4: Building the Topology – Connecting the Hosts to Switches

Select a switch, by clicking once on **Switches** and once on a **2950-24** switch.

Add the switch by moving the plus sign "+"

Step 5: Connect PCs to switch by first choosing Connections

Click once on the **Copper Straight-through** cable

Click once on **PC2**

Choose **Fast Ethernet**

Drag the cursor to **Switch0**

Click once on **Switch0**

Notice the green link lights on **PC** Ethernet NIC and amber light **Switch port**. The switch port is temporarily not forwarding frames, while it goes through the stages for the Spanning Tree Protocol (STP) process. After about 30 seconds the amber light will change to green indicating that the port has entered the forwarding stage. Frames can now forwarded out the switch port.

Step 6: Configuring IP Addresses and Subnet Masks on the Hosts

To start communication between the hosts IP Addresses and Subnet Masks had to be Configured on the devices. Click once on PC0. Choose the Config tab and click on

FastEthernet0. Type the IP address in its field. Click on the subnet mask it will be generated automatically.

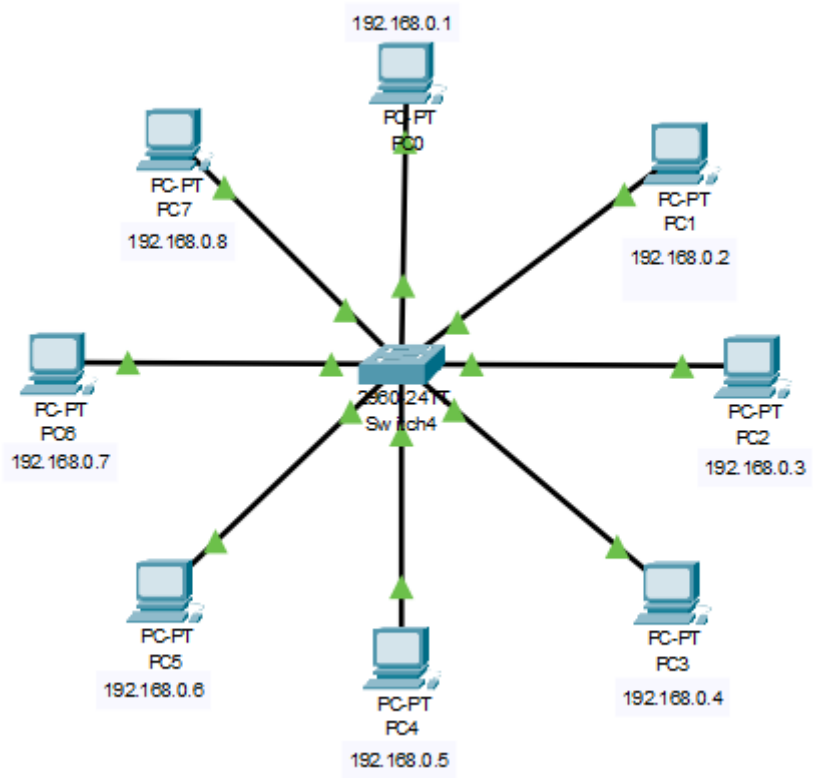
Step 7: To confirm Data transfer between the devices

Click on the node. Select desktop option and then command prompt. Once the window

pops up, ping the IP address of the device to which node0 is connected. Ping statistics will be

displayed.

Diagram:



Output:

Result: Thus the Star topology is implemented with Packet Tracer simulation Tool.

Date:

EXPERIMENT-3

IMPLEMENTATION OF BUS TOPOLOGY USING PACKET TRACER

Aim: To Implement a Bus topology using packet tracer and hence to transmit data between the devices connected using Bus topology.

Software / Apparatus required: Packet Tracer / End devices, Hubs, connectors.

Steps for building topology:

Step 1: Start Packet Tracer

Step 2: Choosing Devices and Connections

Step 3: Building the Topology – Adding Hosts

Single click on the **End Devices**.

Single click on the **Generic** host.

Move the cursor into topology area.

Single click in the topology area and it copies the device.

Step 4: Building the Topology – Connecting the Hosts to Switches

Select a switch, by clicking once on **Switches** and once on a **2950-24** switch.

Add the switch by moving the plus sign "+"

Step 5: Connect PCs to switch by first choosing connections

Click once on the **Copper Straight-through** cable

Click once on **PC2**

Choose **Fast Ethernet**

Drag the cursor to **Switch0**

Click once on **Switch0**

Notice the green link lights on **PC** Ethernet NIC and amber light **Switch port**. The switch port is temporarily not forwarding frames, while it goes through the stages for the Spanning Tree Protocol (STP) process. After about 30 seconds the amber light will change to green indicating that the port has entered the forwarding stage. Frames can now forward out the switch port.

Step 6: Configuring IP Addresses and Subnet Masks on the Hosts

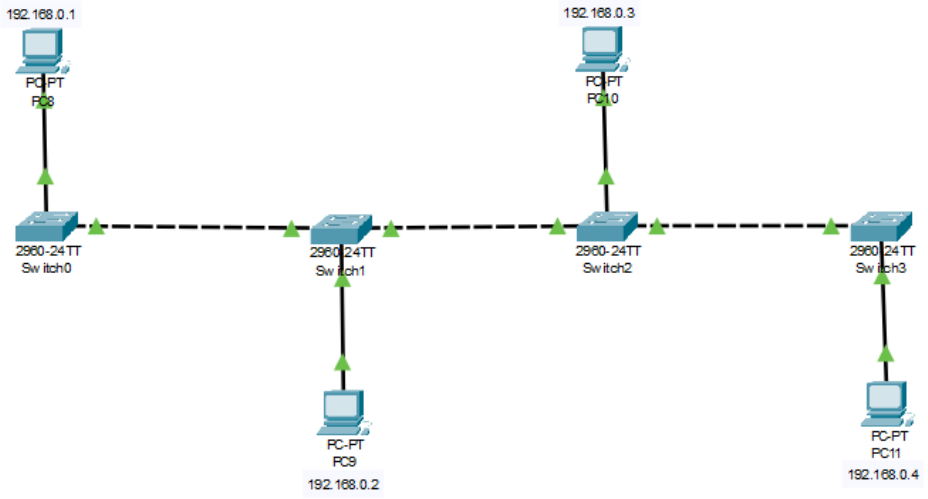
To start communication between the hosts IP Addresses and Subnet Masks had to be configured on the devices. Click once on PC0. Choose the Config tab and click on FastEthernet0. Type the IP address in its field. Click on the subnet mask it will

be generated automatically.

Step 7: To confirm Data transfer between the devices

Click on the node. Select desktop option and then command prompt. Once the window pops up, ping the IP address of the device to which node0 is connected. Ping statistics will be displayed.

Diagram:



Output:

Result: Thus the Bus topology is implemented with Packet Tracer simulation Tool.

Date:

EXPERIMENT-4

IMPLEMENTATION OF RING TOPOLOGY USING PACKET TRACER

Aim: To Implement a Ring topology using packet tracer and hence to transmit data between the devices connected using Ring topology.

Software / Apparatus required: Packet Tracer / End devices, Hubs, Connectors.

Steps for building topology:

Step 1: Start Packet Tracer

Step 2: Choosing Devices and Connections

Step 3: Building the Topology – Adding Hosts

Single click on the **End Devices**.

Single click on the **Generic** host.

Move the cursor into topology area.

Single click in the topology area and it copies the device.

Step 4: Building the Topology – Connecting the Hosts to Switches

Select a switch, by clicking once on **Switches** and once on a **2950-24** switch.

Add the switch by moving the plus sign "+"

Step 5: Connect PCs to switch by first choosing connections

Click once on the **Copper Straight-through** cable

Click once on **PC2**

Choose **Fast Ethernet**

Drag the cursor to **Switch0**

Click once on **Switch0**

Notice the green link lights on **PC** Ethernet NIC and amber light **Switch port**. The switch port is temporarily not forwarding frames, while it goes through the stages for the Spanning Tree Protocol (STP) process. After about 30 seconds the amber light will change to green indicating that the port has entered the forwarding stage. Frames can now forward out the switch port.

Step 6: Configuring IP Addresses and Subnet Masks on the Hosts

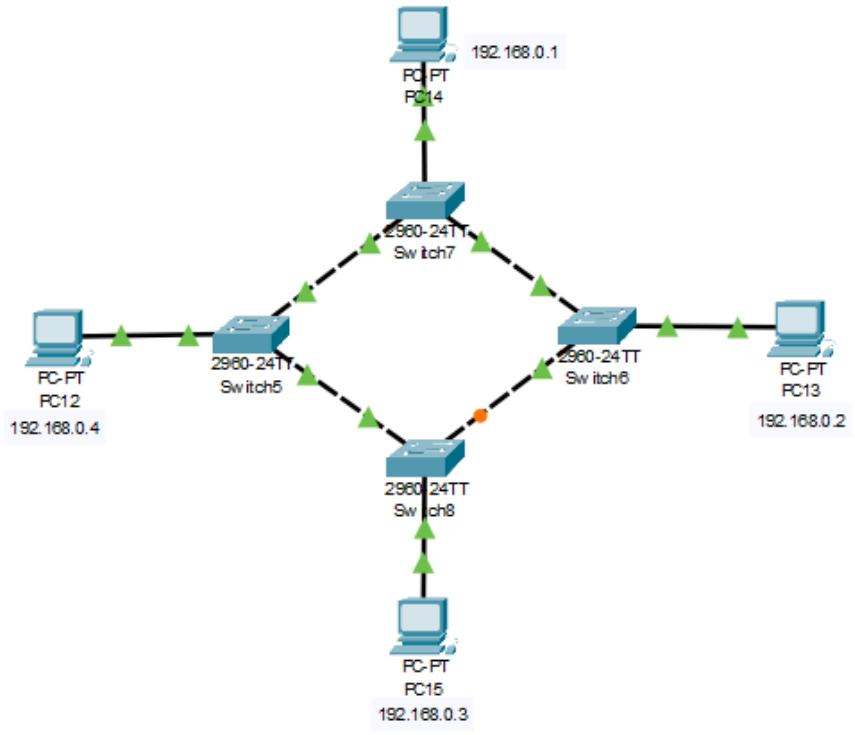
To start communication between the hosts IP Addresses and Subnet Masks had to be configured on the devices. Click once on PC0. Choose the Config tab and click on FastEthernet0. Type the IP address in its field. Click on the subnet mask it will

be generated automatically.

Step 7: To confirm Data transfer between the devices

Click on the node. Select desktop option and then command prompt. Once the window pops up, ping the IP address of the device to which node0 is connected. Ping statistics will be displayed.

Diagram:



Output

Result: Thus the Ring topology is implemented with Packet Tracer simulation Tool.

Date:

EXPERIMENT-5

IMPLEMENTATION OF MESH TOPOLOGY USING PACKET TRACER

Aim: To Implement a Mesh topology using packet tracer and hence to transmit data between the devices connected using Mesh topology.

Software / Apparatus required: Packet Tracer / End devices, Hubs, Connectors.

Steps for building topology:

Step 1: Start Packet Tracer

Step 2: Choosing Devices and Connections

Step 3: Building the Topology – Adding Hosts

Single click on the **End Devices**.

Single click on the **Generic** host.

Move the cursor into topology area.

Single click in the topology area and it copies the device.

Step 4: Building the Topology – Connecting the Hosts to Switches

Select a switch, by clicking once on **Switches** and once on a **2950-24** switch.

Add the switch by moving the plus sign "**+**"

Step 5: Connect PCs to switch by first choosing connections

Click once on the **Copper Straight-through** cable

Click once on **PC2**

Choose **Fast Ethernet**

Drag the cursor to **Switch0**

Click once on **Switch0**

Notice the green link lights on **PC** Ethernet NIC and amber light **Switch port**. The switch port is temporarily not forwarding frames, while it goes through the stages for the Spanning Tree Protocol (STP) process. After about 30 seconds the amber light will change to green indicating that the port has entered the forwarding stage. Frames can now forward out the switch port.

Step 6: Configuring IP Addresses and Subnet Masks on the Hosts

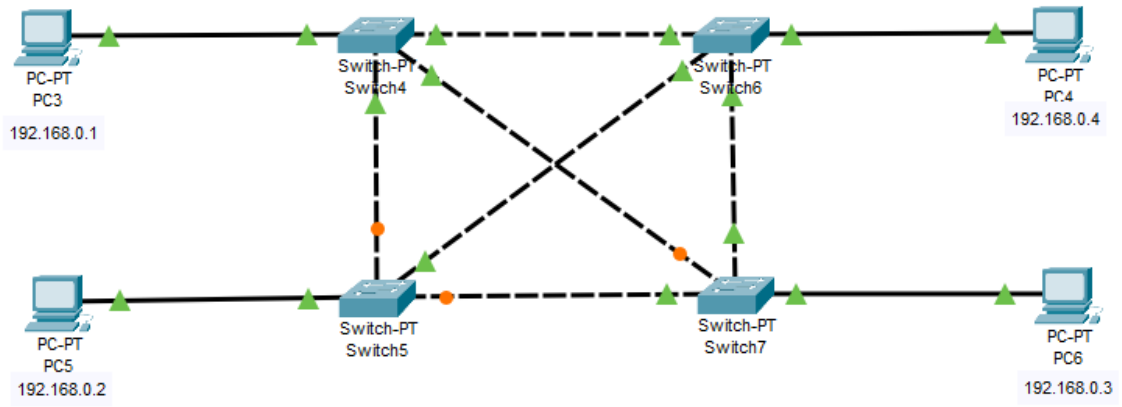
To start communication between the hosts IP Addresses and Subnet Masks had to be configured on the devices. Click once on PC0. Choose the Config tab and click

on FastEthernet0. Type the IP address in its field. Click on the subnet mask it will be generated automatically.

Step 7: To confirm Data transfer between the devices

Click on the node. Select desktop option and then command prompt. Once the window pops up, ping the IP address of the device to which node0 is connected. Ping statistic will be displayed.

Diagram:



Output:

Result: Thus the Mesh topology is implemented with Packet Tracer simulation Tool.

Date:

EXPERIMENT-6

IMPLEMENTATION OF TREE TOPOLOGY USING PACKET TRACER

Aim: To Implement a tree topology using packet tracer and hence to transmit data between the devices connected using tree topology.

Software / Apparatus required: Packet Tracer / End devices, Hubs, connectors.

Procedure:

Steps for building topology:

Step 1: Start Packet Tracer

Step 2: Choosing Devices and Connections

Step 3: Building the Topology – Adding Hosts

Single click on the **End Devices**.

Single click on the **Generic** host.

Move the cursor into topology area.

Single click in the topology area and it copies the device.

Step 4: Building the Star Topology – Connecting the Hosts to Hubs

Select a Hub, by clicking once on **Hub** and once on a **generic Hub**

Add the Hub by moving the plus sign "+"

Step 5: Connect PCs to Hub by first choosing Connections

Click once on the **Automatic cable selector**

Click once on **PC2**

Choose **Fast Ethernet**

Drag the cursor to **Hub0**

Click once on **Hub0**

Proceeding in this way create three star topologies

Step 6: Building the Tree Topology – Connecting the Hubs to Active Hub

Connect the hubs of star topologies to active hub to create tree topology.

Step 7: Configuring IP Addresses and Subnet Masks on the Hosts

To start communication between the hosts IP Addresses and Subnet Masks had to be configured on the devices. Click once on PC0. Choose the Config tab and click on Fast Ethernet0. Type the IP address in its field. Click on the subnet mask. It will be

generated automatically.

Step 8: Verifying Connectivity in Real time Mode

Be sure you are in **Real time** mode.

Select the **Add Simple PDU** tool used to ping devices.

Click once on PC0, then once on PC3.

The PDU **Last Status** should show as **Successful**.

Step 9: **Verifying Connectivity in Simulation Mode**

Be sure you are in **Simulation** mode.

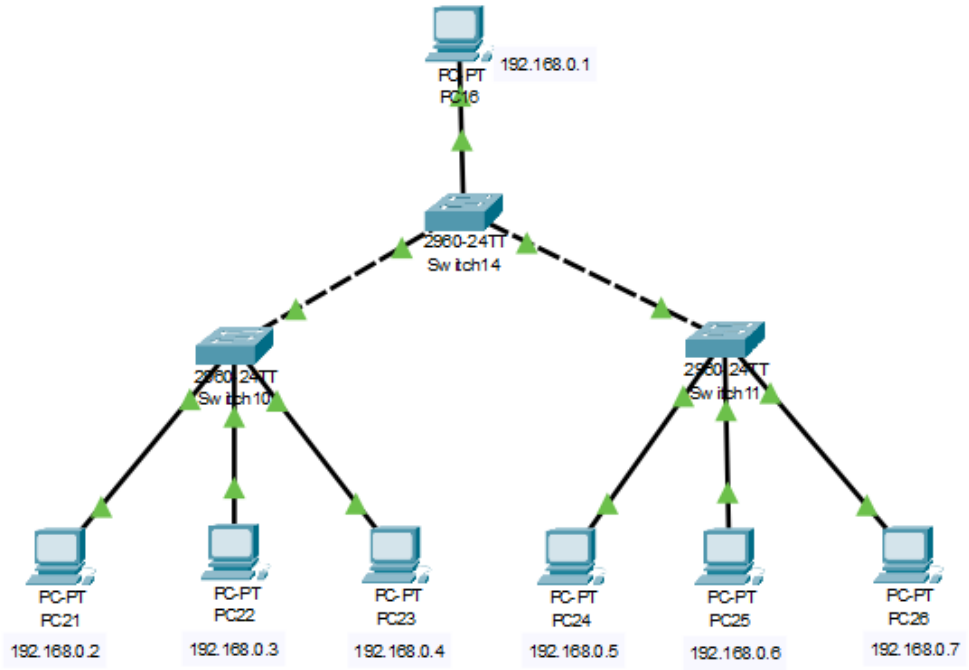
Deselect all filters (All/None) and select only **ICMP**.

Select the **Add Simple PDU** tool used to ping devices

Click once on PC0, then once on PC3.

Continue clicking **Capture/Forward** button until the ICMP ping is completed. You should see the ICMP messages move between the hosts, hub and switch. The PDU **last status** should show as **Successful**.

Diagram:



Output:

Result: Thus the Tree topology is implemented with Packet Tracer simulation Tool.

Date:

EXPERIMENT-7

IMPLEMENTATION OF HYBRID TOPOLOGY (BUS AND RING TOPOLOGY) USING PACKET TRACER

Aim: To Implement a hybrid topology using packet tracer and hence to transmit data between the devices connected using tree topology.

Software / Apparatus required: Packet Tracer / End devices, Hubs, connectors.

Steps for building topology:

Step 1: Start Packet Tracer

Step 2: Choosing Devices and Connections

Step 3: Building the Topology – Adding Hosts

Single click on the **End Devices**.

Single click on the **Generic** host.

Move the cursor into topology area.

Single click in the topology area and it copies the device.

Step 4: Building the Bus Topology – Connecting the Hosts to Hubs

Select a Hub, by clicking once on **Hub** and once on a **generic Hub**

Add the Hub by moving the plus sign "+"

Step 5: Building the Ring Topology – Connecting the Hosts to Hubs

Select a Hub, by clicking once on **Hub** and once on a **generic Hub**

Add the Hub by moving the plus sign "+"

Step 5: Connect PCs to Hub by first choosing Connections

Click once on the **Automatic cable selector**

Click once on **PC2**

Choose **Fast Ethernet**

Drag the cursor to **Hub0**

Click once on **Hub0**

Proceeding in this way create three Bus topologies

Step 6: Building the Tree Topology – Connecting the Hubs to Active Hub

Connect the hubs of star topologies to active hub to create tree topology.

Step 7: Configuring IP Addresses and Subnet Masks on the Hosts

To start communication between the hosts IP Addresses and Subnet Masks had to be configured on the devices. Click once on PC0. Choose the Config tab and click on FastEthernet0. Type the IP address in its field. Click on the subnet mask. It will be Generated automatically.

Step 8: Verifying Connectivity in Realtime Mode

Be sure you are in **Realtime** mode.

Select the **Add Simple PDU** tool used to ping devices.

Click once on PC0, then once on PC3.

The PDU **Last Status** should show as **Successful**.

Step 9: **Verifying Connectivity in Simulation Mode**

Be sure you are in **Simulation** mode.

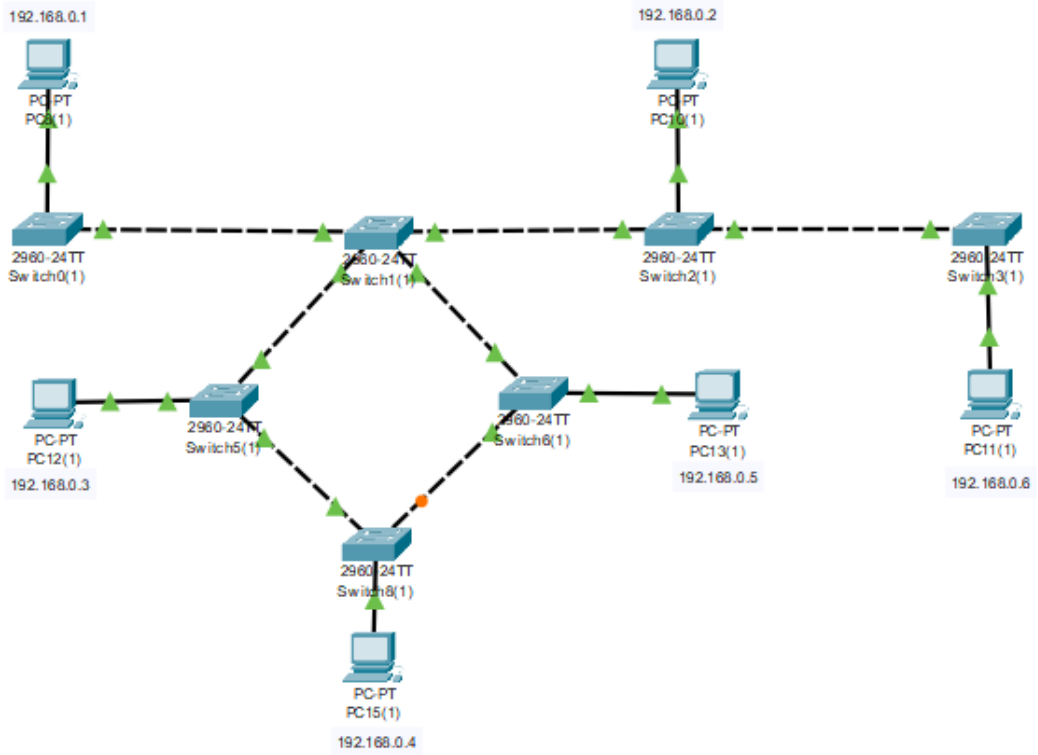
Deselect all filters (All/None) and select only **ICMP**.

Select the **Add Simple PDU** tool used to ping devices

Click once on PC0, then once on PC3.

Continue clicking **Capture/Forward** button until the ICMP ping is completed. The ICMP messages move between the hosts, hub and switch. The PDU **Last Status** should show as **Successful**.

Diagram:



Output:

Result: Thus the Hybrid topology is implemented with Packet Tracer simulation Tool.

Date:

EXPERIMENT-8

DATA LINK LAYER TRAFFIC SIMULATION USING PACKET TRACER

ANALYSIS OF ARP

Aim: To implement Data Link Layer Traffic Simulation using Packet Tracer Analysis of ARP.

Software / Apparatus required: Packet Tracer / End devices, Switches, connectors.

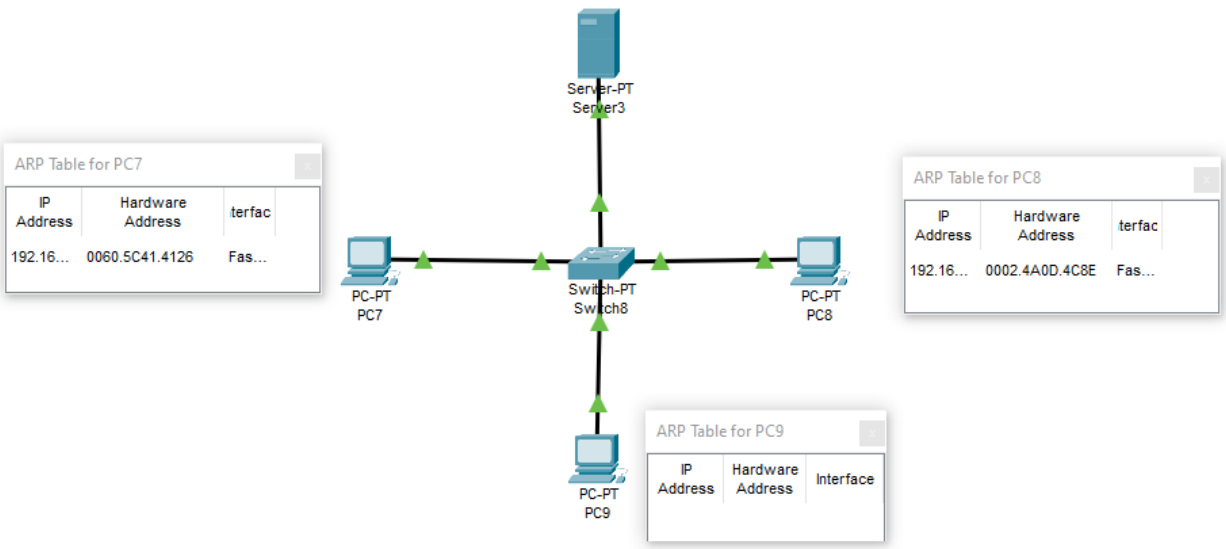
Requirements:

1. End device - They are the devices through which we can pass message from one device to another and they are interconnected.
2 Switch/Hub - Interface Between two devices.
3. Cable - Used to connect two devices

Procedure:

1. Open packet tracer.
2. Click on the list the available capture interface.
3. Choose the PCS, server and Hub.
4. Later give connection from hub to the remaining pcs.
5. Give IP address to the pcs with configuration.
6. Simulate the source and destination.

Diagram



Output :

Result: Thus the Data Link Layer Traffic Simulation using Packet Tracer Analysis of ARP is implemented.

Date:

EXPERIMENT-9

DATA LINK LAYER TRAFFIC SIMULATION USING PACKET TRACER

ANALYSIS OF CSMA/CD & CSMA/CA

Aim: To implement Data Link Layer Traffic Simulation using Packet Tracer Analysis of CSMA/CD & CSMA/CA.

Software / Apparatus required: Packet Tracer / End devices, Switches, connectors.

Requirements:

1. End device - They are the devices through which we can pass message from one device to another and they are interconnected.
2 Switch/Hub - Interface Between two devices.
3. Cable - Used to connect two devices

Procedure:

STEP 1: Click on end devices, select generic Pc's drag and drop it on the window. Click on SWITCH drag and drop it on the window.

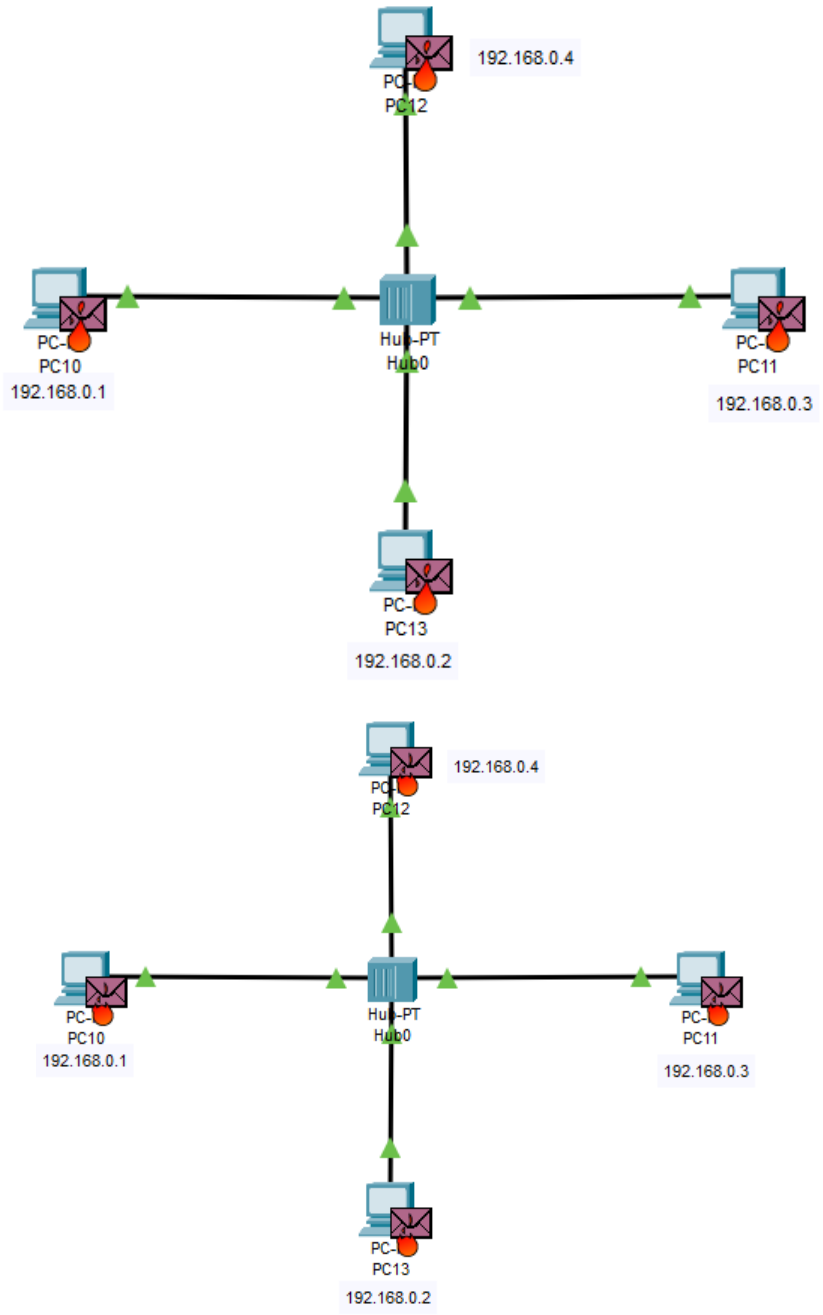
STEP 2: Select the straight through cable and connect all end device to switch. Assign the IP address for all end devices. (Double click the end device Select → desktop → IP configuration static)

STEP 3: Now set the IP address to Host A (192.168.1.1) in static mode. Similarly set IP address for Host B (192.168.1.2) and Host C (192.168.1.3)

STEP 4: To view the IP address, give ip config command in command prompt. Using ping command, we can establish communication between two host devices.

STEP 5: Now display the packet transmission in simulation mode.

Diagram:



Output:

Result: Thus Data Link Layer Traffic Simulation using Packet Tracer Analysis of CSMA/CD & CSMA/CA is implemented successfully.

Date:

EXPERIMENT-10

MAKING COMPUTER LAB IN CISCO PACKET TRACER

Aim: Making Computer Lab in Cisco Packet Tracer.

Software / Apparatus required: Packet Tracer / End devices, Switches, connectors.

Procedure:

Step 1: Launch Cisco Packet Tracer and create a new project.

Step 2: Select the appropriate network devices for your lab. In this case, you will need computers, switches, and routers. You can find these devices in the "End Devices," "Switches," and "Routers" sections of the device list.

Step 3: Drag and drop a switch onto the workspace area. Connect the switch to the power source by clicking on the "Connection" option and selecting "Power."

Step 4: Connect computers to the switch by dragging and dropping them onto the workspace area. Click on the "Connection" option and select "Fast Ethernet" to connect the computers to the switch.

Step 5: Repeat Step 4 to add more computers to the lab. You can adjust the number of computers as per your requirements.

Step 6: Connect the switch to a router. Drag and drop a router onto the workspace area and connect it to the switch using a serial cable. To do this, click on the "Connection" option, select "Serial," and then select the appropriate serial interface on the router.

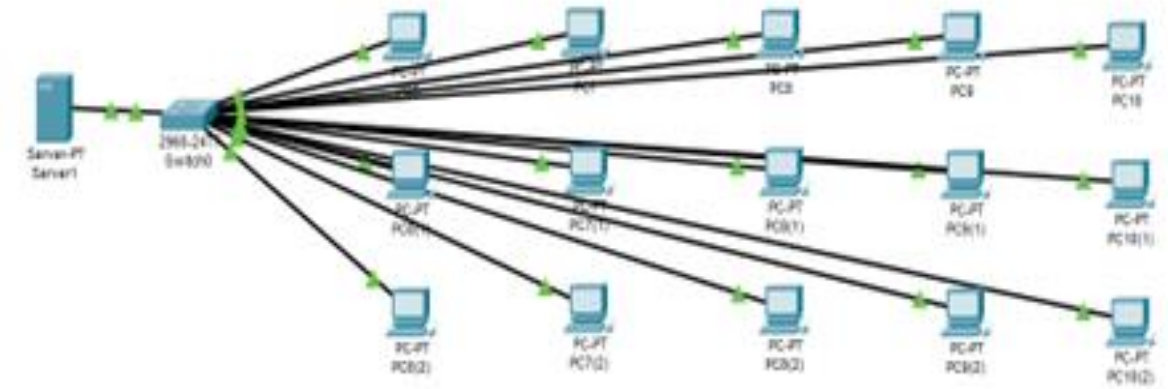
Step 7: Configure IP addresses on the computers. Select a computer, click on the "Desktop" tab in the device configuration panel, and configure the IP address, subnet mask, and default gateway for each computer.

Step 8: Configure IP addresses on the router interfaces. Select the router, click on the "CLI" tab in the device configuration panel, and enter the interface configuration mode. Assign IP addresses to the router interfaces connected to the switch and computers.

Step 9: Test connectivity. Open the command prompt on each computer and try to ping other computers and the router's interfaces to ensure connectivity.

Step 10: Customize and expand the lab as desired. You can add additional devices, configure VLANs, implement security measures, or set up servers within the lab environment.

Diagram:



Output:

Result: Thus the Computer Lab in Cisco Packet Tracer is set up successfully.

Date:

EXPERIMENT-11

CONFIGURATION OF A SIMPLE STATIC ROUTING IN PACKET TRACER USING A SIMPLE TOPOLOGY WITH TWO ROUTERS

Aim: To Configure a router using packet tracer software and hence to transmit data between the devices in real time mode and simulation mode.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Procedure:

Steps for building topology:

Step 1: Start Packet Tracer

Step 2: Choosing Devices and Connections

Step 3: Single click on the **End Devices**.

Single click on the **Generic** Host.

Place PC0, PC1 on topology area.

Connect PCs to Switch 1.

Similarly Place PC2, PC3 on topology area for receiver side

Connect these PCs with switch 1 and 2 respectively through connecting wires.

Select Router and place the router between two switches.

Connect these switches into router through connecting wires.

Step 3: Configuring IP Addresses, Gate Way and Subnet Masks on the Hosts

To start communication between the hosts IP Addresses, subnet Masks and Gate way had to be configured on the devices. Click once on PCs. Choose the Config tab and click on FastEthernet0. Type the IP address in its field. Based on router create gate way click on the subnet mask. It will be generated automatically.

Step 4: Verifying Connectivity in Real time Mode

Be sure you are in **Real time** mode.

Select the **Add Simple PDU** tool used to ping devices.

Click once on PC0, then once on PC3.

The PDU **Last Status** should show as **Successful**.

Step 5: **Verifying Connectivity in Simulation Mode**

Be sure you are in **Simulation** mode.

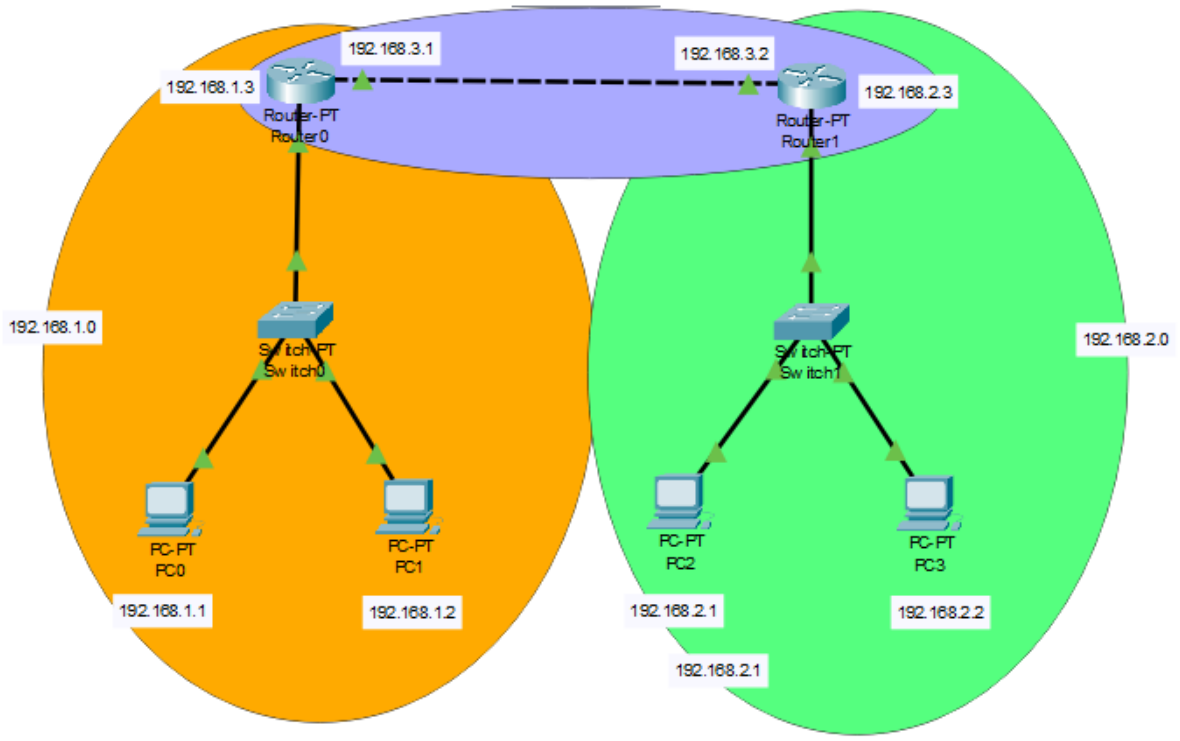
Deselect all filters (All/None) and select only **ICMP**.

Select the **Add Simple PDU** tool used to ping devices

Click once on PC0, then once on PC3.

Continue clicking **Capture/Forward** button until the ICMP ping is completed. The ICMP messages move between the hosts, hub and switch. The PDU **Last Status** should show as **Successful**.

Diagram:



Output:

Result: Thus Configuration of a simple static routing in packet tracer using a simple topology with two routers was done successfully.

Date:

EXPERIMENT-12

DESIGN THE FUNCTIONALITIES AND EXPLORATION OF TCP USING PACKET TRACER

Aim: To design the Functionalities and Exploration of TCP using Packet Tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Procedure:

Step 1: Setup the network topology

To begin, we will create a simple network topology consisting of two computers connected by a router. Open Packet Tracer and drag two PCs and a router onto the workspace. Connect the two PCs to the router using Ethernet cables.

Step 2: Configure IP addresses

Next, we will configure IP addresses for the computers. Double-click on each PC to open the configuration window and navigate to the Desktop tab. Click on the IP Configuration icon and enter the IP address and subnet mask for each computer. For example, PC1 can have an IP address of 192.168.1.1 with a subnet mask of 255.255.255.0 and PC2 can have an IP address of 192.168.1.2 with the same subnet mask.

Step 3: Configure the router

Now, we will configure the router. Double-click on the router to open the configuration window and navigate to the CLI tab.

COMMANDS:

enable

configure terminal

interface FastEthernet0/0

ip address 192.168.1.254 255.255.255.0

no shutdown

exit

exit

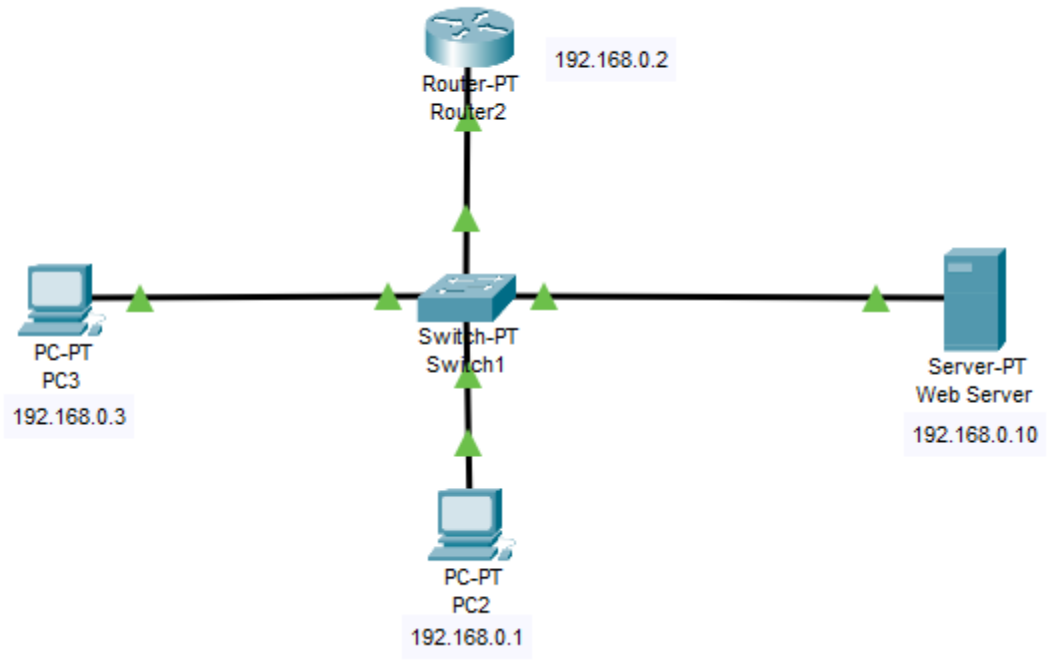
Step 4: Test the connection

Now that the network is set up and configured, we can test the connection between the two computers. Open a command prompt on PC1 and ping PC2 by typing ping 192.168.1.2 in the command prompt. If the ping is successful, it means that the two computers are communicating with each other.

Step 5: Explore TCP functionalities

Now, let's explore the functionalities of TCP. We will use the Netcat utility to establish a TCP connection between the two computers. Netcat is a versatile networking tool that can be used for various purposes, including establishing TCP connections.

Diagram



Output:

Result: Thus the Functionalities and Exploration of TCP using Packet Tracer is designed successfully.

Date:

EXPERIMENT-13

DESIGN THE NETWORK MODEL FOR SUBNETTING – CLASS C ADDRESSING USING PACKET TRACER

AIM: To design the network model for subnetting-class C addressing using packet tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Algorithm:

1. Determine the network requirements: Identify the number of subnets and hosts required for each subnet.
2. Choose a subnet mask: Select a subnet mask that can accommodate the required number of subnets and hosts.
3. Calculate the subnet mask and prefix length: Use the formula 2^p - 2 >= n, where p is the number of host bits and n is the required number of hosts per subnet, to calculate the number of host bits required. Add these host bits to the Class C network address to create the subnet address. The remaining bits in the subnet mask will be the prefix length.
4. Configure the router: Configure the router interface with the subnet address and subnet mask.
5. Configure the hosts: Configure each host with an IP address and subnet mask that matches the subnet address and subnet mask used on the router interface.
6. Test the network: Verify that the hosts can communicate with each other and with devices on other subnets.
7. Monitor network traffic: Use Packet Tracer's built-in network monitoring tools to monitor network traffic and identify any potential issue.

Procedure:

STEP 1: Click on end devices, select generic Pc's drag and drop it on the window. Click on SWITCH drag and drop it on the window.

STEP 2: Select the straight through cable and connect all end device to switch. Assign the IP address for all end devices. (Double click the end device Select → desktop → IP configuration static

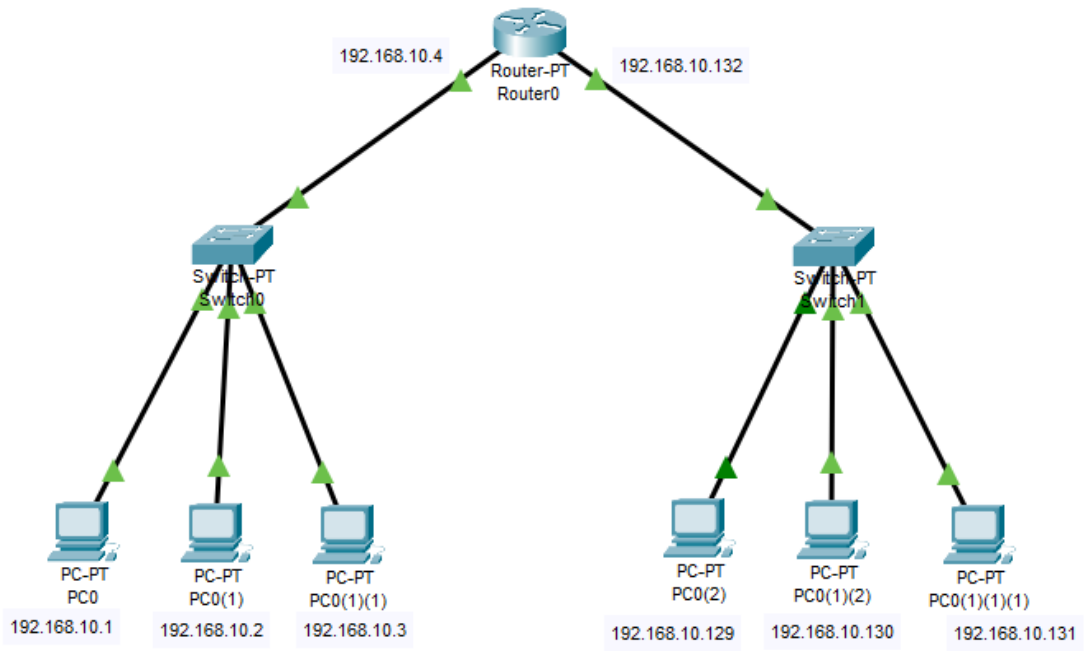
STEP 3: Now set the IP address to Host A (192.168.1.1) in static mode. Similarly set IP address

for Host B (192.168.1.2) and Host C (192.168.1.3)

STEP 4: To view the IP address, give ipconfig command in command prompt. Using ping command, we can establish communication between two host devices.

STEP 6: Now display the packet transmission in simulation mode.

Diagram



Output

Result:

There for designing for network model subnetting has been successfully implemented using packet tracer.

Date:

EXPERIMENT: 14

SIMULATING X, Y, Z COMPANY NETWORK DESIGN AND SIMULATE USING PACKET TRACER

Aim: To simulate X,Y,Z company network design and stimulate using packet tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Algorithm:

1. Identify the network requirements: Determine the number of users, devices, and servers that will be connected to the network.
2. Create a network diagram: Use a network diagramming tool to create a visual representation of the network design, including the devices, servers, switches, routers, and connections.
3. Configure the routers: Configure the routers with IP addresses, subnet masks, and routing protocols as needed.
4. Configure the switches: Configure the switches with VLANs, and assign ports to each VLAN.
5. Configure the servers: Configure the servers with IP addresses, subnet masks, and any necessary applications or services.
6. Configure the workstations: Configure the workstations with IP addresses, subnet masks, and any necessary applications or services.
7. Configure security: Configure security measures such as firewalls, access control lists, and intrusion detection systems as needed.
8. Test the network: Test the network connectivity by pinging devices and verifying that data can be transmitted between them.
9. Monitor network traffic: Use Packet Tracer's built-in network monitoring tools to monitor network traffic and identify any potential issues.
10. Make adjustments as needed: Make adjustments to the network configuration as needed to improve performance, security, or functionality.

Procedure:

1. Start Packet Tracer: Launch Packet Tracer on your computer.
2. Create a new project: Click on "File" and select "New", then select "Network" from the options.
3. Add devices: Click on the "Devices" tab in the bottom-left corner of the window, and

drag and drop devices onto the workspace. Add devices such as routers, switches, servers, and workstations.

4. Connect devices: Use the "Cable" tool to connect the devices together. Configure the connections as needed.

5. Configure devices: Double-click on each device to open its configuration menu, and configure its settings such as IP address, subnet mask, and routing protocols. Configure security measures such as firewalls, access control lists, and intrusion detection systems as needed.

6. Add applications: Click on the "Applications" tab in the bottom-left corner of the window, and drag and drop applications onto the workstations and servers. Configure the applications as needed.

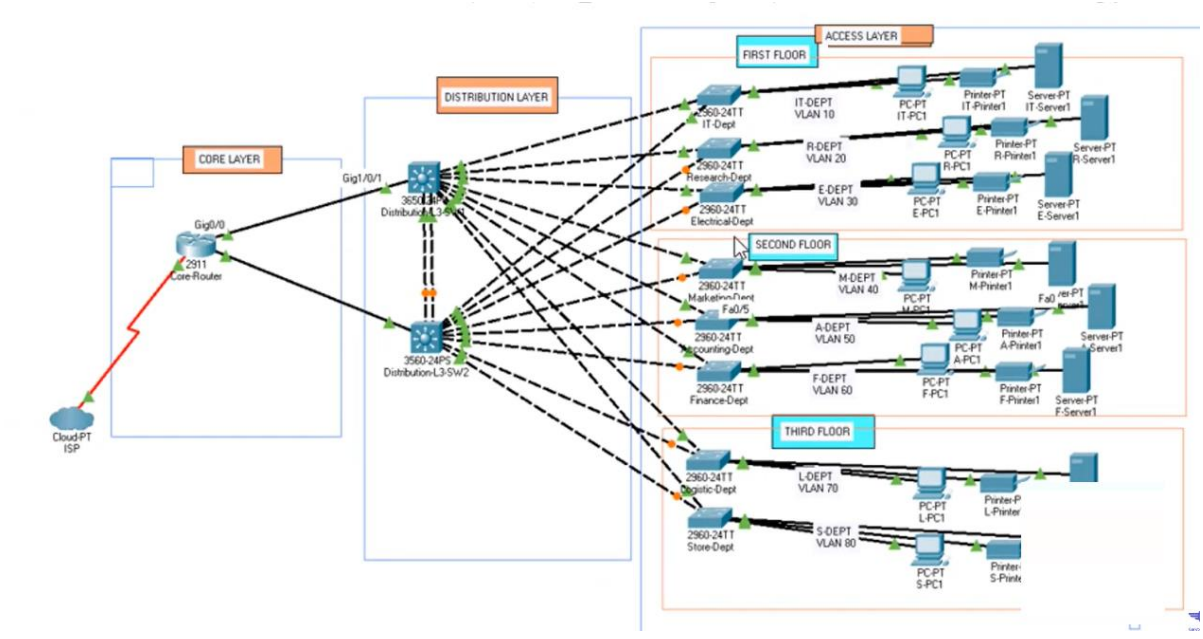
7. Test the network: Use Packet Tracer's built-in testing tools to verify that the network is working correctly. Test the network connectivity by pinging devices and verifying that data can be transmitted between them.

8. Monitor network traffic: Use Packet Tracer's built-in network monitoring tools to monitor network traffic and identify any potential issues.

9. Make adjustments as needed: Make adjustments to the network configuration as needed to improve performance, security, or functionality.

10. Save the project: Click on "File" and select "Save" to save the project.

Diagram



Output

Result: Therefore stimulating of companies network designing has been successfully done using packet tracer.

Date:

EXPERIMENT: 15

CONFIGURATION OF DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL) IN PACKET TRACER

Aim: To configure DHCP (dynamic host configuration protocol) in packet tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Algorithm:

1. Start:

• Set up the network topology in Packet Tracer with a DHCP server and DHCP clients connected to a switch.

2. Configure the DHCP server:

• Assign an IP address to the server interface.

• Enable the DHCP service on the server.

• Define the IP address pool range that the server can assign to clients.

• Specify additional DHCP options like default gateway, DNS server, and subnet mask.

3. Configure the switch.

• Enable the switch interfaces that connect to the DHCP clients.

4. Configure the DHCP clients.

• Configure the clients to obtain their IP addresses automatically using DHCP.

• Verify that the clients are set to use DHCP as the preferred method for IP assignment.

5. Client request and server response:

• When a DHCP client boots up or its lease expires, it sends a DHCP discover message as a broadcast on the local network.

• The DHCP server receives the discover message and responds with a DHCP offer message containing an available IP address from the configured IP address pool.

• The server includes other network configuration parameters in the offer message.

6. Client selection and request:

• The client receives multiple offer messages from different DHCP servers if available.

• The client selects one offer and sends a DHCP request message to the chosen server, requesting the offered IP address and confirming other network parameters.

7. Server acknowledgement:

• The DHCP server receives the request message and sends a DHCP acknowledge (ACK) message to the client, confirming the IP address assignment and providing additional network

configuration details.

8. Client configuration:

• The client receives the ACK message and configures its network interface with the assigned IP address, subnet mask, default gateway, DNS server, and any other parameters provided by the DHCP server.

9. Lease renewal and expiration:

• The client periodically contacts the DHCP server to renew its lease before it expires.

• If the client doesn't renew the lease or is unable to contact the DHCP server, the IP address lease eventually expires, and the IP address returns to the pool for future assignment.

10. End:

Procedure:

1. Launch Cisco Packet Tracer and create a new network topology or open an existing one.

2. Add the necessary network devices to your topology, including a DHCP server, switch, and DHCP clients. Connect them using appropriate cables.

3. Configure the DHCP server:

• Select the DHCP server device and open its configuration panel.

• Assign an IP address to the server interface connected to the switch.

• Enable the DHCP service on the server by checking the "DHCP" option.

• Define the IP address pool range that the server can assign to clients. Specify the starting and ending IP addresses.

• Optionally, set other DHCP options like default gateway, DNS server, and subnet mask.

• Save the configuration.

4. Configure the switch:

• Select the switch device and open its configuration panel.

• Enable the interfaces that connect to the DHCP clients. This allows the clients to communicate with the DHCP server.

• Save the configuration.

5. Configure the DHCP clients:

• Select each DHCP client device and open its configuration panel.

• Set the IP address assignment method to "DHCP" or "Obtain an IP address automatically."

• Save the configuration for each client.

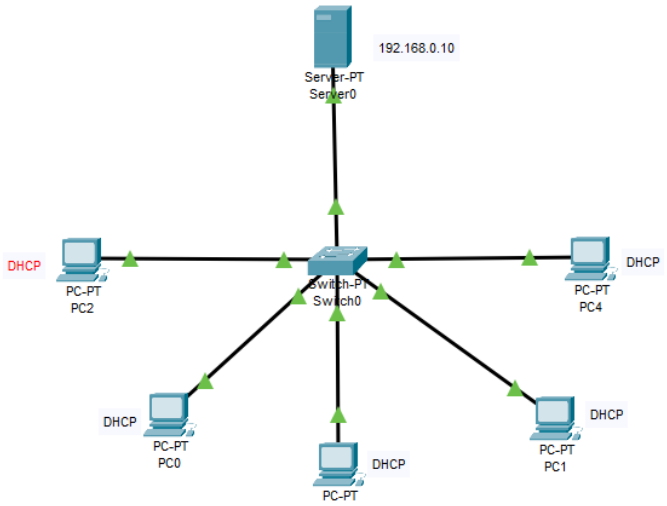
6. Start the simulation:

- Click the "Start/Stop Simulation" button to start the simulation.

7. Verify DHCP operation:

- Wait for the DHCP clients to boot up or refresh their IP configurations.
- Check if the DHCP clients receive IP addresses from the DHCP server.
- Verify that the clients have the correct IP address, subnet mask, default gateway, and DNS server settings.

Diagram



Output:

Result: Therefore the configuration for DHCP has been successfully executed using packet tracer.

Date:

EXPERIMENT-16

MAKING COMPUTER LAB IN CISCO PACKET TRACER

Aim: Making Computer Lab in Cisco Packet Tracer.

Software / Apparatus required: Packet Tracer / End devices, Switches, connectors.

Procedure:

Step 1: Launch Cisco Packet Tracer and create a new project.

Step 2: Select the appropriate network devices for your lab. In this case, you will need computers, switches, and routers. You can find these devices in the "End Devices," "Switches," and "Routers" sections of the device list.

Step 3: Drag and drop a switch onto the workspace area. Connect the switch to the power source by clicking on the "Connection" option and selecting "Power."

Step 4: Connect computers to the switch by dragging and dropping them onto the workspace area. Click on the "Connection" option and select "Fast Ethernet" to connect the computers to the switch.

Step 5: Repeat Step 4 to add more computers to the lab. You can adjust the number of computers as per your requirements.

Step 6: Connect the switch to a router. Drag and drop a router onto the workspace area and connect it to the switch using a serial cable. To do this, click on the "Connection" option, select "Serial," and then select the appropriate serial interface on the router.

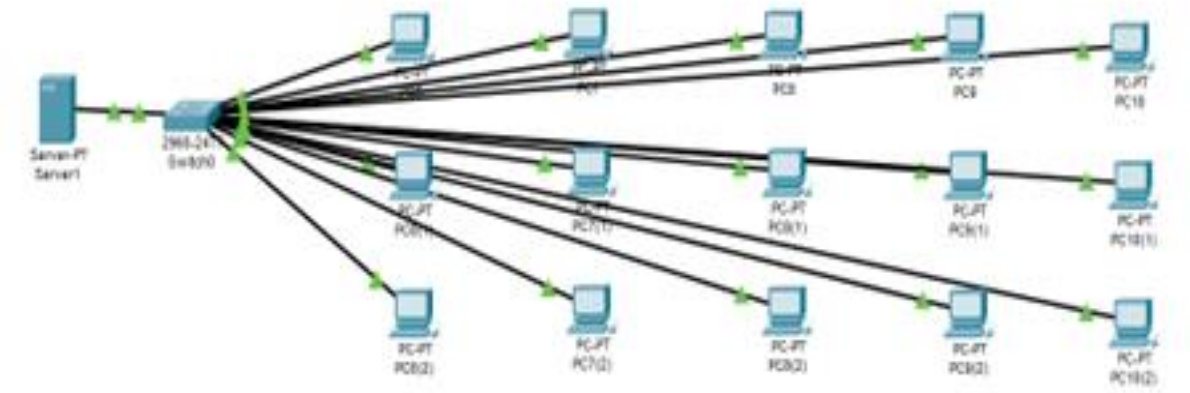
Step 7: Configure IP addresses on the computers. Select a computer, click on the "Desktop" tab in the device configuration panel, and configure the IP address, subnet mask, and default gateway for each computer.

Step 8: Configure IP addresses on the router interfaces. Select the router, click on the "CLI" tab in the device configuration panel, and enter the interface configuration mode. Assign IP addresses to the router interfaces connected to the switch and computers.

Step 9: Test connectivity. Open the command prompt on each computer and try to ping other computers and the router's interfaces to ensure connectivity.

Step 10: Customize and expand the lab as desired. You can add additional devices, configure VLANs, implement security measures, or set up servers within the lab environment.

Diagram:



Output:

Result: Thus the Computer Lab in Cisco Packet Tracer is set up successfully.

Date:

EXPERIMENT-17

CONFIGURATION OF FIREWALL IN PACKET TRACER

Aim: To configure firewall in packet tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Procedure:

Step 1: Set up the network topology

To begin, we will create a simple network topology consisting of three computers, a router, and a firewall. Open Packet Tracer and drag three PCs, a router, and a firewall onto the workspace. Connect the three PCs to the router using Ethernet cables, and connect the firewall to the router using another Ethernet cable.

Step 2: Configure IP addresses

Next, we will configure IP addresses for the computers. Double-click on each PC to open the configuration window and navigate to the Desktop tab. Click on the IP Configuration icon and enter the IP address and subnet mask for each computer. For example, PC1 can have an IP address of 192.168.1.1 with a subnet mask of 255.255.255.0, PC2 can have an IP address of 192.168.1.2 with the same subnet mask, and PC3 can have an IP address of 192.168.1.3 with the same subnet mask

Step 3: Configure the router

Now, we will configure the router. Double-click on the router to open the configuration window and navigate to the CLI tab. Enter the following commands:

Commands :

enable

configure terminal

interface FastEthernet0/0

ip address 192.168.1.254 255.255.255.0

no shutdown

exit

Step 4: Configure the firewall

Now, we will configure the firewall. Double-click on the firewall to open the configuration window

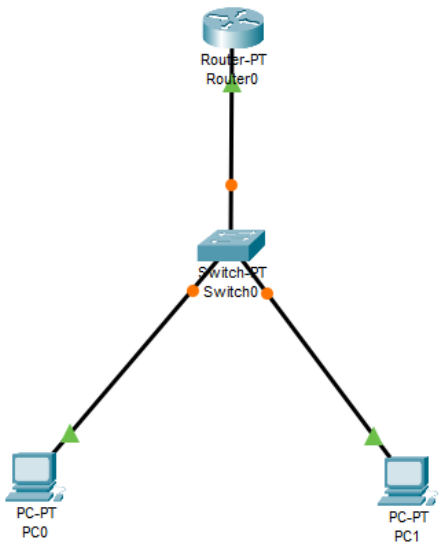
Step 5: Test the connection

Now that the firewall is configured, we can test the connection between the computers. Open a command prompt on PC1 and ping PC2 and PC3 by typing ping 192.168.1.2 and ping 192.168.1.3 in the command prompt. If the pings are successful, it means that the computers are communicating with each other.

Step 6: Test the firewall

To test the firewall, try to connect to PC1 from the internet using a protocol or port that is not allowed by the access rule. For example, you can try to connect to PC1 using Telnet on port 23.

Diagram



Output:

Result: Hence the configuration of firewall in packet tracer is successful.

Date:

EXPERIMENT-18

SIMULATE A MULTIMEDIA NETWORK IN CISCO PACKET TRACER

Aim: To simulate a Multimedia Network in Cisco Packet Tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Algorithm:

Procedure:

Step 1: Launch Cisco Packet Tracer and create a new project.

Step 2: Select the appropriate network devices for your multimedia network. You will need computers, switches, routers, and multimedia devices such as IP phones and IP cameras. You can find these devices in the "End Devices," "Switches," "Routers," "Phones," and "IP Cameras" sections of the device list.

Step 3: Design the network topology. Determine the layout of your network and the connections between devices. For example, you can connect the computers, IP phones, and IP cameras to a switch, and then connect the switch to a router for internet connectivity.

Step 4: Drag and drop the devices onto the workspace area. Connect the devices using appropriate cables or wireless connections. For example, use Ethernet cables to connect computers and IP phones to the switch.

Step 5: Configure IP addresses on the devices. Assign IP addresses, subnet masks, and default gateways to the computers, IP phones, and IP cameras. Configure the router's interface with an IP address provided by your ISP or use a DHCP server if available.

Step 6: Set up multimedia services. Configure the necessary services for multimedia communication, such as VoIP (Voice over IP) for IP phones and streaming protocols for IP cameras. This may involve configuring protocols like SIP (Session Initiation Protocol) for IP phones or RTSP (Real-Time Streaming Protocol) for IP cameras.

Step 7: Test connectivity and multimedia services. Verify that devices can communicate with each other and multimedia services are functioning correctly. For example, try making a call

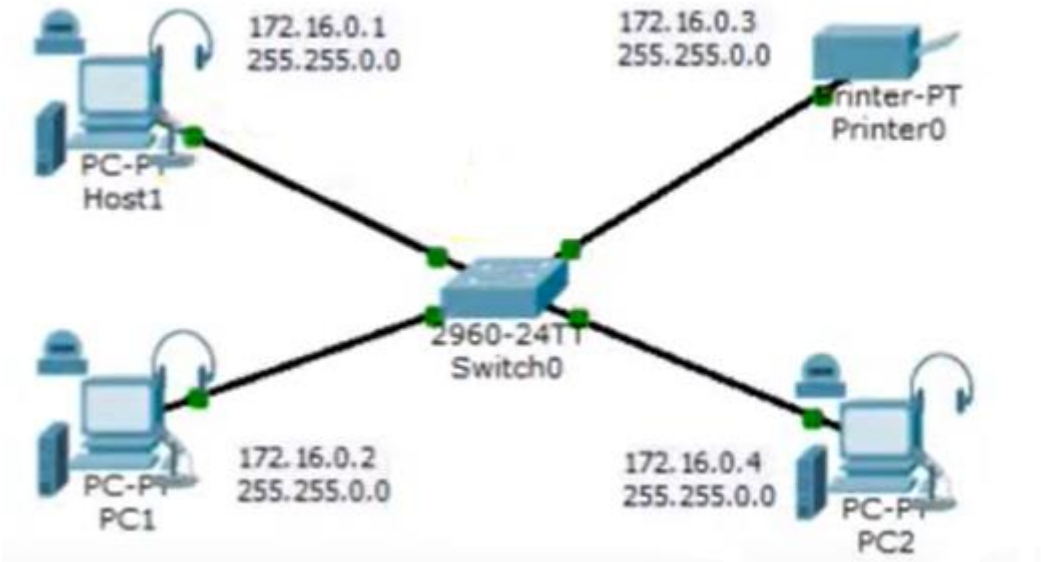
between IP phones or access the video feed from IP cameras.

Step 8: Monitor and troubleshoot. Use the network monitoring tools in Cisco Packet Tracer to observe network traffic and performance. Troubleshoot any issues that arise, such as connectivity problems or audio/video quality degradation.

Step 9: Document the lab experiment. Record observations, configurations, and any issues encountered during the simulation. This documentation will help to analyze the results and make improvements if necessary.

Remember to save your project regularly to preserve your progress. Cisco Packet Tracer provides a simulated environment to experiment with multimedia networks, allowing you to understand the challenges and requirements of such networks in a virtual setting.

Diagram



Result: Thus a Multimedia Network in Cisco Packet Tracer is simulated successfully.

Date:

EXPERIMENT-19
IOT BASED SMART HOME APPLICATIONS

Aim: To implement IoT based smart home applications in Cisco Packet Tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

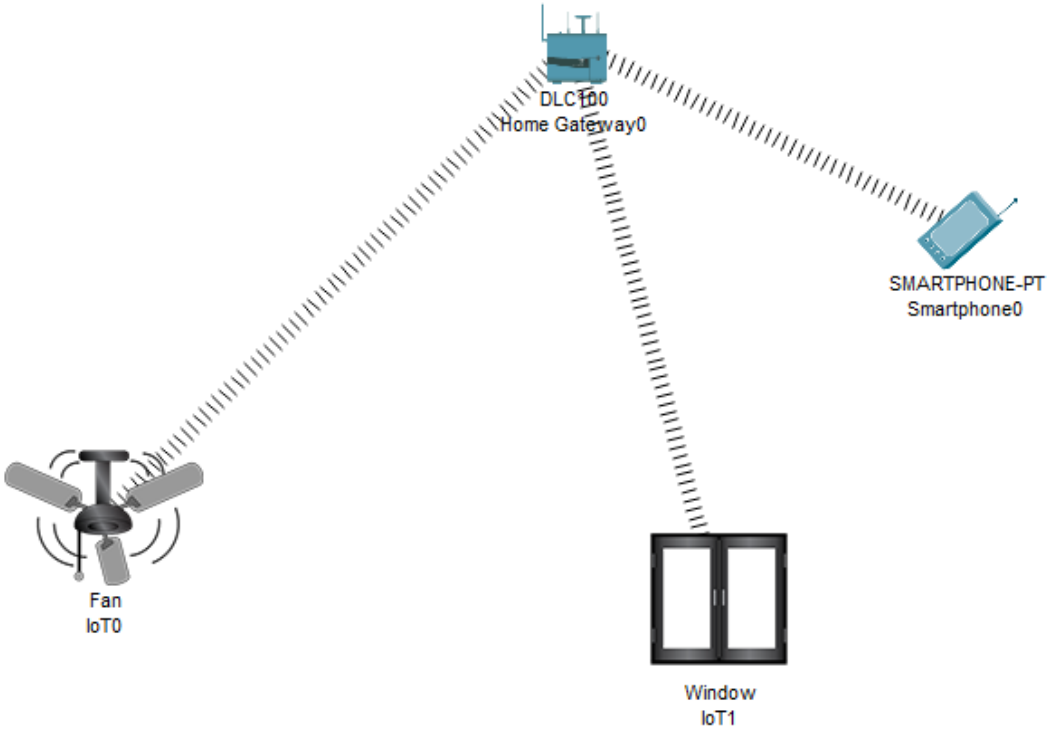
Procedure:

Steps:

1. Create a network topology in Cisco Packet Tracer that includes IoT devices such as sensors, actuators, and gateways.
2. Configure the IoT devices with appropriate IP addresses, subnet masks, and gateway addresses.
3. Set up a communication protocol between the IoT devices using MQTT, CoAP, or any other protocol of your choice.
4. Write a code to collect data from the sensors and send it to the gateway.
5. Use the gateway to process the data and send commands to the actuators.
6. Finally, use a web interface or mobile application to monitor and control the IoT devices.

By following these steps an IoT-based smart application in Cisco Packet Tracer , can be created. This can be used for various applications such as home automation, smart cities, and industrial automation.

Diagram



Output:

Result: Thus IoT based smart home applications in Cisco Packet Tracer is implemented successfully.

Date:

EXPERIMENT: 20

IMPLEMENTATION OF IOT BASED SMART GARDENING

Aim: To implement IOT based smart gardening using Cisco packet tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, Connectors.

Procedure:

Step 1: Create a new project in Cisco Packet Tracer and drag a generic IoT device from the IoT devices section onto the workspace.

Step 2: Right-click on the IoT device and select Config/Attributes.

Step 3: In the Configuration tab, select the device's IoT server from the drop-down list. You can choose Cisco IoT Cloud or another cloud service of your choice.

Step 4: In the Attributes tab, add the following attributes:

- Temperature
- Humidity
- Soil Moisture
- Light Intensity

Step 5: Create a soil moisture sensor and a light sensor from the Sensors section of the devices panel. Drag and drop these sensors onto the workspace.

Step 6: Connect the sensors to the IoT device using the wiring tool.

Step 7: Configure the sensors by right-clicking on them and selecting Config/Attributes. Set the sensor type, unit of measurement, and other necessary parameters.

Step 8: Create a water pump and a light bulb from the Actuators section of the devices panel. Drag and drop these actuators onto the workspace.

Step 9: Connect the actuators to the IoT device using the wiring tool.

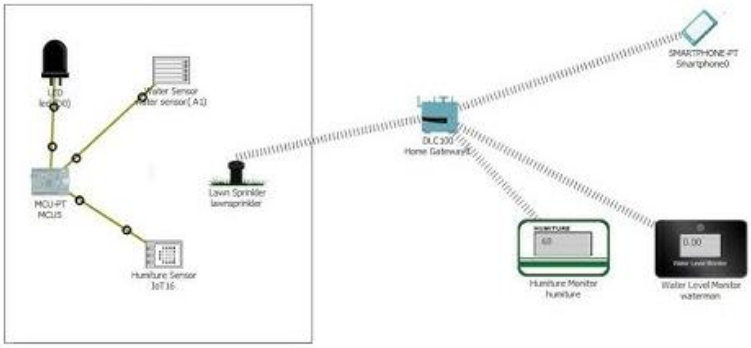
Step 10: Configure the actuators by right-clicking on them and selecting Config/Attributes. Set the actuator type, command, and other necessary parameters.

Step 11: Save the configuration and run the simulation to test your IoT Smart Garden.

Step 12: Monitor the temperature, humidity, soil moisture, and light intensity readings on the IoT device dashboard.

Step 13: Use the dashboard to control the water pump and light bulb based on the sensor readings.

Diagram:



Result: Implementation of smart gardening is carried out using IOT successfully.

Date:

EXPERIMENT: 21

IMPLEMENTATION OF IOT DEVICES IN NETWORKING

Aim: To implement an IOT devices in networking using Cisco Packet Tracer.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Procedure:

Steps:

1. Open Cisco Packet Tracer and create a new project.

Drag and drop a router from the "Devices" panel onto the workspace area.

2. Connect the router to the Internet by dragging and dropping a "Cloud" device from the "Devices" panel onto the workspace area, and then connecting the router to the cloud using a straight-through cable.

3. Add an IoT device to the network by dragging and dropping a device from the "Devices" panel onto the workspace area. There are various IoT devices available in the "Devices" panel, such as a Raspberry Pi or an Arduino.

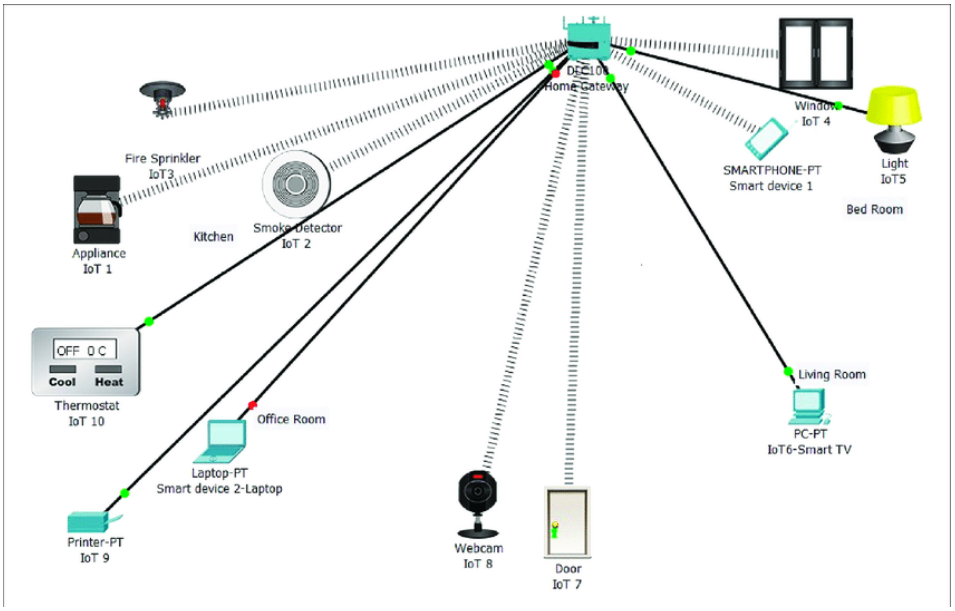
4. Connect the IoT device to the router using an Ethernet cable. To do this, click on the IoT device and then click on the "Config" tab. Under the "Interfaces" section, select the Ethernet interface and then click on the "+" button to add a new interface. Connect the new interface to the router.

5. Configure the IoT device by clicking on it and then clicking on the "CLI" tab. This will bring up the command line interface for the IoT device, where you can configure its settings.

6. Test the connectivity of the IoT device by pinging it from the router or from another device on the network.

7. These are just general steps and the specifics of the implementation will depend on the specific IoT device and network configuration you want to create. Additionally, you may need to configure the router and the cloud device to enable Internet connectivity for the IoT device.

Diagram:



Result: Thus an IOT device in networking is implemented using Cisco Packet Tracer successfully.

Date:

EXPERIMENT: 22

IoT based AAA Local and Server based authentication configuration

Aim: Designing an IoT based AAA Local and Server based authentication configuration.

Software/Apparatus required: Packet Tracer/End devices, Hubs, connectors.

Procedure:

Algorithm:

1. Define the Components:

IoT Devices: These are the devices that need to be authenticated and authorized to access the network resources.

Local AAA Server: This server will handle authentication and authorization requests locally.

Central AAA Server: This server will provide an additional layer of authentication and authorization for higher-level access control.

2. Setup Local AAA Server:

Configure the local AAA server with the necessary software and databases to handle authentication and authorization requests.

Define user profiles or roles and their associated permissions on the local AAA server.

Set up a secure communication channel between the IoT devices and the local AAA server.

3. Implement Local Authentication:

When an IoT device wants to connect to the network, it sends an authentication request to the local AAA server.

The local AAA server verifies the credentials provided by the IoT device against its user database.

If the credentials are valid, the local AAA server generates an authentication token or session key and sends it back to the IoT device.

4. Implement Local Authorization:

Once authenticated, the IoT device sends an authorization request to the local AAA server.

The local AAA server checks the user profile or role associated with the IoT device and verifies if it has the necessary permissions to access the requested resources.

If authorized, the local AAA server sends an authorization response to the IoT device.

5. Configure Central AAA Server:

Set up a central AAA server that will provide an additional layer of authentication and authorization for critical resources or higher-level access control.

Configure the central AAA server with user profiles or roles and associated permissions.

6. Implement Server Authentication:

After local authentication, the IoT device establishes a secure connection with the central AAA server.

The IoT device sends its authentication token or session key to the central AAA server for verification.

The central AAA server validates the authentication token or session key received from the IoT device.

7. Implement Server Authorization:

Once the central AAA server verifies the authentication token or session key, it performs additional authorization checks.

The central AAA server ensures that the IoT device has the necessary permissions to access critical resources or perform higher-level operations.

If authorized, the central AAA server sends an authorization response to the IoT device.

8. Logging and Accounting:

Both the local and central AAA servers maintain logs of authentication and authorization events for auditing and accounting purposes.

They record information such as user/device identification, timestamps, and actions taken.

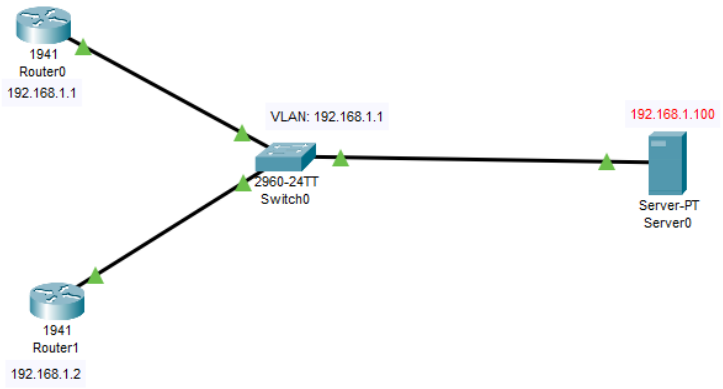
9. Revocation and Updates:

Implement mechanisms to handle credential revocation, such as disabling user accounts or tokens when necessary.

Regularly update user profiles, roles, and permissions in both the local and central AAA servers to reflect changes in the network environment.

Remember to implement appropriate security measures such as encryption, secure communication protocols, and strong password policies to ensure the integrity and confidentiality of the authentication process.

Diagram:



Output:

Result: IoT based AAA Local and Server based authentication is designed successfully.

EXPERIMENT-23

DESIGN THE FUNCTIONALITIES AND EXPLORATION OF UDP USING PACKET TRACER

Aim:

To design the functionalities and exploration of UDP (User Datagram Protocol) using Packet Tracer.

Software/Apparatus required:

Packet Tracer, End devices (PCs), Router, Switch, Server, Ethernet cables.

Procedure:

Step 1: Setup the network topology

1. Open Packet Tracer and create a network topology as shown in the diagram.
2. Drag the following devices onto the workspace:
    - Router0 (ISR 331)
    - Switch0 (Switch-PT)
    - Server0 (Server-PT) with IP address 192.168.1.10
    - PC0 (PC-PT) with IP address 192.168.1.1
    - PC1 (PC-PT) with IP address 192.168.1.2
3. Connect the devices as follows:
    - Connect PC0 and PC1 to Switch0 using Ethernet cables.
    - Connect Switch0 to Router0.
    - Connect Server0 to Router0.

Step 2: Configure IP addresses

1. Double-click on each PC and the server to open the configuration window.
2. Navigate to the Desktop tab and click on the IP Configuration icon.
3. Assign IP addresses and subnet masks:
    - PC0: IP address = 192.168.1.1, Subnet mask = 255.255.255.0
    - PC1: IP address = 192.168.1.2, Subnet mask = 255.255.255.0
    - Server0: IP address = 192.168.1.10, Subnet mask = 255.255.255.0

Step 3: Configure the router

1. Double-click on Router0 to open the configuration window.
2. Navigate to the CLI tab and enter the following commands:

```
enable
configure terminal
interface FastEthernet0/0
ip address 192.168.1.254 255.255.255.0
```

no shutdown

exit

exit

This configures the router's interface with the IP address 192.168.1.254 and enables it.

Step 4: Test the connection

1. Open the command prompt on PC0 and ping PC1 by typing:

ping 192.168.1.2

2. Open the command prompt on PC1 and ping Server0 by typing:

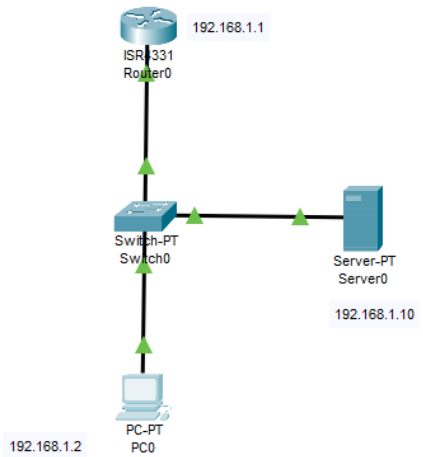
ping 192.168.1.10

3. If the pings are successful, it confirms that the devices are communicating.

Step 5: Explore UDP functionalities

1. Use a UDP-based application or utility (e.g., a simple UDP sender/receiver script or a network tool like Netcat) to simulate UDP communication.
2. On PC0, set up a UDP sender to send data to Server0 on a specific port (e.g., port 5000).
3. On Server0, set up a UDP receiver to listen on the same port (5000).
4. Observe the data transmission. Note that UDP does not guarantee delivery, order, or error-checking, unlike TCP.

Diagram



Output:

Result:

Thus, the functionalities and exploration of UDP using Packet Tracer were designed successfully.

EXPERIMENT-24

DESIGNING TWO DIFFERENT NETWORKS WITH DYNAMIC ROUTING TECHNIQUES (RIP & OSPF) USING PACKET TRACER

Aim:

To design two different networks using dynamic routing protocols (RIP and OSPF) and analyze their functionalities using Packet Tracer.

Software/Apparatus required:

Packet Tracer, Routers (ISR 331, Router-PT), Switches (2560-2XT), PCs, Ethernet cables.

Procedure:

Network 1: Dynamic Routing using RIP

Step 1: Setup the network topology

1. Open Packet Tracer and create the first network topology as shown in Diagram 1:
    - Router0 (ISR 331)
    - Router1 (ISR 331)
    - Switch0 (2560-2XT) connected to Router0
    - Switch1 (2560-2XT) connected to Router1
    - PC0 and PC1 connected to Switch0
    - PC2 and PC3 connected to Switch1

Step 2: Configure IP addresses

1. Assign IP addresses to the PCs:
    - PC0: 192.168.1.1/24
    - PC1: 192.168.1.2/24
    - PC2: 192.168.2.1/24
    - PC3: 192.168.2.2/24
2. Configure the router interfaces:
    - Router0 (ISR 331):
        - Interface connected to Switch0: 192.168.1.254/24
        - Interface connected to Router1: 10.0.0.1/30
    - Router1 (ISR 331):
        - Interface connected to Switch1: 192.168.2.254/24
        - Interface connected to Router0: 10.0.0.2/30

Step 3: Configure RIP routing

1. On Router0, enter the following commands:

enable

configure terminal

router rip

version 2

network 192.168.1.0

network 10.0.0.0

no auto-summary

exit

2. On Router1, enter the following commands:

enable

configure terminal

router rip

version 2

network 192.168.2.0

network 10.0.0.0

no auto-summary

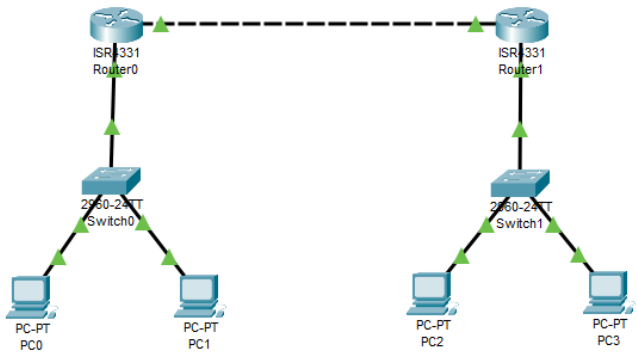
exit

Step 4: Test the connection

1. Use the ping command to test connectivity between PCs across the network.
    - For example, ping PC2 from PC0:

ping 192.168.2.1

Diagram



Output :

Network 2: Dynamic Routing using OSPF

Step 1: Setup the network topology

1. Open Packet Tracer and create the second network topology as shown in Diagram 2:
    - Router3 (Router-PT)
    - Router4 (Router-PT)
    - Router5 (Router-PT)
    - Router6 (Router-PT)
    - Router7 (Router-PT)
    - PC0, PC1, PC2, and PC3 connected to respective routers.

Step 2: Configure IP addresses

1. Assign IP addresses to the PCs:
    - PC0: 192.168.10.1/24
    - PC1: 192.168.20.1/24
    - PC2: 192.168.30.1/24
    - PC3: 192.168.40.1/24
2. Configure the router interfaces:
    - Assign IP addresses to all router interfaces based on the network design.

Step 3: Configure OSPF routing

1. On each router, enable OSPF and advertise the connected networks. For example, on Router3:

enable

configure terminal

router ospf 1

network 192.168.10.0 0.0.0.255 area 0

network <connected network> <wildcard mask> area 0

exit

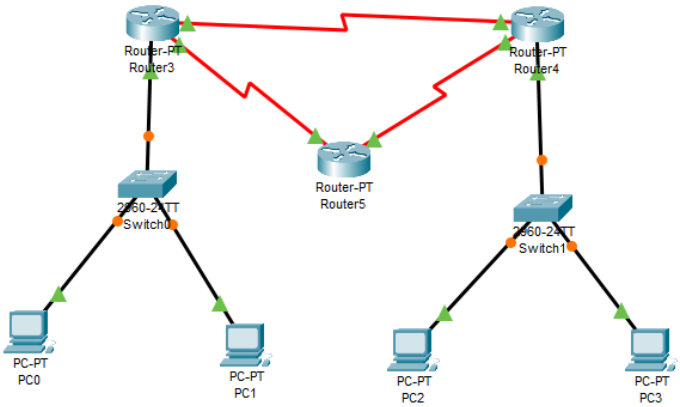
2. Repeat the OSPF configuration on all routers, ensuring all networks are advertised in Area 0.

Step 4: Test the connection

1. Use the ping command to test connectivity between PCs across the network.
    - For example, ping PC3 from PC0:

ping 192.168.40.1

Diagram:



Output:

Result:

Thus, two different networks using dynamic routing techniques (RIP and OSPF) were designed and analyzed successfully using Packet Tracer.

Date:

EXPERIMENT: 25

TRANSPORT LAYER PROTOCOL HEADER ANALYSIS USING WIRE SHARK-TCP

Aim: To analyze capturing of Transport layer protocol header analysis using Wire shark- TCP

SOFTWARE USED:

Wire shark network analyzer

Procedure:

1. Open wire shark.
2. Click on list the available capture interface.
3. Choose the LAN interface.
4. Click on start button.
5. Active packets will be displayed.
6. Capture the packets & select any IP address from the source.
7. Click on the expression and select IPV4 → IP addr source address in the field name.
8. Select the double equals (==) from the selection and enter the selected IP source address.
9. Click on apply button.
10. All the packets will be filtered using source address.

Result: Hence, the capturing of packets using wire shark network analyzer was analyzed for TCP

Date:

EXPERIMENT: 26

TRANSPORT LAYER PROTOCOL HEADER ANALYSIS USING WIRE SHARK- UDP

Aim: To analyze capturing of Transport layer protocol header analysis using Wire shark- UDP.

SOFTWARE USED:

Wire shark network analyzer

Procedure:

1. Open wire shark.
2. Click on list the available capture interface.
3. Choose the LAN interface.
4. Click on start button.
5. Active packets will be displayed.
6. Capture the packets & select any IP address from the source.
7. Click on the expression and select IPV4 → IP addr source address in the field name.
8. Select the double equals (==) from the selection and enter the selected IP source address.
9. Click on apply button.
10. All the packets will be filtered using source address.

Result: Hence, the capturing of packets using wire shark network analyzer was analyzed for UDP.

Date:

EXPERIMENT-27

NETWORK LAYER PROTOCOL HEADER ANALYSIS USING WIRE SHARK – SMTP

Aim: To analyze capturing of Transport layer protocol header analysis using Wire shark- SMTP

SOFTWARE USED:

Wire shark network analyzer

Procedure:

1. Open wire shark.
2. Click on list the available capture interface.
3. Choose the LAN interface.
4. Click on start button.
5. Active packets will be displayed.
6. Capture the packets & select any IP address from the source.
7. Click on the expression and select IPV4 → IP addr source address in the field name.
8. Select the double equals (==) from the selection and enter the selected IP source address.
9. Click on apply button.
10. All the packets will be filtered using source address.

Result: Hence, the capturing of packets using wire shark network analyzer was analyzed for SMTP

Date:

EXPERIMENT-28

NETWORK LAYER PROTOCOL HEADER ANALYSIS USING WIRE SHARK –ICMP

Aim: To analyze capturing of Transport layer protocol header analysis using Wire shark- ICMP.

SOFTWARE USED:

Wire shark network analyzer

Procedure:

1. Open wire shark.
2. Click on list the available capture interface.
3. Choose the LAN interface.
4. Click on start button.
5. Active packets will be displayed.
6. Capture the packets & select any IP address from the source.
7. Click on the expression and select IPV4 → IP addr source address in the field name.
8. Select the double equals (==) from the selection and enter the selected IP source address.
9. Click on apply button.
10. All the packets will be filtered using source address.

Result: Hence, the capturing of packets using wire shark network analyzer was analyzed for ICMP.

Date:

EXPERIMENT-29

NETWORK LAYER PROTOCOL HEADER ANALYSIS USING WIRE SHARK – ARP

AIM: To analyze capturing of Transport layer protocol header analysis using Wire shark- ARP

SOFTWARE USED:

Wire shark network analyzer

PROCEDURE:

1. Open wire shark.
2. Click on list the available capture interface.
3. Choose the LAN interface.
4. Click on start button.
5. Active packets will be displayed.
6. Capture the packets & select any IP address from the source.
7. Click on the expression and select IPV4 → IP addr source address in the field name.
8. Select the double equals (==) from the selection and enter the selected IP source address.
9. Click on apply button.
10. All the packets will be filtered using source address.

Result: Hence, the capturing of packets using wire shark network analyzer was analyzed for ARP

Date:

EXPERIMENT-30

NETWORK LAYER PROTOCOL HEADER ANALYSIS USING WIRE SHARK –HTTP

AIM: To analyze capturing of Transport layer protocol header analysis using Wire shark- HTTP.

SOFTWARE USED:

Wire shark network analyzer

PROCEDURE:

1. Open wire shark.
2. Click on list the available capture interface.
3. Choose the LAN interface.
4. Click on start button.
5. Active packets will be displayed.
6. Capture the packets & select any IP address from the source.
7. Click on the expression and select IPV4 →IP addr source address in the field name.
8. Select the double equals (==) from the selection and enter the selected IP source address.
9. Click on apply button.
10. All the packets will be filtered using source address.

Result: Hence, the capturing of packets using wire shark network analyzer was analyzed for HTTP.

Date:

EXPERIMENT: 31

IMPLEMENTATION OF SERVER – CLIENT USING TCP SOCKET PROGRAMMING

Aim:

To implement a server-client communication model using TCP socket programming in C.

Software/Apparatus Required:

- C Compiler (GCC or any compatible compiler)
- Linux-based OS (or any OS supporting POSIX sockets)
- Text editor (e.g., Vim, Nano, or any IDE)

Procedure:

Step 1: Write the Server-Side Code

1. Open a text editor and write the server-side C program as provided.
2. Save the file as server.c.

Step 2: Write the Client-Side Code

1. Open a text editor and write the client-side C program as provided.
2. Save the file as client.c.

Step 3: Compile the Programs

1. Open the terminal and navigate to the directory containing the server.c and client.c files.
2. Compile the server program using the following command:

gcc server.c -o server

3. Compile the client program using the following command:

gcc client.c -o client

Step 4: Run the Server

1. Execute the server program using the following command:

./server

2. The server will start listening on port 8080.

Step 5: Run the Client

1. Open another terminal window and navigate to the same directory.
2. Execute the client program using the following command:

./client

3. The client will connect to the server running on 127.0.0.1 (localhost) and port 8080.

Step 6: Test the Communication

1. On the client side, type a message and press Enter. The message will be sent to the server.
2. The server will receive the message, display it, and prompt for a response.
3. The server's response will be sent back to the client and displayed on the client's terminal.
4. To end the communication, type "exit" on either the client or server side.

Code:

//SERVER SIDE

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h> // read(), write(), close()
#define MAX 80
#define PORT 8080
#define SA struct sockaddr

// Function designed for chat between client and server.
void func(int connfd)
{
    char buff[MAX];
    int n;
    // infinite loop for chat
    for (;;) {
        bzero(buff, MAX);

        // read the message from client and copy it in buffer
        read(connfd, buff, sizeof(buff));
        // print buffer which contains the client contents
```

```
        printf("From client: %s\t To client : ", buff);
        bzero(buff, MAX);
        n = 0;
        // copy server message in the buffer
        while ((buff[n++] = getchar()) != '\n')
            ;

        // and send that buffer to client
        write(connfd, buff, sizeof(buff));

        // if msg contains "Exit" then server exit and chat ended.
        if (strncmp("exit", buff, 4) == 0) {
            printf("Server Exit...\n");
            break;
        }
    }
}

// Driver function
int main()
{
    int sockfd, connfd, len;
    struct sockaddr_in servaddr, cli;

    // socket create and verification
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
        printf("socket creation failed...\n");
        exit(0);
    }
    else
        printf("Socket successfully created..\n");
    bzero(&servaddr, sizeof(servaddr));
```

```
// assign IP, PORT
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
servaddr.sin_port = htons(PORT);

// Binding newly created socket to given IP and verification
if ((bind(sockfd, (SA*)&servaddr, sizeof(servaddr))) != 0) {
    printf("socket bind failed...\n");
    exit(0);
}
else
    printf("Socket successfully binded..\n");

// Now server is ready to listen and verification
if ((listen(sockfd, 5)) != 0) {
    printf("Listen failed...\n");
    exit(0);
}
else
    printf("Server listening..\n");
len = sizeof(cli);

// Accept the data packet from client and verification
connfd = accept(sockfd, (SA*)&cli, &len);
if (connfd < 0) {
    printf("server accept failed...\n");
    exit(0);
}
else
    printf("server accept the client...\n");

// Function for chatting between client and server
func(connfd);
```

```
    // After chatting close the socket
    close(sockfd);
}
```

//CLIENT SIDE

```
// Online C compiler to run C program online
#include <arpa/inet.h> // inet_addr()
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h> // bzero()
#include <sys/socket.h>
#include <unistd.h> // read(), write(), close()
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void func(int sockfd)
{
    char buff[MAX];
    int n;
    for (;;) {
        bzero(buff, sizeof(buff));
        printf("Enter the string : ");
        n = 0;
        while ((buff[n++] = getchar()) != '\n')
            ;
        write(sockfd, buff, sizeof(buff));
        bzero(buff, sizeof(buff));
        read(sockfd, buff, sizeof(buff));
        printf("From Server : %s", buff);
        if ((strncmp(buff, "exit", 4)) == 0) {
            printf("Client Exit...\n");
```

```
            break;
        }
    }
}

int main()
{
    int sockfd, connfd;
    struct sockaddr_in servaddr, cli;

    // socket create and verification
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
        printf("socket creation failed...\n");
        exit(0);
    }
    else
        printf("Socket successfully created..\n");
    bzero(&servaddr, sizeof(servaddr));

    // assign IP, PORT
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);

    // connect the client socket to server socket
    if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr))
        != 0) {
        printf("connection with the server failed...\n");
        exit(0);
    }
    else
        printf("connected to the server..\n");
```

```
    // function for chat
    func(sockfd);

    // close the socket
    close(sockfd);
}
```

Output:

1. Server-side output:

Copy

Socket successfully created..

Socket successfully binded..

Server listening..

server accept the client...

From client: <Client Message> To client: <Server Response>

2. Client-side output:

Copy

Socket successfully created..

connected to the server..

Enter the string: <Client Message>

From Server: <Server Response>

Result:

Thus, the server-client communication using TCP socket programming was implemented successfully.

EXPERIMENT-32

IMPLEMENTATION OF SERVER – CLIENT USING UDP SOCKET PROGRAMMING

Aim:

To implement a server-client communication model using UDP socket programming in C.

Software/Apparatus Required:

- C Compiler (GCC or any compatible compiler)
- Linux-based OS (or any OS supporting POSIX sockets)
- Text editor (e.g., Vim, Nano, or any IDE)

Procedure:

Step 1: Write the Server-Side Code

1. Open a text editor and write the server-side C program as provided.
2. Save the file as udp_server.c.

Step 2: Write the Client-Side Code

1. Open a text editor and write the client-side C program as provided.
2. Save the file as udp_client.c.

Step 3: Compile the Programs

1. Open the terminal and navigate to the directory containing the udp_server.c and udp_client.c files.
2. Compile the server program using the following command:

```
gcc udp_server.c -o udp_server
```

3. Compile the client program using the following command:

```
gcc udp_client.c -o udp_client
```

Step 4: Run the Server

1. Execute the server program using the following command:

```
./udp_server
```

2. The server will start listening on port 5000.

Step 5: Run the Client

1. Open another terminal window and navigate to the same directory.
2. Execute the client program using the following command:

```
./udp_client
```

3. The client will send a message to the server running on 127.0.0.1 (localhost) and port 5000.

Step 6: Test the Communication

1. The client sends a message ("Hello Server") to the server.
2. The server receives the message, prints it, and sends a response ("Hello Client") back to the client.
3. The client receives the server's response and prints it.

Code:

Implementation of server – client using UDP socket programming

```
// server program for udp connection
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include<netinet/in.h>
#define PORT 5000
#define MAXLINE 1000

// Driver code
int main()
{
   char buffer[100];
   char *message = "Hello Client";
   int listenfd, len;
   struct sockaddr_in servaddr, cliaddr;
   bzero(&servaddr, sizeof(servaddr));

   // Create a UDP Socket
   listenfd = socket(AF_INET, SOCK_DGRAM, 0);
   servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
   servaddr.sin_port = htons(PORT);
   servaddr.sin_family = AF_INET;

   // bind server address to socket descriptor
```

```
    bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));

    //receive the datagram
    len = sizeof(cliaddr);
    int n = recvfrom(listenfd, buffer, sizeof(buffer),
            0, (struct sockaddr*)&cliaddr,&len); //receive message from server
    buffer[n] = '\0';
    puts(buffer);

    // send the response
    sendto(listenfd, message, MAXLINE, 0,
          (struct sockaddr*)&cliaddr, sizeof(cliaddr));
}


// udp client driver program
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include<netinet/in.h>
#include<unistd.h>
#include<stdlib.h>

#define PORT 5000
#define MAXLINE 1000

// Driver code
int main()
{
    char buffer[100];
    char *message = "Hello Server";
    int sockfd, n;
```

```
    struct sockaddr_in servaddr;

    // clear servaddr
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;

    // create datagram socket
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    // connect to server
    if(connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
    {
        printf("\n Error : Connect Failed \n");
        exit(0);
    }

    // request to send datagram
    // no need to specify server address in sendto
    // connect stores the peers IP and port
    sendto(sockfd, message, MAXLINE, 0, (struct sockaddr*)NULL, sizeof(servaddr));

    // waiting for response
    recvfrom(sockfd, buffer, sizeof(buffer), 0, (struct sockaddr*)NULL, NULL);
    puts(buffer);

    // close the descriptor
    close(sockfd);
}
```

Output:

1. Server-side output:

Hello Server

2. Client-side output:

Hello Client

Result:

Thus, the server-client communication using UDP socket programming was implemented successfully.

EXPERIMENT-33
IMPLEMENTATION OF BIT STUFFING MECHANISM USING C

Aim:
To implement the bit stuffing mechanism using the C programming language.

Software/Apparatus required:
C compiler (e.g., GCC), Code editor (e.g., VS Code, Dev C++).

Procedure:

Step 1: Understand the Bit Stuffing Mechanism

1. Bit stuffing is a technique used in data communication to ensure that a specific pattern (e.g., five consecutive 1s) is not mistaken for a control signal.
2. If five consecutive 1s are detected, a 0 is stuffed (inserted) after them to differentiate the data from control signals.

Step 2: Write the C Program

1. Open a code editor and write the following C program to implement bit stuffing:

Step 3: Compile and Run the Program

1. Save the program with a .c extension (e.g., bit_stuffing.c).
2. Compile the program using a C compiler.
3. Run the compiled program
4. **Step 4: Analyze the Output**

The program will output the stuffed bit sequence.
For the input array {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}, the output will be:
111110111110
Here, a 0 is stuffed after every five consecutive 1s.

Program

```
#include <stdio.h>
#include <string.h>

// Function for bit stuffing
void bitStuffing(int N, int arr[])
{
   // Stores the stuffed array
   int brr[30];

   // Variables to traverse arrays
   int i, j, k;
   i = 0;
   j = 0;

   // Loop to traverse in the range [0, N)
   while (i < N) {

      // If the current bit is a set bit
      if (arr[i] == 1) {

         // Stores the count of consecutive ones
         int count = 1;

         // Insert into array brr[]
         brr[j] = arr[i];

         // Loop to check for
         // next 5 bits
```

```
            for (k = i + 1;
                 arr[k] == 1 && k < N && count < 5; k++) {
                j++;
                brr[j] = arr[k];
                count++;

                // If 5 consecutive set bits
                // are found insert a 0 bit
                if (count == 5) {
                    j++;
                    brr[j] = 0;
                }
                i = k;
            }
        }

        // Otherwise insert arr[i] into
        // the array brr[]
        else {
            brr[j] = arr[i];
        }
        i++;
        j++;
    }

    // Print Answer
    for (i = 0; i < j; i++)
        printf("%d", brr[i]);
}
 // Driver Code
int main()
{
    int N = 12;
    int arr[] = { 1, 1, 1, 1, 1, 1,1,1,1,1,1,1 };

    bitStuffing(N, arr);

    return 0;
}
```

Output:
111110111110

Result:
Thus, the bit stuffing mechanism was successfully implemented using the C programming language.