# Computer Networks Lab Record

# (Course Code: 20CS3552)

by

**G. Venkata Sai Ram (22501A0557)**

**III B.Tech I Sem Section-1**

**in**

**Computer Science and Engineering**

## Prasad V Potluri Siddhartha Institute of Technology

Accredited with A+ grade by NAAC, NBA Accredited, and Autonomous

ISO 9001:2015 Certified Institute

Permanently Affiliated to JNTUK-Kakinada and approved by AICTE

**Kanuru, Vijayawada-520 007**

# Prasad V Potluri Siddhartha Institute of Technology

Accredited with A+ grade by NAAC, NBA Accredited, and Autonomous

ISO 9001:2015 Certified Institute

Permanently Affiliated to JNTUK-Kakinada and approved by AICTE

**Kanuru, Vijayawada-520 007**

# CERTIFICATE

This is to certify that Mr. / Ms. _____ bearing Reg. no. _____ of III B. Tech I semester **Computer Science and Engineering** branch has satisfactorily completed **Computer Networks Lab Record (Course Code: (20CS3552))** during the academic year 2024-25.

Signature of Faculty

Signature of HOD

Signature of Internal Examiner

Signature of External Examiner

# Index

| Exp. No. | Title of the Experiment | Date | Page No. | Marks | | Sign |
|---|---|---|---|---|---|---|
| | | | | Day to Day | Record | |
| 1. | Identify different types of Network devices/cables and practically implement the cross-wired cable and straight through cable using crimping tool. | | | | | |
| 2. | Demonstrate connectivity of wired & wireless devices in Local Area Network using Hub, Switch and Router. | | | | | |
| 3. | Experiment with the basic network commands like Ping, IPCONFIG, and Tracer in real networks. | | | | | |
| 4. | Analyze Network Traffic Using Wireshark tool/ TCP dump tool | | | | | |
| 5. | Implement Framing Mechanism using any Programming Language. | | | | | |
| 6. | Implement Error Detection technique using any Programming Language. | | | | | |
| 7. | Experiment with configuration of Host IP, Subnet Mask and Default Gateway of a device in LAN and establish Peer to Peer network connection. | | | | | |
| 8. | Demonstrate Static and Dynamic Addressing Mechanisms. | | | | | |
| 9. | Implement Unicast Routing Algorithm using any Programming Language | | | | | |
| 10. | Demonstrate Network Address Translation (NAT) | | | | | |
| 11. | Show the working of Application Layer Protocols - FTP, DNS, Telnet | | | | | |
| 12. | Case study/Project: Analyse, Design and build a network for an organization using Network Simulation tool. | | | | | |

Date:

# Experiment No. 1

## Aim:

- To identify different types of network devices and cables
- Implement the cross-wired cable and straight through cable using crimping tool

## Apparatus:

- Network Devices
- Network Cables
- Ethernet cable – unshielded Cat5 or 6
- RJ-45 crimping tool
- RJ-45 Cat5 or 6 connectors
- RJ-45 boots
- Side cutters
- Ethernet cable tester

## Description:

A network is the interconnection of devices capable of communication. Networking devices connect computers or other electronic devices to share files or resources like printers. These connectivity devices include repeaters, hubs, bridges, switches, routers, gateways, servers, modems, and network interface cards (NICs).

Networking cables are hardware used to connect network devices or computers to share peripherals like printers and scanners. Types of network cables include coaxial, optical fiber, and twisted pair cables, chosen based on the network's physical layer, topology, and size. These connections can range from a few meters (e.g., Ethernet) to nearly unlimited distances (e.g., the Internet).

In this practical experiment, we'll explore network devices and cable configurations. First, we'll identify various network devices, from routers to switches. Then, armed with a crimping tool, we'll create two types of cables: cross-wired and straight-through. The cross-wired cable connects devices of the same type (e.g., PC to PC), while the straight-through cable links different devices (e.g., PC to switch). By the end, you'll have hands-on experience and a deeper understanding of networking essentials.

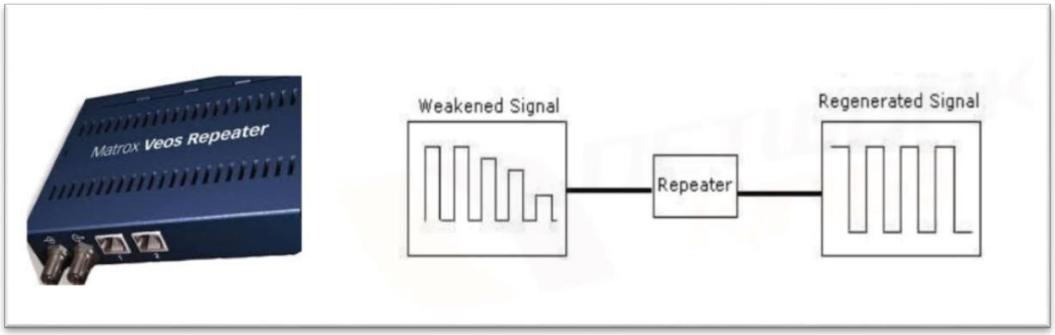
## Procedure:

## Network Devices

**Repeater:**

A repeater operates at the physical layer. Its task is to regenerate the signal over the same network before the signal becomes too weak or corrupted. Unlike an amplifier, a repeater is a regenerator.

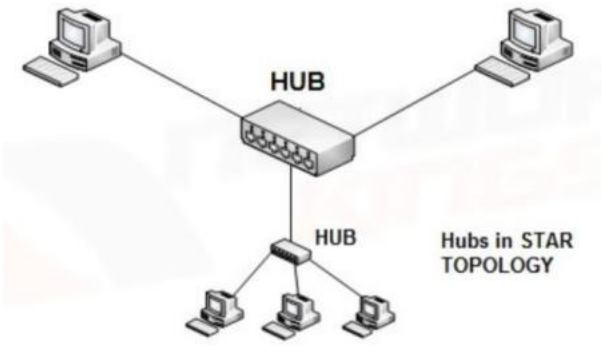
When the signal becomes weak, the repeater copies the signal bit by bit and regenerates it at its original strength. It is a two-port device designed to ensure that data can travel longer distances without degradation.




**Hub:**

A hub is a device used to connect several computers together. Hubs cannot filter data, so data packets are sent to all connected devices. Additionally, they lack the intelligence to find the best path for data packets, leading to inefficiencies and wastage. Hubs do not provide any security.

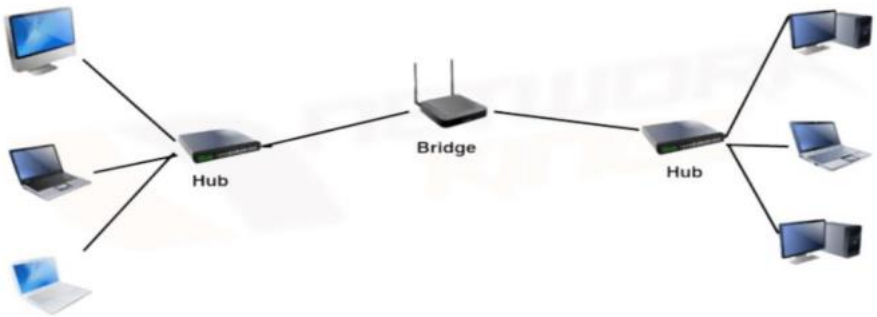
There are two types of hubs: active and passive. Active hubs use electronics to amplify and clean up the signal before it is broadcast to the other ports. Passive hubs simply connect all ports together electrically and are not powered.



**Bridge:**

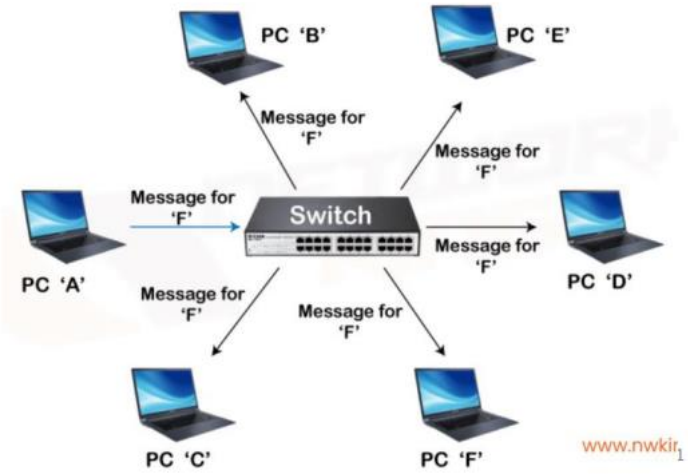
A bridge is a network device that connects multiple LANs (local area networks) to form a larger LAN, making the different components appear as parts of a single network. Operating at the data link layer of the OSI model, bridges are used to divide a LAN into smaller segments or to connect LANs using the same protocol.

Unlike hubs and repeaters, a bridge performs filtering by reading the MAC addresses of the source and destination. It does not send the received frame to all other interfaces. Instead, it uses a table for filtering decisions, enhancing the efficiency of data transmission within the network.




**Switch:**

Switches are networking devices that operate at layer 2, the data link layer of the OSI model. They connect devices in a network and use packet switching to send, receive, or forward data packets or frames across the network. Switches perform error checking before forwarding data, making them very efficient by ensuring that only error-free packets are sent to the correct port.



**Router:**

A router is a device similar to a switch but routes data packets based on their IP addresses. Operating at layer 3, the network layer of the OSI model, routers connect LANs and WANs. They use a dynamically updating routing table to make decisions on routing data packets.

Routers are responsible for receiving, analyzing, and forwarding data packets among connected networks. When a data packet arrives, the router inspects the destination address, consults its routing table to decide the optimal route, and then transfers the packet along that route.

**Gateway:**

A gateway is a device that connects two different networks, often working on different networking models. Acting as a messenger, it takes data from one system, interprets it, and transfers it to another system. Gateways are also known as protocol converters and can operate at any network layer.

**Brouter:**

A brouter, also known as a bridging router, is a device that combines the features of both a bridge and a router. It can operate at the data link layer or the network layer. When functioning as a router, it routes packets across networks. When working as a bridge, it filters local area network traffic, providing the functionalities of both devices in a single unit.

**Modem:**

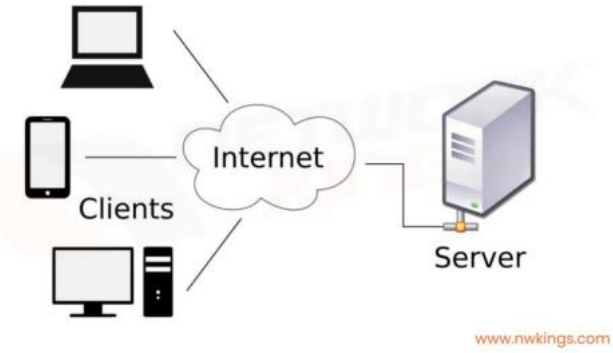
A modem (Modulator-Demodulator) converts digital information into analog signals for transmission over analog mediums like telephone lines or radio waves (modulation). It also converts received analog signals back into digital information for digital devices to process (demodulation).

**Network Interface Card (NIC):**

A network interface card (NIC) is essential hardware for connecting a computer to a network. It is a circuit board installed in a computer that provides a dedicated network connection, acting as a physical interface between the computer and the network cabling. NICs prepare data for transmission, control data flow, and translate received data into bytes for the CPU. They have specific MAC addresses and are also known as network interface controllers, network adapters, or LAN adapters.

**Server:**

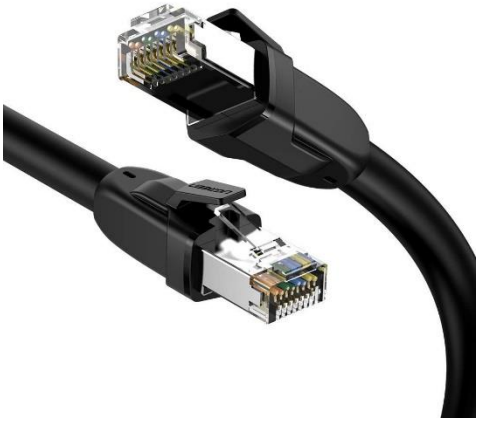
A server is a powerful computer that stores, sends, and receives data to fulfill client requests. For example, when you watch a YouTube video on your computer, mobile, or laptop, the video is hosted on a YouTube server, which serves your request to stream the video. Servers are essential in managing network resources and providing services such as web hosting, email, and file sharing.



# Network Cables

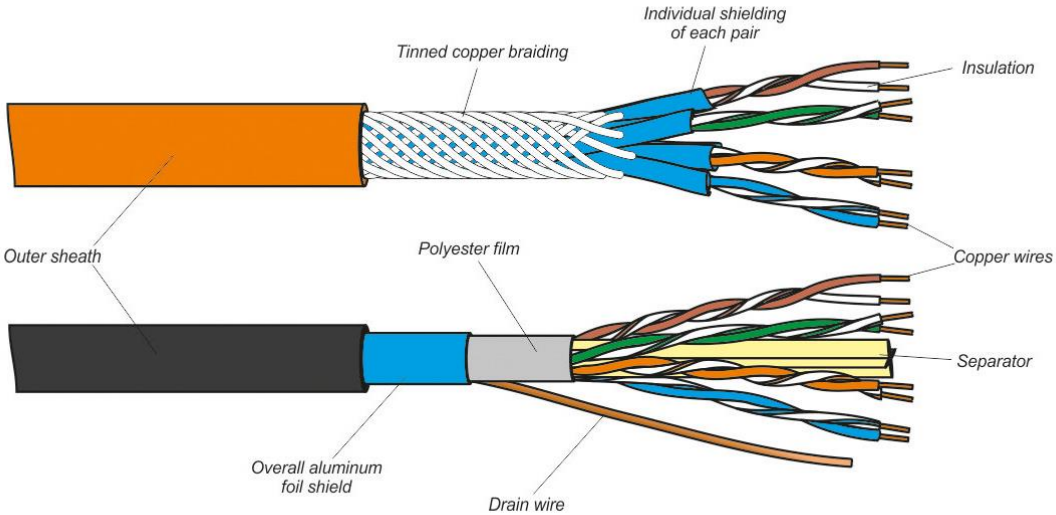
**Ethernet Cable (Cat5/Cat6):**

Ethernet cables, such as Cat5 and Cat6, are used for wired networking. Cat5 supports up to 100 Mbps speeds, while Cat6 supports up to 10 Gbps over shorter distances.



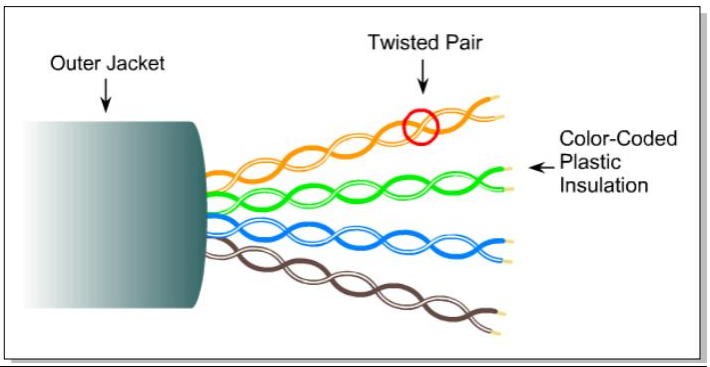
**Twisted Pair Cable:**

A twisted pair cable consists of two insulated wires twisted together to reduce electromagnetic interference. It is commonly used in networking and telecommunications.



**UTP Cable:**

Unshielded Twisted-Pair (UTP) cable is a type of twisted pair cable without additional shielding. It is widely used for Ethernet networks due to its cost-effectiveness and ease of installation.




UTP Cables

| Category | Bandwidth | Digital/Analog | Use |
|---|---|---|---|
| 1 | very low | Analog | Telephone |
| 2 | < 2 MHz | Analog/digital | LANs |
| 3 | 16 MHz | Digital | LANs |
| 4 | 20 MHz | Digital | LANs |
| 5 | 100 MHz | Digital | LANs |
| 5e | 100 MHz | Digital | LANs |
| 6 | 250 MHz | Digital | LANs |
| 6a | 550 MHz | Digital | LANs |
| 7 | 600 MHz | Digital | LANs |

UTP Cable standard

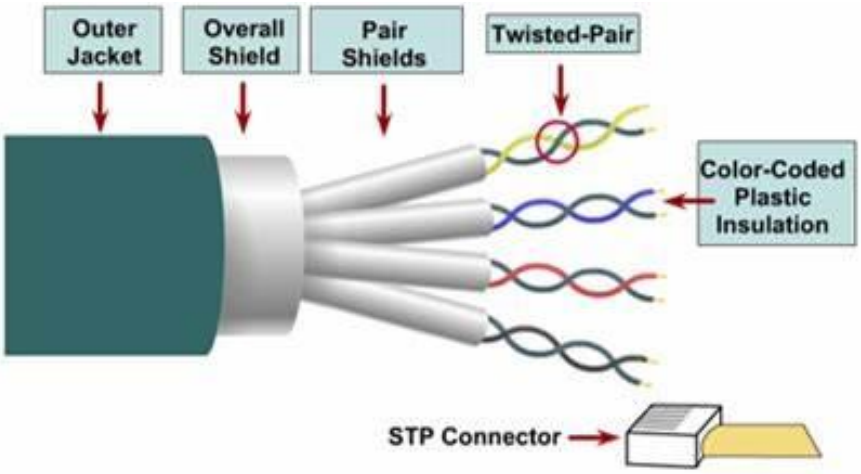
| Types | UTP Cable Type | Duplex | Max Distance |
|---|---|---|---|
| 10Base-T | Cat3/Cat5 | Half/Full | 100 m |
| 100Base-T4 | Cat3 | Half/Full | 100 m |
| 100Base-TX | Cat5 | Full | 100 m |
| 1000Base-T | Cat5e/Cat6 | Full | 100 m |
| 1000Base-TX | Cat6 | Full | 100 m |
| 10GBase-T | Cat6a/Cat7 | Full | 100 m |

UTP Cables and Ethernet

- 10 refers to a maximum transmission speed of 10 Mbps,
- BASE refers to baseband signaling,
- T-twisted-pair

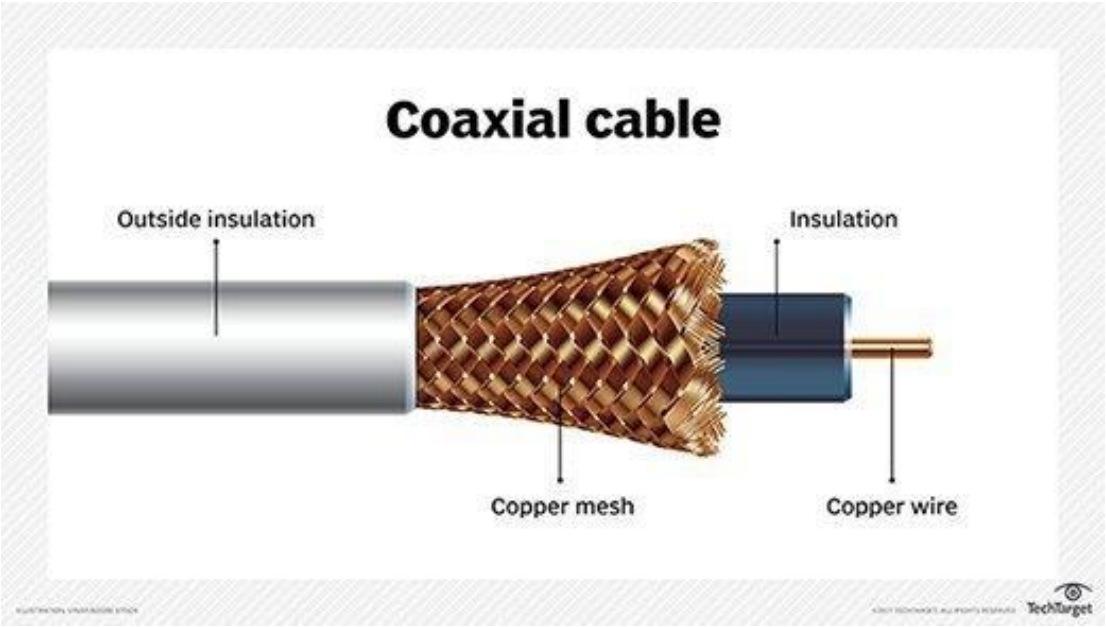
**Shielded Twisted-Pair Cable:**

Shielded Twisted-Pair (STP) cable includes a shielding layer that protects against electromagnetic interference, making it suitable for environments with high interference.



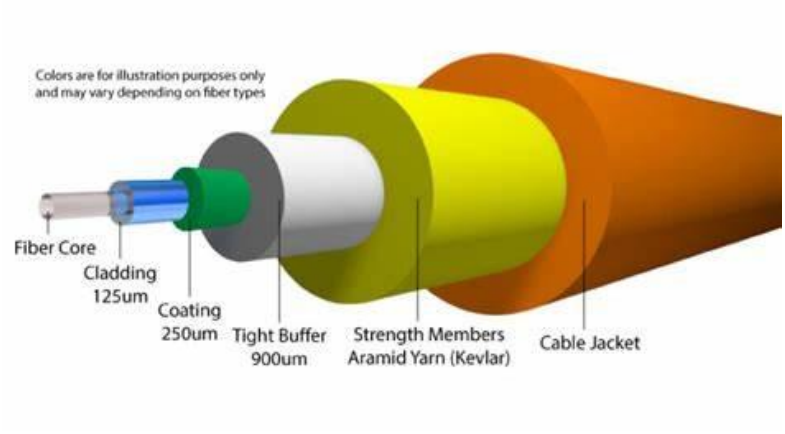

**Coaxial Cable:**

Coaxial cable has a central conductor, an insulating layer, a metallic shield, and an outer insulating layer. It is used for cable television, internet, and other high-frequency signal transmission.



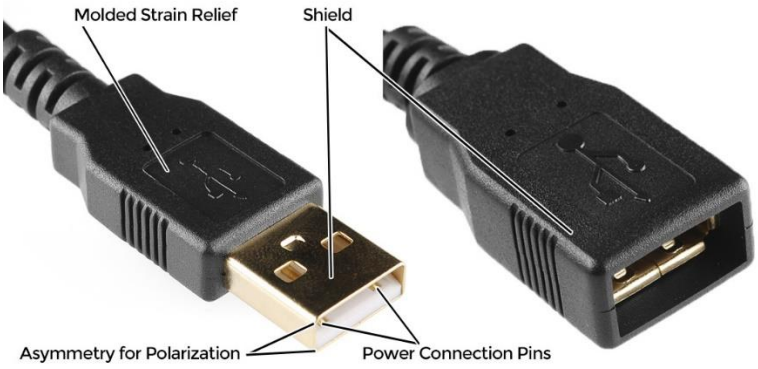
**Fiber Optic Cable:**

Fiber optic cables transmit data as light signals through glass or plastic fibers, offering high bandwidth and long-distance capabilities. They are used in high-speed data communication networks.



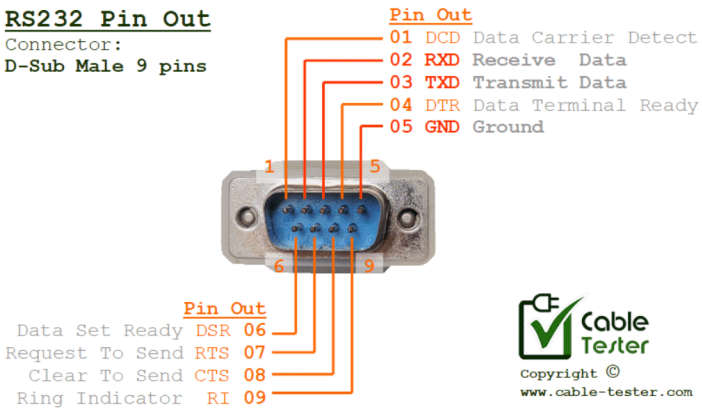

**USB Cable:**

USB cables connect peripherals to computers, allowing data transfer and power supply. Common examples include connecting keyboards, mice, and external storage devices.



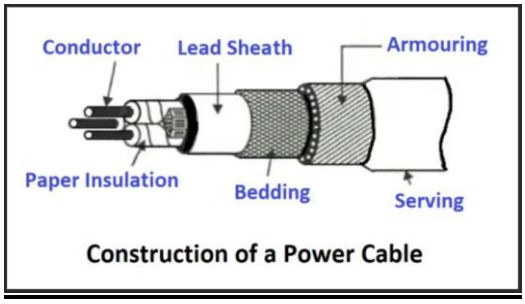

**Serial Cable (RS-232):**

Serial cables, such as RS-232, are used for serial communication between devices. They transmit data one bit at a time and are used in older computer peripherals and networking equipment.



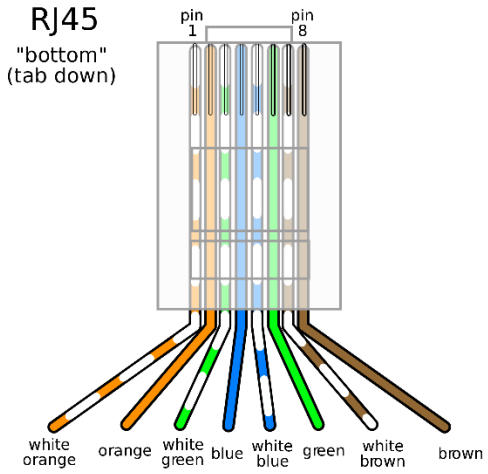
**Power Cable:**

Power cables deliver electrical power to devices. They are essential for powering computers, monitors, and other electronic equipment.

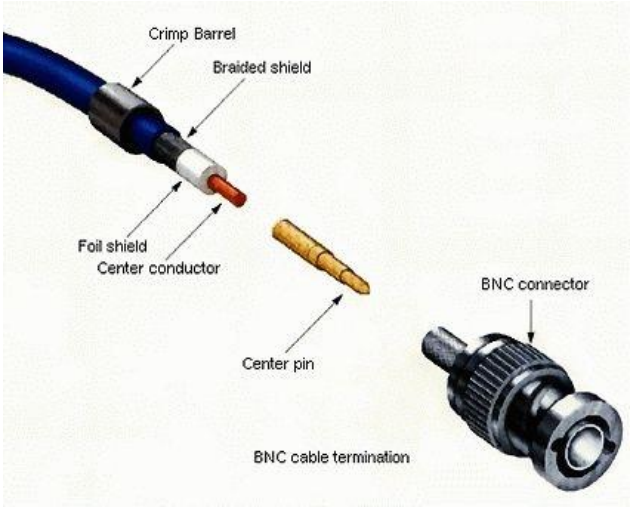



**Connectors:**

- **RJ-45 Connector:** Commonly used for Ethernet cables, RJ-45 connectors connect network devices in a wired LAN.




- **BNC Connector:** Used with coaxial cables, BNC connectors provide secure connections for radio frequency signals in networking and video applications.



**Wireless Media:**

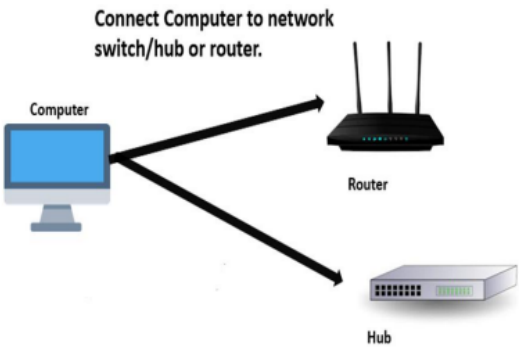
- **Wireless LAN or WLAN:** A wireless local area network that uses radio waves for communication.
- **Wi-Fi ("Wireless Fidelity"):** A set of standards based on IEEE 802.11 for wireless LANs, widely used for internet access.
- **Wi-Max:** Emerging technology capable of covering ranges up to 10 miles or more, used for metropolitan area networks and rural broadband.
- **Satellite/Microwave:** High-speed media for long-distance communication, often used in remote locations and satellite communications.

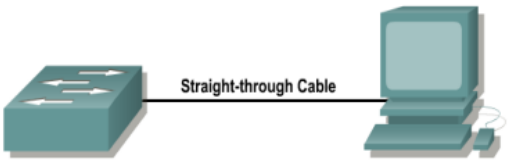
**Straight Through Cable:**

A straight-through cable is a type of CAT5 cable with RJ-45 connectors at each end, following either the T568A or T568B standards. It connects different types of network devices, such as computers to switches or routers to modems. It has the same pin configuration on both ends.

**When to Use Straight Through Cable:**

- Connect a computer to a switch/hub's normal port.
- Connect a computer to a cable/DSL modem's LAN port.
- Connect a router's WAN port to a cable/DSL modem's LAN port.
- Connect two switches or hubs using a normal port on one and an uplink port on the other.



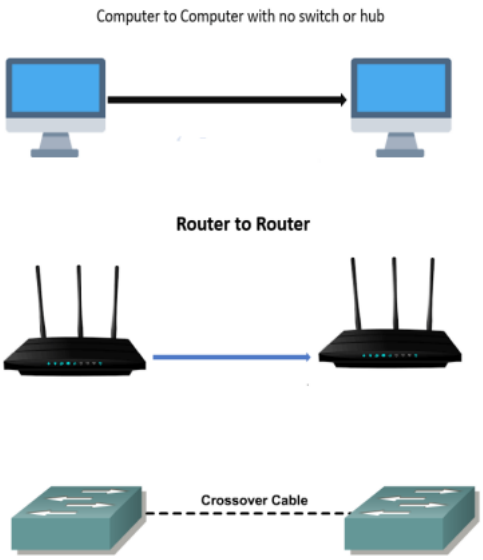


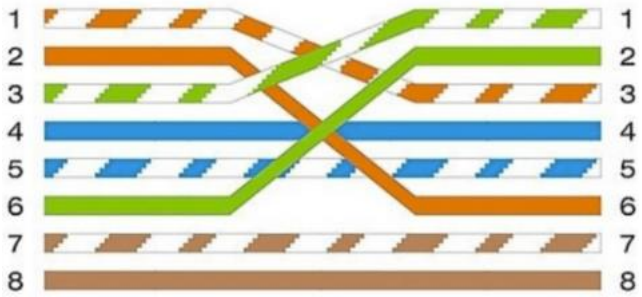
**Crossover Cable:**

A crossover cable is a type of CAT5 cable where one end follows T568A configuration and the other end follows T568B configuration. This cable crosses specific pins (Pin 1 with Pin 3, and Pin 2 with Pin 6) to enable direct device-to-device connections without a switch or hub.

**When to Use Crossover Cable:**

- Connect two computers directly.
- Connect network devices directly, such as router to router.
- Establish a direct connection between two computing devices using Ethernet ports.
- Connect two hubs/switches using a normal port on both devices.




**Choosing Between Straight Through and Crossover Cable:**

- Use a **straight-through cable** to connect different types of devices (e.g., PC to switch).
- Use a **crossover cable** to connect similar devices directly (e.g., PC to PC, router to router).

## Straight Through vs. Crossover Cable, which one to choose?

| Network Devices | HUB | SWITCH | ROUTER | PC |
|---|---|---|---|---|
| Hub | Crossover | Crossover | Straight | Straight |
| Switch | Crossover | Crossover | Straight | Straight |
| Router | Straight | Straight | Crossover | Crossover |
| PC | Straight | Straight | Crossover | Crossover |

**Ethernet Media and Connector:**

- Standards: EIA/TIA-568 (SP-2840), governing commercial building telecommunications wiring standards.
- **RJ-45 Color Code:**
  - **EIA/TIA 568A:** 1. White/Green, 2. Green, 3. White/Orange, 4. Blue, 5. White/Blue, 6. Orange, 7. White/Brown, 8. Brown.
  - **EIA/TIA 568B:** 1. White/Orange, 2. Orange, 3. White/Green, 4. Blue, 5. White/Blue, 6. Green, 7. White/Brown, 8. Brown.

**Creating a fully functional Ethernet cable using the T568B wiring standard :**

1. Prepare the Cable**:**
   - Cut the cable to the desired length using the wire cutter.
   - Strip about 1-2 inches of the outer jacket off the cable using the wire stripper. Be careful not to nick the inner wires.
2. Untwist and Arrange the Wires**:**
   - Untwist the pairs of wires and arrange them according to the T568B wiring standard:
     - T568B Color Order**:**
       - Pin 1: White/Orange
       - Pin 2: Orange
       - Pin 3: White/Green
       - Pin 4: Blue
       - Pin 5: White/Blue
       - Pin 6: Green
       - Pin 7: White/Brown
       - Pin 8: Brown
   - Hold the wires flat between your thumb and forefinger in the correct order.
3. Trim the Wires**:**
   - Trim the wires to make sure they are all even and about 1/2 inch in length from the cut end of the outer jacket.
4. Insert Wires into the RJ-45 Connector**:**
   - Carefully insert the arranged wires into the RJ-45 connector with the clip facing down. Make sure each wire goes into its correct slot and reaches the end of the connector.
5. Crimp the Connector:
   - Insert the RJ-45 connector into the crimping tool.
   - Squeeze the crimping tool firmly to secure the wires in the connector and to cut off any excess wire.
6. Repeat for the Other End:
   - Repeat steps 1-5 for the other end of the cable.
7. Test the Cable:
   - Use an Ethernet cable tester to ensure the cable is wired correctly and functioning.
   - Plug each end of the cable into the tester, and check that all pins are connected properly.

## Results and Observations:

We identify network devices and create cross-wired and straight-through cables, testing connectivity for accurate communication.

## Viva-Voce Questions:

**Q)** Explain the roles and functionalities of Repeater, Hub, Bridge, Switch, Router, Gateway, Modem, Network Interface Card (NIC), Firewall, and access points in a typical network setup. Discuss how these devices work together to manage data traffic, ensure connectivity, and maintain network security. Provide examples of common configurations and scenarios where each device is essential.

**A.** Network Devices: Roles and Functionalities

- Repeater: Amplifies signals to extend network reach.
- Hub: Connects devices but lacks intelligence; sends data to all connected devices.
- Bridge: Filters data based on MAC addresses, interconnects LANs.
- Switch: Efficiently forwards data to specific devices based on MAC addresses.
- Router: Connects LANs, directs traffic between networks (e.g., LAN to internet).
- Gateway: Translates protocols between different networks (e.g., LAN to Wi-Fi).
- NIC: Hardware card enabling device-to-network communication.
- Firewall: Protects against unauthorized access and threats.
- Access Point: Provides wireless connectivity.

**Q)** Explain guided transmission media and unguided transmission media used in computer networks in detail.

**A.** Guided vs. Unguided Transmission Media

- Guided (Wired): Includes coaxial cables, twisted-pair cables, and fiber optics. Signals travel along a physical path.
- Unguided (Wireless): Includes radio waves, microwaves, and infrared. Signals propagate freely through the air.

**Q)** What are the primary differences between Cat5e and Cat6 Ethernet cables in terms of data transfer speed, bandwidth, and maximum transmission distance? Additionally, describe the standard color code for the wiring of these cables according to the T568A and T568B wiring standards. Discuss their respective advantages and typical use cases.

**A.** Cat5e vs. Cat6 Ethernet Cables

- Cat5e: Up to 1 Gbps speed, 100 MHz bandwidth, max distance of 100 meters. Commonly used for home networks.
- Cat6: Up to 10 Gbps speed, 250 MHz bandwidth, max distance of 55 meters (10 Gbps) or 100 meters (1 Gbps). Ideal for high-performance networks.
- Color Code (T568B): Orange-white, orange, green-white, blue, blue-white, green, brown-white, brown.
- Advantages: Cat6 offers better performance, reduced crosstalk, and future-proofing.

**Q)** Create a fully functional Ethernet cable using the T568B wiring standard.

**A.** Creating an Ethernet Cable (T568B)

- Gather materials: Cat6 cable, RJ45 connectors, crimping tool.
- Strip cable jacket, arrange wires in T568B order.
- Insert wires into connector, crimp securely.
- Test cable for connectivity.

**Q)** What is the primary purpose of a firewall in a network, and how does it enhance security?

**A.** A firewall serves as a critical security barrier within a network. Its primary purpose is to control and monitor incoming and outgoing traffic, safeguarding against unauthorized access and potential threats. Here's how it enhances security:

**1.** Access Control:
- Firewalls filter network packets based on predefined rules.
- They allow or block traffic based on factors like source IP, destination port, and protocol.
- By enforcing access policies, firewalls prevent unauthorized entry into the netwk.

**2.** Traffic Inspection:
- Firewalls analyze data packets to detect anomalies or malicious patterns.
- They inspect packet headers and payload content.
- Suspicious or harmful traffic is flagged and either blocked or logged for further investigation.

**3.** Network Segmentation:
- Firewalls divide the network into security zones (e.g., internal, DMZ, external).
- They isolate critical systems from less secure areas.
- This segmentation limits the impact of breaches and contains threats.

**4.** Stateful Packet Inspection (SPI):
- Firewalls maintain state tables to track active connections.
- They allow only legitimate responses to outbound requests.
- SPI prevents unauthorized data flow and ensures session integrity.

**5.** Application Layer Filtering:
- Modern firewalls inspect traffic at the application layer (Layer 7).
- They understand specific protocols (e.g., HTTP, FTP, DNS) and enforce granular rules.
- This helps block application-specific threats.

Date:

# Experiment No. 2

## Aim:

Demonstrate connectivity of wired & wireless devices in Local Area Network using Hub, Switch and Router in packet tracer.

## Apparatus:

- Modern
- Hub
- Switch
- Router
- Cloud
- Home PC
- Family PC
- Home Printer
- Server
- CISCO Packet Tracer

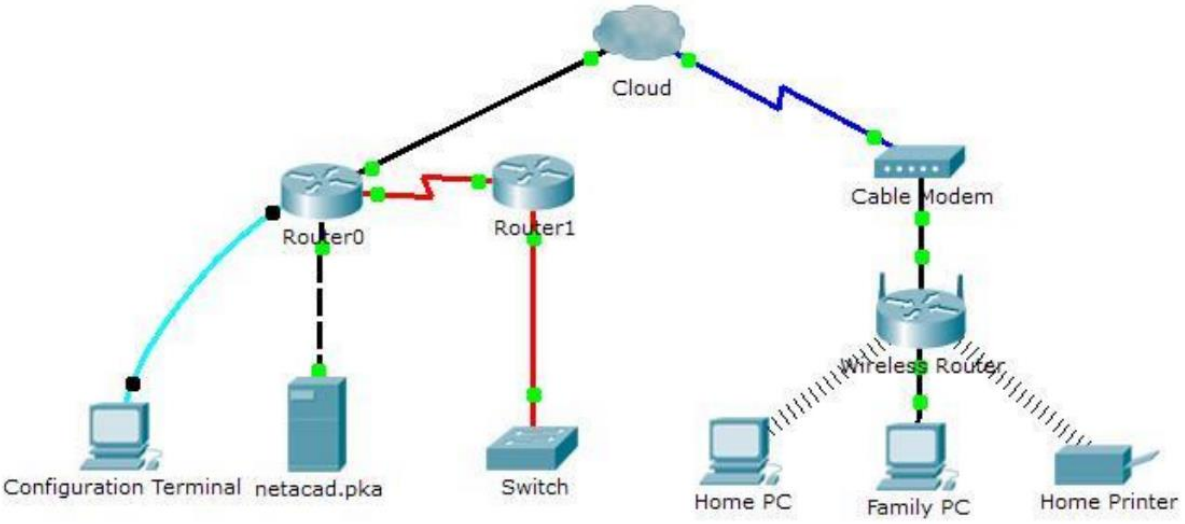
## Description:

In a network setup, we demonstrate connectivity using wired and wireless devices. Here's how it works:

1. Wired Connectivity:
    - We connect devices (such as PCs) using hubs or switches.
    - Each device gets a unique IP address.
    - Testing involves pinging between devices to ensure communication.
2. Wireless Connectivity:
    - We add a router, switch, access point, and wireless devices (like laptops).
    - LAN-connected devices receive IP addresses.
    - Wireless devices connect to the access point (using SSID and password).
    - We verify connectivity using ping commands.

Remember to assign IPs and test successful communication!

**Description: Network Diagram**




## Addressing Table

| Device | Interface | IP Address | Connects To |
|---|---|---|---|
| Cloud | Eth6 | N/A | F0/0 |
| | Coax7 | N/A | Port0 |
| Cable Modem | Port0 | N/A | Coax7 |
| | Port1 | N/A | Internet |
| Router0 | Console | N/A | RS232 |
| | F0/0 | 192.168.2.1/24 | Eth6 |
| | F0/1 | 10.0.0.1/24 | F0 |
| | Ser0/0/0 | 172.31.0.1/24 | Ser0/0 |
| Router1 | Ser0/0 | 172.31.0.2/24 | Ser0/0/0 |
| | F1/0 | 172.16.0.1/24 | F0/1 |
| WirelessRouter | Internet | 192.168.2.2/24 | Port 1 |
| | Eth1 | 192.168.1.1 | F0 |
| Family PC | F0 | 192.168.1.102 | Eth1 |
| Switch | F0/1 | 172.16.0.2 | F1/0 |
| Netacad.pka | F0 | 10.0.0.254 | F0/1 |
| Configuration Terminal | RS232 | N/A | Console |

## Procedure:

**Part 1: Packet Tracer installation**

https://www.netacad.com/courses/packet-tracer
Sign up
Login
Download Packet tracer
Lear all options

**Part 2: Connect to the Cloud**

Step 1: Connect the cloud to Router0.

a. At the bottom left, click the orange lightning icon to open the available Connections.
b. Choose the correct cable to connect Router0 Fa0/0 to Cloud Eth6. Cloud is a type of switch, so use a Copper Straight-Through connection.

If you attached the correct cable, the link lights on the cable turn green.

Step 2: Connect the cloud to Cable Modem.

Choose the correct cable to connect Cloud Coax7 to Modem Port0.
If you attached the correct cable, the link lights on the cable turn green.

**Part 3: Connect Router0**

Step 1: Connect Router0 to Router1.

Choose the correct cable to connect Router0 Ser0/0/0 to Router1 Ser0/0. Use one of the available Serial cables.
If you attached the correct cable, the link lights on the cable turn green.

Step 2: Connect Router0 to netacad.pka.

Choose the correct cable to connect Router0 Fa0/1 to netacad.pka Fa0. Routers and computers traditionally use the same wires to transmit (1 and 2) and receive (3 and 6). The correct cable to choose consists of these crossed wires. Although many NICs can now autosense which pair is used to transmit and receive, Router0 and netacad.pka do not have autosensing NICs.
If you attached the correct cable, the link lights on the cable turn green.

Step 3: Connect Router0 to the Configuration Terminal.

Choose the correct cable to connect Router0 Console to Configuration Terminal RS232. This cable does not provide network access to Configuration Terminal, but allows you to configure Router0 through its terminal.
If you attached the correct cable, the link lights on the cable turn black

**Part 4: Connect Remaining Devices**

Step 1: Connect Router1 to Switch.

Choose the correct cable to connect Router1 Fa1/0 to Switch Fa0/1.
If you attached the correct cable, the link lights on the cable turn green.
Allow a few seconds for the light to transition from amber to green.

Step 2: Connect Cable Modem to Wireless Router.

Choose the correct cable to connect Modem Port1 to Wireless Router Internet port.
If you attached the correct cable, the link lights on the cable will turn green.

Step 3: Connect Wireless Router to Family PC.

Choose the correct cable to connect Wireless Router Ethernet 1 to Family PC.
If you attached the correct cable, the link lights on the cable turn green

**Part 5: Verify Connections**

Step 1: Test the connection from Family PC to netacad.pka.

a. Open the Family PC command prompt and ping netacad.pka.
b. Open the Web Browser and the web address http://netacad.pka.

Step 2: Ping the Switch from Home PC.

Open the Home PC command prompt and ping the Switch IP address of to verify the connection.

Step 3: Open Router0 from Configuration Terminal.

a. Open the Terminal of Configuration Terminal and accept the default settings.
b. Press Enter to view the Router0 command prompt.
c. Type show ip interface brief to view interface statuses

**Part 6: Examine the Physical Topology**

Step 1: Examine the Cloud.

a. Click the Physical Workspace tab or press Shift+P and Shift+L to toggle between the logical and physical workspaces.
b. Click the Home City icon.
c. Click the Cloud icon. How many wires are connected to the switch in the blue rack? 2
d. Click Back to return to Home City.

Step 2: Examine the Primary Network.

a. Click the Primary Network icon. Hold the mouse pointer over the various cables. What is located on the table to the right of the blue rack?
Configuration Terminal
b. Click Back to return to Home City.

Step 3: Examine the Secondary Network.

a. Click the Secondary Network icon. Hold the mouse pointer over the various cables. Why are there two orange cables connected to each device?
Fiber cables come in pairs, one for transmit, the other for receive
b. Click Back to return to Home City.

Step 4: Examine the Home Network.

a. Why is there an oval mesh covering the home network?
It represents the range of the wireless network
b. Click the Home Network icon. Why is there no rack to hold the equipment?
Home networks typically do not have racks.
c. Click the Logical Workspace tab to return to the logical topology.

## Observations:

When connecting devices in a network setup, the yellow light typically indicates initialization—a negotiation phase where devices establish settings. Once successful, the light turns green, signifying a stable connection.

Yellow Light (Initialization):

- When devices connect, they negotiate settings (speed, duplex mode, etc.).
- During this handshake, the light often appears yellow or amber.
- It signifies the negotiation process—a cautious introduction.

Green Light (Stable Connection):

- Once negotiation completes successfully, the light turns green.
- Green means go! Devices are now confidently linked.
- Data flows smoothly, and the network is ready for action.

## Results

In the network connectivity experiment using wired and wireless devices, successful communication occurs via hubs or switches (wired) and wireless access points. Each device obtains an IP address, and verification through ping commands confirms seamless connectivity.

## Viva-Voce Questions:

**Q)** What causes slow network speeds on a wired connection?
**A.** Slow network speeds on a wired connection are often due to network overload. Solutions include adding switches or creating VLANs with switches. Similarly, wireless connections may be slow due to overload on the wireless access point (WAP) or wireless router.

**Q)** What is the most challenging issue when troubleshooting intermittent wireless connectivity?
**A.** Intermittent connectivity—when wireless connections appear and disappear—is difficult to troubleshoot. Addressing latency, jitter, port flapping, and slow network speeds helps narrow down the issue

**Q)** What does “limited connectivity” mean in networking?
**A.** Limited connectivity occurs when a device is connected to the network but can’t reach resources outside the local network. Incorrect Wi-Fi passwords or IP address issues are common causes.

**Q)** Which devices connect IoT devices to a home network?
**A.** IoT devices can connect via a home gateway or registration server, whether wired or wirelessly.

**Q)** What distinguishes wired networks from wireless networks?
**A.** Wired networks use physical cables (e.g., Ethernet) to connect devices, while wireless networks rely on electromagnetic waves (e.g., Wi-Fi) without physical wires.

Date:

# Experiment No. 3

## Aim:

Experiment with the basic network commands link ping, ipconfig, and tracert in real networks

## Apparatus:

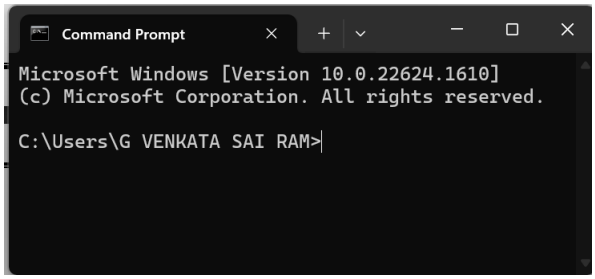
A working computer with terminal or command prompt in it.

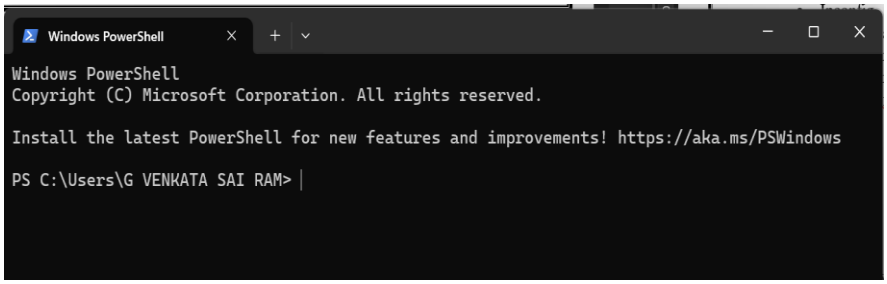
## Description:

In this experiment we will practice basic network commands and network configuration commands. They are:

- Hostname
- Ipconfig
- Ipconfig /all
- Ipconfig /release
- Ipconfig /renew
- Ipconfig /flushdns
- Nbtstat -a
- Netstat
- Nslookup
- Pathping
- Ping
- Route
- Tracet

Now let us observe how a command prompt/Terminal of windows looks like:



What is the main difference between command prompt and a Terminal(Power Shell)?

Windows has both Command Prompt and PowerShell (its more powerful counterpart) to cater to different user needs. Command Prompt is simpler and retains compatibility with older scripts and commands. PowerShell, on the other hand, offers advanced scripting capabilities and access to a broader range of system management tools, making it more suitable for modern administrative tasks and automation. Having both allows users to choose the tool that best fits their requirements.

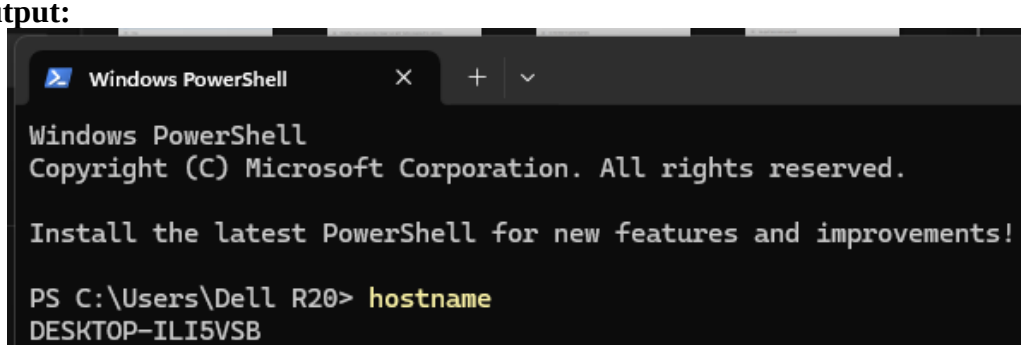
## Procedure:

- Turn on the PC you are working on
- When the PC is successfully turned on
- Finish the password formalities and then you can see Desktop
- When you see the desktop, there are many ways to open command prompt/terminal
    - Method – 1
        - Press + r
        - Enter "cmd" in the input and press enter
        - Now you can see the command prompt
    - Method – 2
        - Press + s
        - Enter "command prompt" /"Terminal" in the input and press enter
        - Now you can see the command prompt / Terminal
- Now let's dive deeper into each command

**hostname:**

- This is the simplest of all TCP/IP commands.
- It simply displays the name of your computer.
- C:\>hostname

**Output:**



**ipconfig:**

- The ipconfig command displays information about the host (the computer your sitting at) computer TCP/IP configuration.
- C:\>ipconfig

**Output:**

```
PS C:\Users\Dell R20> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::6b1:9e7:a622:22b7%6
   IPv4 Address. . . . . . . . . . . : 172.50.21.57
   Subnet Mask . . . . . . . . . . . : 255.255.224.0
   Default Gateway . . . . . . . . . : 172.50.5.203

Ethernet adapter vEthernet (Default Switch):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8b00:7021:48d7:7efa%17
   IPv4 Address. . . . . . . . . . . : 172.22.80.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
```

**ipconfig /all:**

- This command displays detailed configuration information about your TCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernet adapter in your system.
- C:\>ipconfig /all

**Output:**

```
PS C:\Users\Dell R20> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DESKTOP-ILI5VSB
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Ethernet Connection (11) I219-LM
   Physical Address. . . . . . . . . : 70-B5-E8-5E-78-35
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::6b1:9e7:a622:22b7%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.50.21.57(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.224.0
   Default Gateway . . . . . . . . . : 172.50.5.203
   DHCPv6 IAID . . . . . . . . . . . : 342930920
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-28-22-ED-1A-70-B5-E8-5E-78-35
   DNS Servers . . . . . . . . . . . : 172.50.5.203
                                       8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter vEthernet (Default Switch):

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Hyper-V Virtual Ethernet Adapter
   Physical Address. . . . . . . . . : 00-15-5D-1C-02-10
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::8b00:7021:48d7:7efa%17(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.22.80.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 285218141
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-28-22-ED-1A-70-B5-E8-5E-78-35
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

**ipconfig /release:**

- This command allows you to drop the IP lease from the DHCP server.
- C:\>ipconfig /release

**ipconfig /renew:**

- Using this command will renew all your IP addresses that you are currently (leasing) borrowing from the DHCP server.
- This command is a quick problem solver if you are having connection issues, but does not work if you have been configured with a static IP address.
- C:\>ipconfig /renew

**ipconfig /flushdns:**

- Flush DNS
- This command is only needed if you're having trouble with your networks DNS configuration.
- The best time to use this command is after network configuration frustration sets in, and you really need the computer to reply with flushed.
- C:\>ipconfig /flushdns

**ping (**Packet Internet Grouper):

- The Ping (Packet Internet Grouper) command was first developed in 1983 and is one of the simplest and most effective tools you can use to resolve issues.
- The ping command is one of the most often used networking utilities for troubleshooting network problems.
- Use the ping command to test the availability of a networking device (usually a computer) on a network.
- C:\>ping google.com

**Output:**

```
PS C:\Users\Dell R20> ping google.com

Pinging google.com [142.250.193.110] with 32 bytes of data:
Reply from 142.250.193.110: bytes=32 time=27ms TTL=116
Reply from 142.250.193.110: bytes=32 time=28ms TTL=116
Reply from 142.250.193.110: bytes=32 time=27ms TTL=116
Reply from 142.250.193.110: bytes=32 time=28ms TTL=116

Ping statistics for 142.250.193.110:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 27ms, Maximum = 28ms, Average = 27ms
```

- C:\>ping 192.168.0.1

**Output:**

```
PS C:\Users\Dell R20> ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

- C:\>ping -t 172.50.5.203

```
PS C:\Users\Dell R20> ping -t 172.50.5.203

Pinging 172.50.5.203 with 32 bytes of data:
Reply from 172.50.5.203: bytes=32 time<1ms TTL=64
Reply from 172.50.5.203: bytes=32 time<1ms TTL=64
Reply from 172.50.5.203: bytes=32 time<1ms TTL=64
Reply from 172.50.5.203: bytes=32 time<1ms TTL=64

Ping statistics for 172.50.5.203:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
```

**tracert:**

- This is another tool that has been around since the early days of Unix machines.
- It is useful for tracing a path to a website or server across the internet, to look for any issues along the way.
- The tracert command displays a list of all the routers that a packet has to go through to get from the computer where tracert is run to any other computer on the internet.
- C:\>tracert google.com

**Output:**

```
PS C:\Users\Dell R20> tracert google.com

Tracing route to google.com [142.250.193.110]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  170.50.5.203
  2     1 ms     1 ms     1 ms  202.53.81.81
  3     *        5 ms     *     www.nettlinx.com [202.53.64.5]
  4     6 ms     5 ms     6 ms  mail.nettlinx.org [14.98.115.229]
  5     6 ms     6 ms     *     121.244.3.193.static-mumbai.vsnl.net.in [121.244.3.193]
  6    24 ms    24 ms    24 ms  172.23.78.238
  7    24 ms    24 ms    24 ms  115.113.165.98.static-mumbai.vsnl.net.in [115.113.165.98]
  8    25 ms    25 ms    25 ms  142.251.225.9
  9    24 ms    24 ms    24 ms  142.250.226.66
 10    31 ms    30 ms    30 ms  142.250.212.1
 11    28 ms    29 ms    28 ms  172.253.72.136
 12    33 ms    32 ms    32 ms  142.251.230.53
 13    28 ms    28 ms    28 ms  142.251.55.223
 14    28 ms    28 ms    28 ms  maa05s24-in-f14.1e100.net [142.250.193.110]

Trace complete.
```

**pathping:**

- Pathping is unique to Window's, and is basically a combination of the Ping and Tracert commands.
- Pathping traces the route to the destination address then launches a 25 second test of each router along the way, gathering statistics on the rate of data loss along each hop.
- It's a good tool for finding out additional information regarding a connection across the internet and can be used to identify issues that tracert and ping used individually may not spot.
- C:\>pathping google.com

**Output:**

```
PS C:\Users\Dell R20> pathping google.com

Tracing route to google.com [142.250.193.110]
over a maximum of 30 hops:
  0  DESKTOP-ILI5VSB [172.50.21.57]
  1     *     170.50.5.203
  2  202.53.81.81
  3     *     www.nettlinx.com [202.53.64.5]
  4  mail.nettlinx.org [14.98.115.229]
  5     *     121.244.3.193.static-mumbai.vsnl.net.in [121.244.3.193]
  6  172.23.78.238
  7  115.113.165.98.static-mumbai.vsnl.net.in [115.113.165.98]
  8  142.251.225.9
  9  142.250.226.66
 10  142.250.212.1
 11  172.253.72.136
 12  142.251.230.53
 13  142.251.55.223
 14  maa05s24-in-f14.1e100.net [142.250.193.110]

Computing statistics for 350 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           DESKTOP-ILI5VSB [172.50.21.57]
                                 0/ 100 =  0%   |
  1    0ms     0/ 100 =  0%     0/ 100 =  0%  170.50.5.203
                                 0/ 100 =  0%   |
  2    0ms     0/ 100 =  0%     0/ 100 =  0%  202.53.81.81
                                 0/ 100 =  0%   |
  3    5ms     0/ 100 =  0%     0/ 100 =  0%  www.nettlinx.com [202.53.64.5]
                                 0/ 100 =  0%   |
  4    6ms     0/ 100 =  0%     0/ 100 =  0%  mail.nettlinx.org [14.98.115.229]
                                 0/ 100 =  0%   |
  5    6ms     0/ 100 =  0%     0/ 100 =  0%  121.244.3.193.static-mumbai.vsnl.net.in [121.244.3.193]
                                 0/ 100 =  0%   |
  6  ---     100/ 100 =100%   100/ 100 =100%  172.23.78.238
                                 0/ 100 =  0%   |
  7   24ms     0/ 100 =  0%     0/ 100 =  0%  115.113.165.98.static-mumbai.vsnl.net.in [115.113.165.98]
                                 0/ 100 =  0%   |
  8   25ms     0/ 100 =  0%     0/ 100 =  0%  142.251.225.9
                                 0/ 100 =  0%   |
  9   25ms     0/ 100 =  0%     0/ 100 =  0%  142.250.226.66
                                 0/ 100 =  0%   |
 10   31ms     0/ 100 =  0%     0/ 100 =  0%  142.250.212.1
                                 0/ 100 =  0%   |
 11   28ms     0/ 100 =  0%     0/ 100 =  0%  172.253.72.136
                                 0/ 100 =  0%   |
 12   32ms     0/ 100 =  0%     0/ 100 =  0%  142.251.230.53
                                 0/ 100 =  0%   |
 13   35ms     0/ 100 =  0%     0/ 100 =  0%  142.251.55.223
                                 0/ 100 =  0%   |
 14   28ms     0/ 100 =  0%     0/ 100 =  0%  maa05s24-in-f14.1e100.net [142.250.193.110]

Trace complete.
```

**netstat (net**work **stat**istics):

- Netstat displays a variety of statistics about a computers active TCP/IP connections. This tool is most useful when you're having trouble with TCP/IP applications such as HTTP, and FTP.
- C:\>netstat –a

**Output:**

```
PS C:\Users\Dell R20> netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49687        DESKTOP-ILI5VSB:49688  ESTABLISHED
  TCP    127.0.0.1:49688        DESKTOP-ILI5VSB:49687  ESTABLISHED
  TCP    127.0.0.1:49689        DESKTOP-ILI5VSB:49690  ESTABLISHED
  TCP    127.0.0.1:49690        DESKTOP-ILI5VSB:49689  ESTABLISHED
  TCP    172.50.21.57:7680      172.50.5.95:51637      ESTABLISHED
  TCP    172.50.21.57:7680      CSERD4:54819           ESTABLISHED
  TCP    172.50.21.57:7680      172.50.12.122:63438    ESTABLISHED
  TCP    172.50.21.57:7680      172.50.12.152:50236    ESTABLISHED
  TCP    172.50.21.57:7680      DESKTOP-2TRMTCT:50346  ESTABLISHED
  TCP    172.50.21.57:7680      DESKTOP-8DIU4NI:51734  ESTABLISHED
  TCP    172.50.21.57:7680      DESKTOP-ANQ5U17:50671  ESTABLISHED
  TCP    172.50.21.57:7680      172.50.21.31:60834     ESTABLISHED
  TCP    172.50.21.57:49558     172.50.13.44:ms-do     ESTABLISHED
  TCP    172.50.21.57:49597     DESKTOP-8ESFV8C:ms-do  ESTABLISHED
  TCP    172.50.21.57:49618     172.50.12.144:ms-do    ESTABLISHED
  TCP    172.50.21.57:49901     20.198.119.84:https    ESTABLISHED
  TCP    172.50.21.57:50269     maa03s41-in-f14:https  ESTABLISHED
  TCP    172.50.21.57:50474     a23-65-124-99:https    CLOSE_WAIT
  TCP    172.50.21.57:50476     13.107.237.254:https   CLOSE_WAIT
  TCP    172.50.21.57:50478     13.107.246.72:https    CLOSE_WAIT
  TCP    172.50.21.57:50479     51.13.112.137:https    CLOSE_WAIT
  TCP    172.50.21.57:50480     204.79.197.222:https   CLOSE_WAIT
  TCP    172.50.21.57:50656     maa03s41-in-f5:https   ESTABLISHED
  TCP    172.50.21.57:50660     maa03s47-in-f1:https   TIME_WAIT
  TCP    172.50.21.57:50680     maa03s41-in-f5:https   ESTABLISHED
  TCP    172.50.21.57:50694     a23-65-124-104:https   TIME_WAIT
  TCP    172.50.21.57:50697     13.89.178.27:https     TIME_WAIT
  TCP    172.50.21.57:50700     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50701     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50704     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50705     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50706     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50707     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50712     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50713     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50714     52.113.194.132:https   ESTABLISHED
  TCP    172.50.21.57:50715     20.24.125.47:https     ESTABLISHED
  TCP    172.50.21.57:50716     20.24.125.47:https     ESTABLISHED
  TCP    172.50.21.57:50718     170.50.17.32:ms-do     SYN_SENT
  TCP    172.50.21.57:50719     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50720     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50721     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50722     dns:domain             TIME_WAIT
  TCP    172.50.21.57:50723     10.1.7.222:ms-do       SYN_SENT
  TCP    172.50.21.57:50724     170.50.4.72:ms-do      SYN_SENT
  TCP    172.50.21.57:65294     13.86.221.35:8883      ESTABLISHED
```

**nbstat:**

- To troubleshoot NetBIOS over TCP/IP connections.
- C:\>nbtstat –a

**Output:**

```
PS C:\Users\Dell R20> nbtstat

Displays protocol statistics and current TCP/IP connections using NBT
(NetBIOS over TCP/IP).

NBTSTAT [ [-a RemoteName] [-A IP address] [-c] [-n]
        [-r] [-R] [-RR] [-s] [-S] [interval] ]

  -a   (adapter status) Lists the remote machine's name table given its name
  -A   (Adapter status) Lists the remote machine's name table given its
                        IP address.
  -c   (cache)          Lists NBT's cache of remote [machine] names and their IP addresses
  -n   (names)          Lists local NetBIOS names.
  -r   (resolved)       Lists names resolved by broadcast and via WINS
  -R   (Reload)         Purges and reloads the remote cache name table
  -S   (Sessions)       Lists sessions table with the destination IP addresses
  -s   (sessions)       Lists sessions table converting destination IP
                        addresses to computer NETBIOS names.
  -RR  (ReleaseRefresh) Sends Name Release packets to WINS and then, starts Refresh

  RemoteName   Remote host machine name.
  IP address   Dotted decimal representation of the IP address.
  interval     Redisplays selected statistics, pausing interval seconds
               between each display. Press Ctrl+C to stop redisplaying
               statistics.
```

**nslookup:**

- Nslookup is used for diagnosing DNS problems. If you can access a resource by specifying an IP address but not it's DNS you have a DNS problem.
- The NSlookup command allows you to question your domains nameservers, and find out much more information regarding your domains DNS.

**Example:**

- In this example we will have a look at the email records for bbc.co.uk and check to make see if any txt records exist for the domain name.
- From a whois search we already know that the domain uses ns1.bbc.co.uk as its nameservers, and by pinging the nameserver we know that the IP address is 132.185.132.21.
    - Open the nslookup tool by entering nslookup
        - C:\> nslookup
    - Next, we need to change the server so that we are asking ns1.bbc.co.uk (132.185.132.21) for the information we require.
        - > server 132.185.132.21
    - Now we want to look for any MX (email) records that exist for this domain name, so we use the set type=mx command.
        - > set type=mx
    - Finally we enter the domain name that we want to query.
        - > bbc.co.uk
    - We can see that bbc.co.uk actually has 4 mx records set up, so if they were experiencing a loss of some emails we could then check each of these mail servers to track down the issue. Most customers will only have one (or sometimes two) MX records.
    - Next, let's see if there are any notes by checking for txt records associated with the domain name. So enter set type=txt and enter the domain name again.
        - > set type=txt
        - >bbc.co.uk

**telnet:**

- Telnet is a set of tools that allow users to remotely access another computer.
- **Adding Telnet to Vista and Windows 7**
    - Step-1: Open the control panel.
    - Step-2: Click programs>Uninstall a program (In classic view click Programs and features).
    - Step-3: Click Turn Windows features on or off.
    - Step-4: Scroll down the list of windows features and tick the box marked Telnet Client.
    - Step-4: Scroll down the list of windows features and tick the box marked Telnet Client.

**route:**

- The route command displays the computers routing table. A typical computer, with a single network interface, connected to a LAN, with a router is fairly simple and generally doesn't pose any network problems. But if you're having trouble accessing other computers on your network, you can use the route command to make sure the entries in the routing table are correct.
- C:\>route
- The route command shows and alters entries in the local routing table. The command syntax is:
    - route <options> <command> <value>
    - The different available commands are:
        - add - Adds a route entry to the table.
        - change - Modifies an entry in the table.
        - delete - Removes a route from the table.
        - print - Displays a route or routes
    - For example, to print all routes from the table, use: – route print

**arp:**

- It will show the IP address of your computer along with the IP address and MAC address of your router.
- The arp (address resolution protocol) command shows and modifies entries in the ARP cache.
- The cache contains one or multiple tables that map IP addresses to resolved physical addresses.
- The syntax for the command is
    - arp <options> <address>
- Without any parameters, the arp command shows the help window.
- To show the ARP cache table, run the following command:
    - C:\>arp –a

## Results and Observation:

We successfully executed various network commands like ping, ipconfig, and tracert. These commands displayed network configurations, tested connectivity, and helped identify network issues, confirming the proper functioning and configuration of the network.

## Viva-Voice Questions:

**Q) What command is used to display the IP configuration of your computer?**

**A.** The `ipconfig` command is used to display the IP configuration of your computer.

**Q) Which command helps to test the connectivity to another device on the network?**

**A.** The `ping` command helps to test the connectivity to another device on the network.

**Q) What is the purpose of the `tracert` command?**

**A.** The `tracert` command traces the route packets take to reach a destination, identifying each hop along the way.

**Q) How do you flush the DNS resolver cache on your computer?**

**A.** You flush the DNS resolver cache using the `ipconfig /flushdns` command.

**Q) Which command provides detailed information about all network adapters and their configurations?**

**A.** The `ipconfig /all` command provides detailed information about all network adapters and their configurations.

Date:

# Experiment No. 4

## Aim:

Analyze Network Traffic using Wireshark tool/ TCP dump tool

## Components and Tools:

System : Desktop Computer/Laptop

Operating system : Windows/Linux Tool: Wireshark

## Description:

### Wireshark Overview

Wireshark is a free and open-source packet analyzer used for network troubleshooting and analysis. It is useful for network engineers, security professionals, and system administrators.

### 1. Troubleshooting Network Connectivity

- **Visually Understand Packet Loss**: Detect and analyze packet loss in network traffic.
- **Review TCP Retransmission**: Identify and review retransmitted TCP packets.
- **Graph High Latency Packet Responses**: Visualize latency issues in packet responses.

### 2. Examination of Application Layer Sessions

- **HTTP Sessions**: View complete HTTP sessions, including all headers and data for both requests and responses.
- **Telnet Sessions**: Observe Telnet sessions, including commands entered, responses, and passwords (note: this may be less relevant for encrypted sessions).
- **SMTP/POP3 Traffic**: Analyze SMTP or POP3 traffic to read emails transmitted over the network.

### 3. Troubleshoot DHCP Issues

- **Examine DHCP Client Broadcast**: View DHCP client broadcast messages.
- **DHCP Offer**: Analyze DHCP offers including address and options provided by the server.
- **Client Requests**: Review client requests for offered addresses.
- **Server Acknowledgment (Ack)**: Confirm server acknowledgment of client requests.

**4. Extract Files from Sessions**

- **HTTP Sessions**: Export objects such as JavaScript, images, or executables from HTTP traffic.
- **SMB Sessions**: Extract files transferred over SMB (Server Message Block) protocol, commonly used for file sharing.

**5. Detect and Examine Malware**

- **Detect Anomalous Behavior**: Identify unusual network behavior that may indicate malware.
- **Search for Unusual Domains/IP Addresses**: Look for suspicious domains or IP endpoints.
- **IO Graphs**: Use IO graphs to find regular connections (beacons) to command and control servers.
- **Filter Traffic**: Differentiate between normal and unusual traffic patterns.
- **Extract Large DNS Responses**: Analyze large DNS responses and other anomalies that may suggest malware.

**6. Examination of Port Scans and Vulnerability Scans**

- **Understand Scanner Traffic**: Analyze the network traffic generated by vulnerability scanners.
- **Troubleshoot Vulnerability Checks**: Investigate false positives and negatives in vulnerability scanning results.

## Procedure:

### Preparation

1. **Turn On Your PC**
    - Press the power button to start your PC. Windows will start automatically.
    - Log in if necessary.

## Activity 1 - Determine System Type

1. **To Determine System Type:**
    - Open the Run dialog by pressing the Windows key and typing "run."
    - Type `msinfo32` and press Enter to display the system information.
    - The system type will be listed as either X86-based PC (32-bit) or X64-based PC (64-bit).
    - Close `msinfo32`.

## Activity 2 - Download Wireshark

1. **To Download Wireshark:**
    - Open a web browser.
    - Navigate to Wireshark Downloads.
    - Click on **Download Wireshark**.

- Choose the Wireshark Windows Installer that matches your system type (32-bit or 64-bit as determined in Activity 1).
- Save the installer in the Downloads folder.
- Close the web browser.

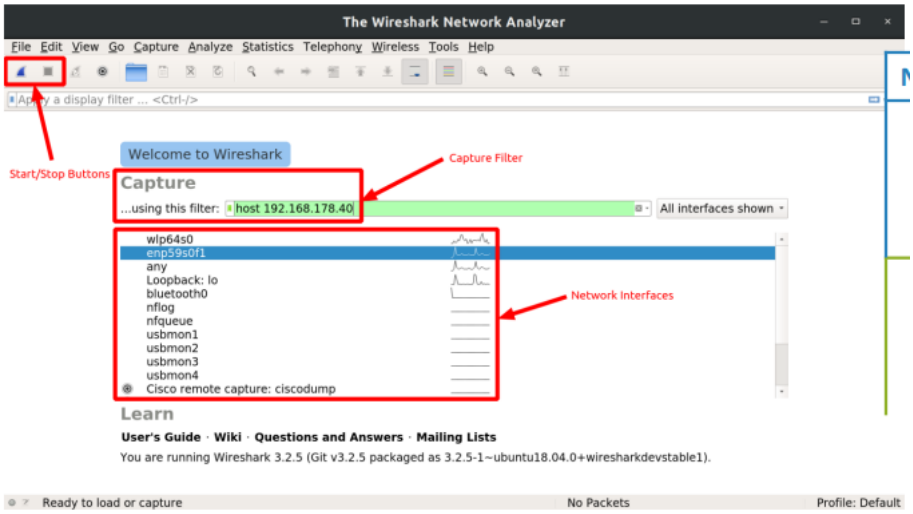
## Wireshark Installation

### Activity 3 - Install Wireshark

1. **To Install Wireshark:**
    - Open Windows Explorer and go to the Downloads folder.
    - Locate the Wireshark installer downloaded in Activity 2. Double-click on the file to open it.
    - If prompted by a User Account Control dialog box, select **Yes** to allow the program to make changes to your computer.
    - Click **Next >** to start the Setup Wizard.
    - Review the license agreement. If you agree, select **I Agree** to continue.
    - Click **Next >** to accept the default components.
    - Choose the shortcuts you want to create and leave the file extensions selected. Click **Next >** to continue.
    - Click **Next >** to accept the default install location.
    - Click **Install** to begin the installation.
    - Click **Next >** to install WinPcap.
    - Review the WinPcap license agreement. If you agree, select **I Agree** to continue.
    - Click **Install** to begin the WinPcap installation.
    - Click **Finish** to complete the WinPcap installation.
    - Click **Next >** to continue with the installation of Wireshark.
    - Click **Finish** to complete the installation of Wireshark.

## Capturing Packets with Wireshark

1. **Launch Wireshark**
    - After downloading and installing Wireshark, open the application.



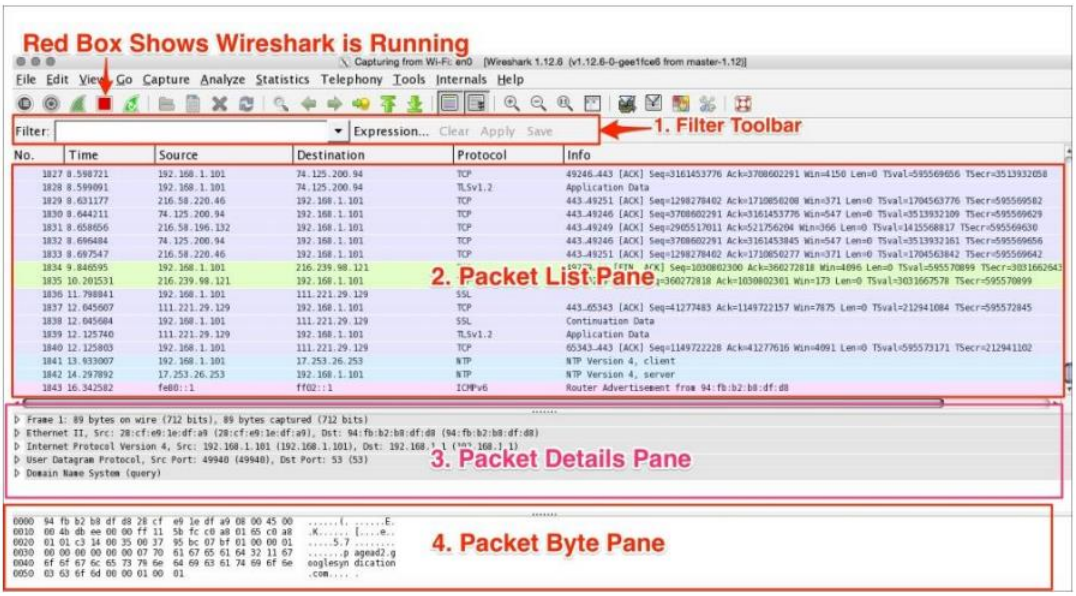
2. **Start Capturing Packets**
   - In the main window, locate the **Capture** section.
   - Double-click the name of the network interface you wish to capture packets on. For example, to capture traffic on your wireless network, select your wireless interface.
   - Alternatively, you can configure advanced settings by clicking **Capture** > **Options**, but this is optional for now.

**Wireshark Capturing Modes**

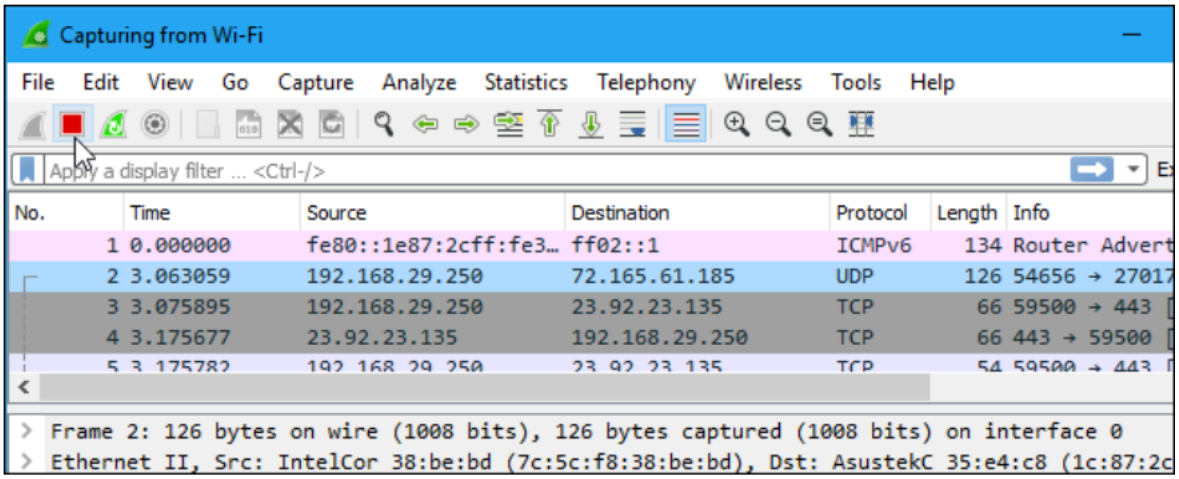
| NAME | DESCRIPTION |
|---|---|
| Promiscuous mode | Sets interface to capture all packets on a network segment to which it is associated to |
| Monitor mode | Setup the wireless interface to capture all traffic it can receive (Unix/ Linux only) |

3. **View Captured Packets**
   - As soon as you click on the interface's name, you will see packets start to appear in real-time. Wireshark captures each packet sent to or from your system.

4. **Promiscuous Mode**
   - If promiscuous mode is enabled (enabled by default), you will see all packets on the network, not just those addressed to your network adapter.
   - To check if promiscuous mode is enabled:
     - Click **Capture** > **Options**.
     - Verify that the "Enable promiscuous mode on all interfaces" checkbox is checked at the bottom of the window.

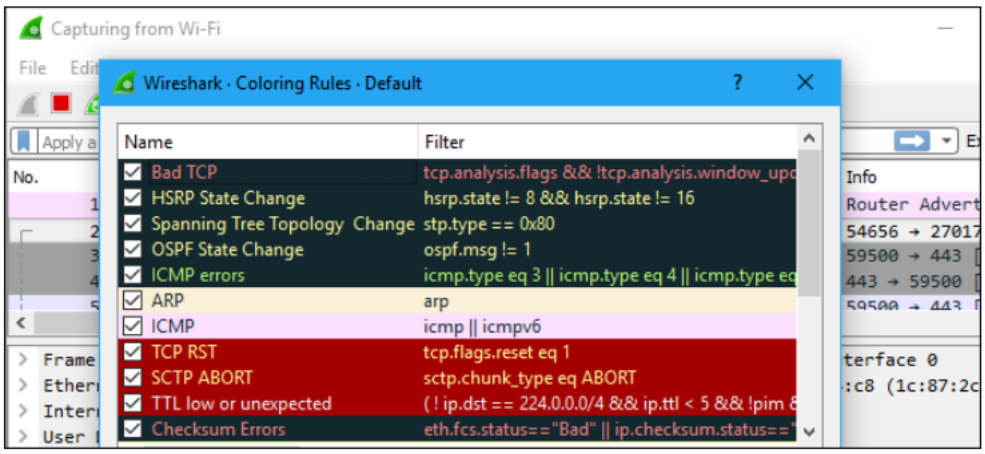


5. **Stop Capturing Traffic**
   - When you want to stop capturing traffic, click the red **Stop** button near the top left corner of the window.



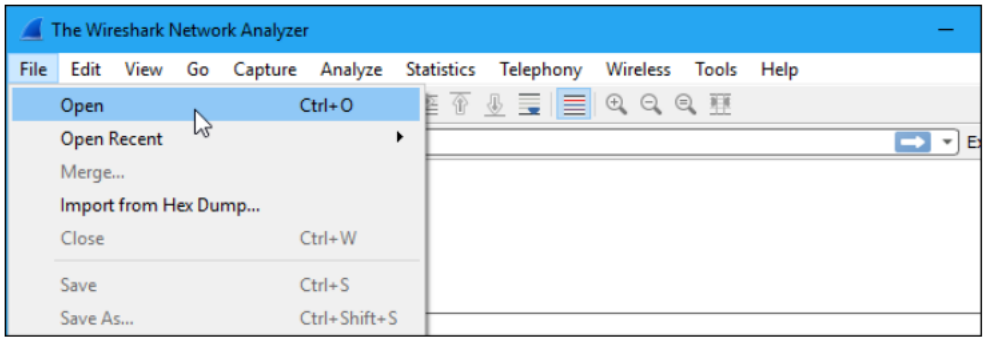

## Color Coding

- **Default Colors:**
  - **Light Purple**: TCP traffic
  - **Light Blue**: UDP traffic
  - **Black**: Packets with errors (e.g., delivered out of order)
- **View and Customize Color Codes:**
  - Click **View** > **Coloring Rules** to see the meaning of the color codes.
  - You can also modify and customize the coloring rules from this menu.



## Sample Captures

- **Using Sample Captures:**
  - If your own network lacks interesting traffic, use sample capture files from Wireshark's wiki.
  - Click **File > Open** in Wireshark and browse for your downloaded sample file to open it.
- **Save Your Own Captures:**
  - To save your captures, click **File > Save** and save the packet capture for future use.




### Default Columns in a Packet Capture Output

| NAME | DESCRIPTION |
|---|---|
| **No.** | **Frame number from the beginning of the packet capture** |
| **Time** | **Seconds from the first frame** |
| **Source (src)** | **Source address, commonly an IPv4, IPv6, or Ethernet address** |
| **Destination (dst)** | **Destination address** |
| **Protocol** | **Protocol used in the Ethernet frame, IP packet, or TCP segment** |
| **Length** | **Length of the frame in bytes** |

## Filtering Packets

- **Basic Filtering:**
  - Type your filter into the filter box at the top of the window and click **Apply** (or press Enter). For example, type `dns` to filter for DNS packets.
  - Wireshark will autocomplete your filter as you type.
- **Advanced Filtering:**
  - Click **Analyze > Display Filters** to choose from default filters or create and save custom filters for future use.
  - For more information on display filtering, refer to the Building Display Filter Expressions page in the Wireshark documentation.
- **Filter Types:**
  - **Capture Filter**: Filter packets during capture.
  - **Display Filter**: Hide packets from the capture display.
- **Follow TCP Stream:**
  - Right-click a packet and select **Follow > TCP Stream** to see the full TCP conversation between the client and the server.
  - You can also follow other protocols if applicable. Closing the window will apply a filter showing only the packets in the conversation.

### Logical Operators

| OPERATOR | DESCRIPTION | EXAMPLE |
|---|---|---|
| and or && | Logical AND: All conditions must match | ip.addr == 192.168.1.1 and tcp.port == 80 |
| or or ** | | ** |
| xor or ^^ | Logical XOR: Only one condition should match, not both | ip.addr == 192.168.1.1 xor ip.addr == 192.168.1.2 |
| not or ! | Not (Negation): Condition does not match | ip.addr != 192.168.1.1 |
| [ n ] or [ … ] | Substring operator: Filter specific text | http.request.uri contains "example" |
| eq or == | Equal | ip.dst == 192.168.1.1 |
| ne or != | Not equal | ip.dst != 192.168.1.1 |
| gt or > | Greater than | frame.len > 100 |
| lt or < | Less than | frame.len < 100 |
| ge or >= | Greater than or equal | frame.len >= 100 |
| le or <= | Less than or equal | frame.len <= 100 |

- **Miscellaneous:**
  - **Slice Operator**: `* … +` – Range of values
  - **Membership Operator**: `{}` - In
  - **CTRL+E**: Start/Stop Capturing

## Filter Syntax

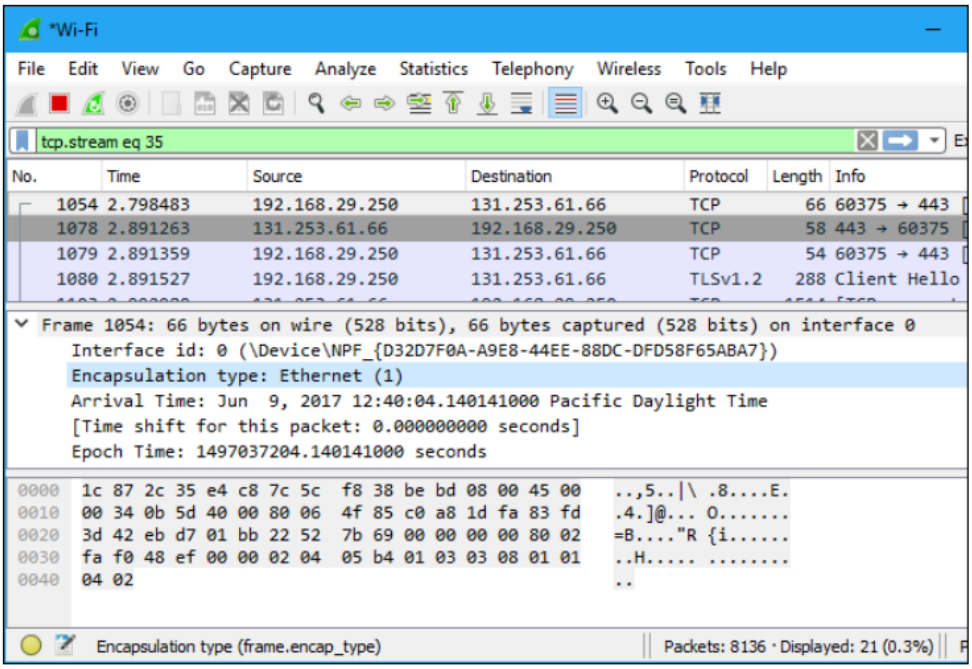
- **Capture Filter Syntax:**
  - `SYNTAX PROTOCOL DIRECTION HOSTS VALUE LOGICAL OPERATOR EXPRESSIONS`
  - Example: `tcp src 192.168.1.1 80 and tcp dst 202.164.30.1`
- **Display Filter Syntax:**
  - `SYNTAX PROTOCOL STRING 1 STRING 2 COMPARISON OPERATOR VALUE LOGICAL OPERATOR EXPRESSIONS`
  - Example: `http dest ip == 192.168.1.1 and tcp port`

## Common Filtering Commands

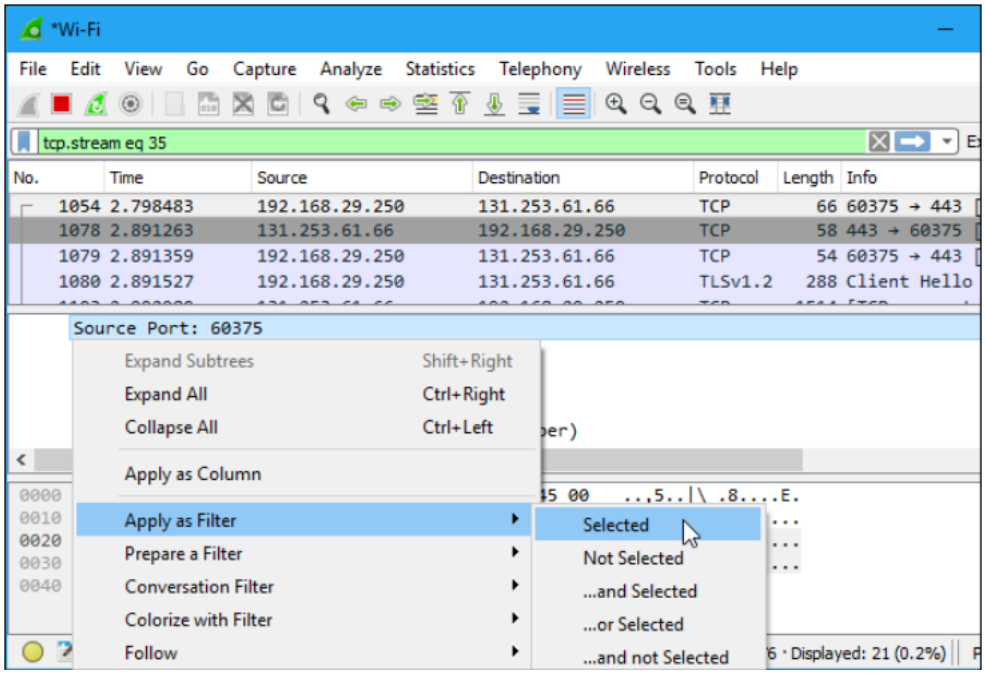
| USAGE | FILTER SYNTAX |
|---|---|
| **Filter by IP** | `ip.addr == 10.10.50.1` |
| **Filter by Destination IP** | `ip.dst == 10.10.50.1` |
| **Filter by Source IP** | `ip.src == 10.10.50.1` |
| **Filter by IP Range** | `ip.addr >= 10.10.50.1 and ip.addr <= 10.10.50.100` |
| **Filter by Multiple IPs** | `ip.addr == 10.10.50.1 or ip.addr == 10.10.50.100` |
| **Filter out IP Address** | `!(ip.addr == 10.10.50.1)` |
| **Filter Subnet** | `ip.addr == 10.10.50.1/24` |
| **Filter by Port** | `tcp.port == 25` |
| **Filter by Destination Port** | `tcp.dstport == 23` |
| **Filter by IP Address and Port** | `ip.addr == 10.10.50.1 and tcp.port == 25` |
| **Filter by URL** | `http.host == "hostname"` |
| **Filter by Timestamp** | `frame.time >= "June 02, 2019 18:04:00"` |
| **Filter SYN Flag** | `tcp.flags.syn == 1 and tcp.flags.ack == 0` |
| **Beacon Filter** | `wlan.fc.type_subtype == 0x08` |
| **Broadcast Filter** | `eth.dst == ff:ff:ff:ff:ff:ff` |
| **Multicast Filter** | `(eth.dst[0] & 1)` |
| **Hostname Filter** | `ip.host == "hostname"` |
| **MAC Address Filter** | `eth.addr == 00:70:f4:23:18:c4` |
| **RST Flag Filter** | `tcp.flags.reset == 1` |

## Inspecting Packets

- **View Packet Details:**
  - Click on a packet to select it and view its detailed information.




  - Right-click on a detail and use **Apply as Filter** to create a filter based on that detail.



**Report**
**Part I**

**1. What is Wireshark and how do you use it?**
**A.** Wireshark is a network protocol analyzer used for network troubleshooting, analysis, software and protocol development, and education. To use it, you install Wireshark on your computer, start it, and then select the network interface you want to capture traffic from. Once you start capturing, Wireshark will record all packets transmitted over the network, which can then be analyzed in detail.

**2. Can Wireshark see texts?**
**A.** Wireshark can capture and display textual data transmitted over the network if it is in plain text or if it is part of a protocol that transmits text in an unencrypted format (e.g., HTTP). However, encrypted communications (e.g., HTTPS) will not be readable in plain text.

**3. What are the 2 types of filters used by Wireshark?**
**A.** The two types of filters used by Wireshark are:

- **Capture filters**: These are set before capturing starts and determine which packets will be captured based on specified criteria.
- **Display filters**: These are applied after capturing and allow you to refine the view of the captured packets based on specified criteria.

**4. Can Wireshark see incognito?**
**A.** Wireshark can capture packets from incognito or private browsing sessions because it monitors network traffic rather than the browser's state. However, it cannot decrypt encrypted traffic, so if the incognito session uses HTTPS, the content will remain encrypted.

**5. How do you capture packets in Wireshark?**
**A.** To capture packets in Wireshark, follow these steps:

1. Open Wireshark.
2. Select the network interface you want to capture traffic from.
3. Click the "Start Capturing Packets" button (the shark fin icon).
4. Wireshark will begin capturing all network traffic on the selected interface.

**6. What can hackers do with Wireshark?**
**A.** Hackers can use Wireshark to:

- **Monitor network traffic**: Capture and analyze packets to gather information about the network and its users.
- **Intercept sensitive data**: Extract unencrypted data from packets if the communication is not secured.
- **Discover vulnerabilities**: Identify weaknesses in the network or application protocols that can be exploited.

**Part II**

**1. Is the frame an outgoing or an incoming frame?**
**A.** This depends on the specific context of the frame in question. In general, a frame could be either outgoing or incoming based on whether it is being sent from or received by the device capturing the traffic.

**2. Source IP address of the network-layer header in the frame:**
**A.** The source IP address in the provided data is `72.14.207.99`.

**3. Destination IP address of the network-layer header in the frame:**
**A.** The destination IP address in the provided data is `216.58.208.99`.

**4. Total number of bytes in the whole frame:**
**A.** The total number of bytes in the whole frame is not directly provided in the text. This typically includes the header and payload and would be found in the frame length field.

**5. Number of bytes in the Ethernet (data-link layer) header:**
**A.** The Ethernet header is typically 14 bytes.

**6. Number of bytes in the IP header:**
**A.** The IP header is typically 20 bytes, but this can vary if options are included.

**7. Number of bytes in the TCP header:**
**A.** The TCP header is typically 20 bytes, but can be longer if options are included.

**8. Total bytes in the message (at the application layer):**
**A.** The total bytes in the message at the application layer can be determined by subtracting the bytes used by the Ethernet, IP, and TCP headers from the total frame size. Based on the given data, it is not explicitly provided but can be inferred from the payload.

## Result and Observation:

The experiment demonstrates how Wireshark captures and analyzes network traffic, providing insights into both encrypted and unencrypted data. By examining packet headers and payloads, users can effectively monitor network activity and detect potential vulnerabilities, highlighting the importance of secure communications and thorough traffic analysis.

## Viva-Voice Questions:

**Q) What is the primary purpose of using Wireshark in network analysis?**

**A.** Wireshark is used to capture and analyze network traffic, allowing users to inspect packet details and troubleshoot network issues.

**Q) How does Wireshark differentiate between encrypted and unencrypted traffic?**

**A.** Wireshark identifies encrypted traffic through the absence of readable data in the payload and specific protocols like HTTPS or TLS in the packet headers.

**Q) What is a common challenge when analyzing encrypted traffic in Wireshark?**

**A.** The primary challenge is that the payload of encrypted traffic is not readable without decryption keys, limiting the analysis to header information only.

**Q) How can packet filtering in Wireshark improve your analysis efficiency?**

**A.** Packet filtering helps by narrowing down the captured data to relevant packets, making it easier to focus on specific traffic or issues.

**Q) Why is it important to understand both TCP and UDP protocols in network analysis with Wireshark?**

**A.** Understanding TCP and UDP is crucial because they handle data transmission differently—TCP ensures reliable, ordered delivery, while UDP provides faster, connectionless communication.

Date:

# Experiment No. 5

## Aim:

Implement Framing Mechanism using Python Programming Language

## Components and Tools:

| | |
|---|---|
| System | : Desktop Computer/Laptop |
| Operating System | : Windows/Linux |
| IDE | : Spyder |
| Language | : Python |

## Description:

### Task 5.1: Character Stuffing and Destuffing

In a data communication system, information is transmitted between a sender and a receiver using a character-based approach. To ensure data integrity and synchronization, character stuffing and destuffing techniques are employed. Character stuffing involves adding special control characters before and after a designated delimiter character in the data stream. This helps distinguish between data characters and control characters, ensuring that the receiver can accurately interpret the transmitted information. Destuffing is the process of removing the added control characters during reception.

**Character Stuffing:** In character stuffing, the sender follows these rules:

- A predefined delimiter character is chosen (e.g., the character sequence DLE STX).
- Before transmitting the data, the sender inserts a control character (e.g., DLE) before any occurrence of the delimiter character within the data.
- The sender also adds an additional control character (e.g., DLE) at the end of the data.
- The modified data is transmitted to the receiver.

**Character Destuffing:** The receiver follows these rules to destuff the received data:

- Upon receiving data, the receiver scans for occurrences of the control character before the delimiter character (e.g., DLE) and removes the control character, treating the delimiter character as a regular data character.
- When the receiver encounters the control character at the end of the data, it simply removes the control character.
- The receiver processes the destuffed data stream, interpreting the original data and recognizing the delimiter character as the end of a data frame.

**Problem Scenario:** Imagine two functions are communicating using character stuffing and character destuffing. The sender has a string of characters that need to be transmitted, and the receiver needs to correctly interpret this data.

**Input Data (character stuffing):** This is the data DLE STX that needs DLE to be stuffed DLE ETX with characters.

**Output Data (after stuffing and before transmission):** This is the data DLE DLE STX that needs DLE DLE to be stuffed DLE DLE ETX with characters.

**Received Data (character destuffing, after transmission and before destuffing):** This is the data DLE DLE STX that needs DLE DLE to be stuffed DLE DLE ETX with characters.

**Output Data (after destuffing):** This is the data DLE STX that needs DLE to be stuffed DLE ETX with characters.

**Task 5.2: Bit Stuffing and Destuffing**

In a serial data transmission system, data is sent as a sequence of bits. However, the sender and receiver might not be perfectly synchronized due to differences in clock speeds, signal propagation delays, and other factors. If the sender sends a long sequence of identical bits (e.g., several consecutive 1s or 0s), the receiver might lose track of where the bit boundaries are, leading to synchronization errors and misinterpretation of the data.

Bit stuffing is a technique used to prevent this problem. When transmitting data, the sender inserts an extra "0" bit after a certain number of consecutive identical bits in the data stream. The receiver knows to remove the extra "0" bits and interpret the original data correctly.

The rules for bit stuffing are as follows:

- Whenever the sender encounters a sequence of "n" consecutive identical bits (where "n" is a predetermined threshold, typically 5), it inserts an extra "0" bit after the "n-1" consecutive bits.
- The receiver recognizes this pattern and removes the extra "0" bit whenever it detects "n" consecutive identical bits followed by a "0" bit.

**Problem Scenario:** Imagine two functions are communicating using BitStuffing and BitDestuffing. The sender has a string of bits that need to be transmitted, and the receiver needs to correctly interpret this data.

1. **BitStuffing**
   - **Input Frame (before bit stuffing):** 0111110 10111111010111110111111101 01111110
   - **Output Frame (after bit stuffing):** 0111110 1011111010101111100111110110 1 0111110
2. **BitDestuffing**
   - **Input Frame (after bit stuffing):** 0111110 1011111010101111100111110110 1 0111110
   - **Output Frame (after destuffing):** 0111110 10111111010111110111111101 01111110

## Algorithm:

### Character Stuffing Algorithm:

1. Initialize an empty string `stuffed` to hold the stuffed data.
2. Loop through the data string in steps of 3 characters:
   - If a substring of 3 characters is equal to a delimiter or the escape character, append the escape character to `stuffed` followed by the substring.
   - Increment the index by 3.
3. Return the `stuffed` string.

### Character Destuffing Algorithm:

1. Initialize an empty string `res` to hold the destuffed data.
2. Loop through the stuffed string in steps of 3 characters:
   - If a substring of 3 characters is equal to the escape character, skip the escape character and append the following 3 characters to `res`.
   - Increment the index by 3.
   - Otherwise, append the 3-character substring to `res`.
3. Return the `res` string.

### Bit Stuffing Algorithm:

1. Initialize an empty string `stuffed` to hold the stuffed bits and a counter `count` to count consecutive 1s.
2. Loop through each bit in the data:
   - If the bit is '1', increment `count` and append the bit to `stuffed`.
   - If `count` reaches 5, append a '0' to `stuffed` and reset `count` to 0.
   - If the bit is '0', reset `count` to 0 and append the bit to `stuffed`.
3. Return the `stuffed` string.

**Bit Destuffing Algorithm:**

1. Initialize an empty string `destuffed` to hold the destuffed bits and a counter `count` to count consecutive 1s.
2. Loop through each bit in the stuffed data:
   - If the bit is '1', increment `count` and append the bit to `destuffed`.
   - If `count` reaches 5 and the next bit is '0', skip the next bit and reset `count` to 0.
   - If the bit is '0', reset `count` to 0 and append the bit to `destuffed`.
3. Return the `destuffed` string.

## Program:

```
def char_stuffing(data, flag=['STX', 'ETX'], esc='DLE'):
    stuffed = ''
    i = 0
    while i < len(data):
        if data[i:i+3] in flag or data[i:i+3] == esc:
            stuffed += esc
            stuffed += data[i:i+3]
            i += 3
        else:
            stuffed += data[i]
            i += 1
    return stuffed

def char_destuffing(stuffed, flag=['STX', 'ETX'], esc='DLE'):
    res = ''
    i = 0
    while i < len(stuffed):
        if stuffed[i:i+3] == esc:
            i += 3
            res += stuffed[i:i+3]
            i += 3
        else:
```

```
            res += stuffed[i]
            i += 1
    return res

def bit_stuffing(data):
    stuffed = ''
    count = 0
    for bit in data:
        if bit == '1':
            stuffed += bit
            count += 1
            if count == 5:
                stuffed += '0'
                count = 0
        else:
            count = 0
            stuffed += bit
    return stuffed

def bit_destuffing(data):
    destuffed = ''
    count = 0
    i = 0
    while i < len(data):
        bit = data[i]
        if bit == '1':
            count += 1
            if count == 5:
                if i + 1 < len(data) and data[i + 1] == '0':
                    i += 1
                count = 0
```

```
        else:
            count = 0
        destuffed += bit
        i += 1
    return destuffed

if __name__ == "__main__":
    message_chars = input("Enter your message (Special
characters - DLE, STX, ETX):")
    stuffed_chars = char_stuffing(message_chars)
    print("Character Stuffed Message:", stuffed_chars)
    destuffed_chars = char_destuffing(stuffed_chars)
    print("Character Destuffed Message:", destuffed_chars)

    message_bits = input("Enter your message (Binary):")
    stuffed_bits = bit_stuffing(message_bits)
    print("Bit Stuffed Message:", stuffed_bits)
    destuffed_bits = bit_destuffing(stuffed_bits)
    print("Bit Destuffed Message:", destuffed_bits)
```

**Output:**

```
G VENKATA SAI RAM@GVSR-Laptop MINGW64 /d/B-Tech/PVP/3rd Year/1st Sem/
$ "C:/Users/G VENKATA SAI RAM/AppData/Local/Microsoft/WindowsApps/pyt
Stuffing.py"
Enter your message (Special characters - DLE, STX, ETX): STXDLEETX
Character Stuffed Message:  DLESTXDLEDLEDLEETX
Character Destuffed Message:  STXDLEETX
Enter your message (Binary):1000001111011111101
Bit Stuffed Message: 10000011110111110101
Bit Destuffed Message: 1000001111011111101
```

## Viva-Voice Questions:

**Q) What is the primary purpose of character stuffing in data communication?**

**A.** To distinguish between data characters and control characters, ensuring accurate interpretation by the receiver

**Q) Why is the escape character used in character stuffing?**

**A.** To indicate the presence of a delimiter or another escape character within the data stream

**Q) Explain what happens when the receiver encounters the escape character during character destuffing**

**A.** The receiver removes the escape character and treats the following character as a regular data character

**Q) What problem does bit stuffing solve in serial data transmission?**

**A.** It prevents synchronization errors caused by long sequences of identical bits, ensuring proper bit boundary recognition

**Q) Describe the process of bit destuffing**

**A.** The receiver removes any extra '0' bits inserted after sequences of a predetermined number of consecutive '1' bits

Date:

# Experiment No. 6

## Aim:

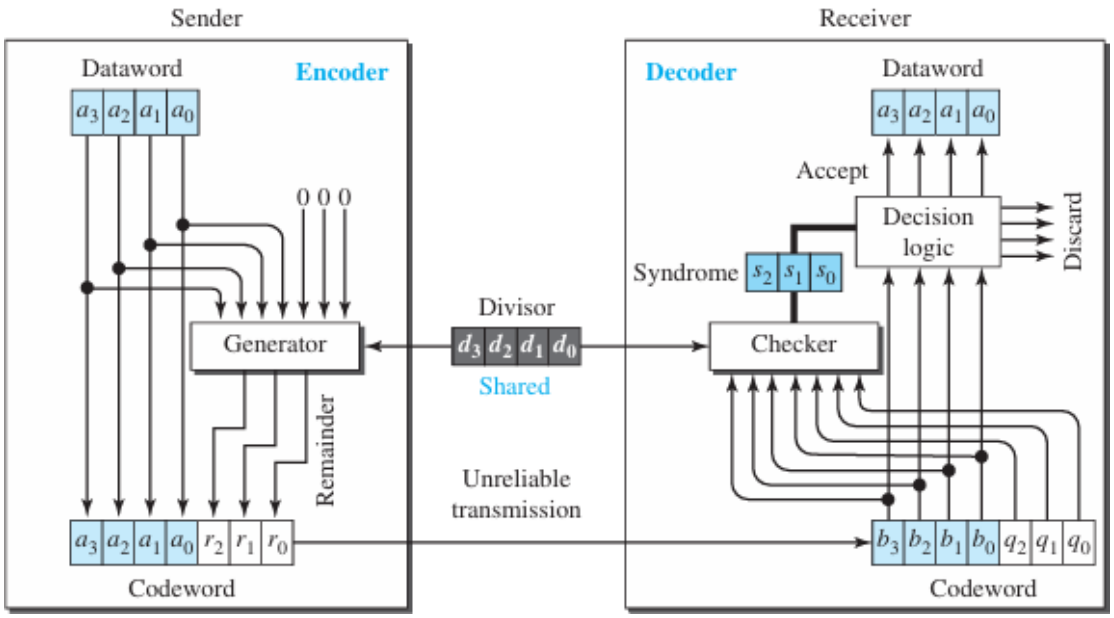
Implement Error Detection technique using Python Programming Language

## Components and Tools:

| | |
|---|---|
| System | : Desktop Computer/Laptop |
| Operating System | : Windows/Linux |
| IDE | : Spyder |
| Language | : Python |

## Description:

In digital communication systems, the Cyclic Redundancy Check (CRC) is a widely used error-detection technique that helps ensure the integrity of transmitted data. It involves appending a checksum (remainder) to the data before transmission and verifying it at the receiver's end to detect any errors that might have occurred during transmission. Your task is to implement a CRC-based error detection system for a given dataset



**Problem Scenario:**

You are provided with the following details:

- A polynomial divisor in binary representation (e.g., `1101`).
- A dataset, represented in binary, needs to be transmitted over a noisy communication channel.

Your tasks are:

1. **Implement a CRC-based error detection system** that appends the appropriate CRC checksum to the dataset before transmission.
2. **Simulate the process of data transmission** by potentially introducing errors into the transmitted dataset.
3. **Implement the receiver's side** to verify the integrity of the received dataset using CRC error detection.
4. **Identify and report any errors** that might have occurred during transmission.

**Input:**

- Dataset to be transmitted (binary format).
- Polynomial divisor for CRC (binary format).

**Output:**

- Transmitted dataset (potentially with introduced errors).
- Received dataset.
- Detection of whether errors were present or not.
- If errors are detected, identify the positions of errors.

**Example:**

**Given:**

- Dataset: `11010101101` (binary)
- Polynomial Divisor: `1011` (binary)

**Transmission:**

1. **Sender appends the CRC checksum to the dataset:**
    - Transmitted dataset: `11010101101100` (binary) [Append the CRC checksum]
2. **Transmission with Errors:**
    - Errors are introduced during transmission: `11010101111100` (binary)

**Reception:**

1. **Receiver receives the potentially erroneous dataset:**
    - Received dataset: `11010101111100` (binary)
2. **Receiver performs CRC error detection:**
    - Errors are detected.

**Error Detection Outcome:**

- **Errors were detected** in the received dataset.
- **Error positions:** `10` (binary index positions where errors occurred)

## Algorithm:

### 1. Initialization

1. **Input Data:**
   - Read the data to be transmitted.
   - Read the polynomial divisor.

### 2. Encoding Process

1. **Prepare the Data:**
   - Append `n-1` zeros to the end of the data, where `n` is the length of the polynomial divisor. This forms the `dividend`.
2. **Division Process:**
   - Initialize the `partXor` with the first segment of the `dividend` equal to the length of the divisor.
   - For each bit in the `dividend` (from the start to the end):
     - Perform the XOR operation between `partXor` and the divisor to get the `remainder`.
     - Append the next bit of the `dividend` to the `remainder`.
     - Update `partXor` with the new `remainder`.
3. **Obtain the Final Remainder:**
   - The final `remainder` after processing the entire `dividend` is the CRC checksum.
4. **Output the Remainder:**
   - This remainder is appended to the original data to create the transmitted data.

### 3. Decoding Process

1. **Prepare the Received Data:**
   - The received data includes the original data and the CRC checksum.
2. **Division Process:**
   - Initialize the `partXor` with the first segment of the received data equal to the length of the divisor.
   - For each bit in the received data (from the start to the end):
     - Perform the XOR operation between `partXor` and the divisor to get the `remainder`.
     - Append the next bit of the received data to the `remainder`.
     - Update `partXor` with the new `remainder`.
3. **Obtain the Final Remainder:**
   - The final `remainder` after processing the entire received data indicates whether an error has occurred.
4. **Check for Errors:**
   - If the final `remainder` contains any `1`s, an error is detected.

- If the remainder is all `0`s, no error is detected.

**4. Example Run**

1. **Encoding Example:**
   - Input Data: `1101`
   - Divisor: `1011`
   - Transmitted Data: `1101011` (original data + CRC remainder)
2. **Decoding Example:**
   - Received Data: `1101011`
   - Final Remainder: `0000` (no error)
   - With an Error Example:
   - Received Data: `110101010011010011`
   - Final Remainder: Non-zero (error detected)

## Program:

```
def getXor(partXor, div):
    remainder = ""
    if partXor[0] == '0':
        return partXor[1:]
    for i in range(1, len(div)):
        if partXor[i] == div[i]:
            remainder += '0'
        else:
            remainder +='1'
    return remainder


def encoder(div, data):
    dividend = data + ("0")*(len(div)-1)
    starter = 0
    borrower = len(div)
    partXor = dividend[starter: borrower]
    for _ in range(len(dividend) - len(div) + 1):
        remainder = getXor(partXor, div)
        if borrower >= len(dividend):
            break
```

```
        remainder += dividend[borrower]
        starter += 1
        borrower += 1
        partXor = remainder
    return getXor(partXor, div)

def decoder(div, data):
    dividend = data
    print("Transmitted data: ", dividend)
    starter = 0
    borrower = len(div)
    partXor = dividend[starter: borrower]
    for _ in range(len(dividend) - len(div) + 1):
        remainder = getXor(partXor, div)
        if borrower >= len(dividend):
            break
        remainder += dividend[borrower]
        starter += 1
        borrower += 1
        partXor = remainder
    return getXor(partXor, div)


if __name__ == "__main__":
    dataset = input("Enter the data to be transmitted: ")
    divisor = input("Enter the Polynomial divisor for CRC: ")

    print('------------------------')
    print('Encoder Process Started:')
    dividendUtil1 = encoder(divisor, dataset)
    print("Final remainder (from encoder): ", end = "")
```

```
    print(dividendUtil1)
    print('------------------------')

    print('Decoder Process Started:')
    data = dataset + dividendUtil1
    dividendUtil2 = decoder(divisor, data)
    print("Final remainder (from decoder): ", end = "")
    print(dividendUtil2)
    if('1' not in dividendUtil2):
        print('No error detected')
    print('------------------------')

    print('Decoder Process with error data')
    error = '110101010011010011'
    dividendUtil3 = decoder(divisor, error)
    print("Final remainder (from error decoder): ", end = "")
    print(dividendUtil3)
    if('1' in dividendUtil3):
        print('Error detected')
```

**Output:**

```
(.venv)
G VENKATA SAI RAM@GVSR-Laptop MINGW64 C:/Users/G VE
$ python -u "d:\OneDrive\Language Specific Projects
Enter the data to be transmitted: 1101010100110
Enter the Polynomial divisor for CRC: 100011
------------------------
Encoder Process Started:
Final remainder (from encoder): 01100
------------------------
Decoder Process Started:
Transmitted data:  110101010011001100
Final remainder (from decoder): 00000
No error detected
------------------------
Decoder Process with error data
Transmitted data:  110101010011010011
Final remainder (from error decoder): 11111
Error detected
```

## Result:

The CRC-based error detection mechanism was successfully implemented using Python. The transmitted data was encoded with a CRC checksum, appended to the original data, and then accurately verified at the receiver's end. The received data was checked against the calculated CRC to ensure its integrity, allowing for the detection and handling of transmission errors.

## Viva-Voice Questions:

**Q) What is the purpose of CRC in data communication?**

**A.** To detect errors in transmitted data by generating and verifying a checksum.

**Q) How does CRC error detection work?**

**A.** CRC uses polynomial division to create a checksum that the receiver uses to verify data integrity.

**Q) What is a generator polynomial in CRC?**

**A.** A generator polynomial is a predefined polynomial used to compute the CRC checksum and verify data integrity.

**Q) Why is polynomial degree important in CRC?**

**A.** The polynomial degree determines the length of the CRC checksum, affecting its error-detection capability.

**Q) How does CRC compare to other error-detection methods?**

**A.** CRC is more robust than simple checksums and parity bits, detecting a wider range of error types due to its polynomial-based approach.

Date:

# Experiment No. 7

## Aim:

To configure a Host IP, Subnet Mask and Default Gateway of a device in LAN.

To establish Peer to Peer network connection using Switch and Router in a LAN.

## Components and Tools:

System : Desktop Computer/Laptop
Operating System : Windows/Linux
Tool : Packet Tracer

## Description:

**Classification of IP address**

| Class | Address Range | Supports |
|---|---|---|
| Class A | 1.0.0.1 to 126.255.255.254 | Supports 16 million hosts on each of 127 networks. |
| Class B | 128.1.0.1 to 191.255.255.254 | Supports 65,000 hosts on each of 16,000 networks. |
| Class C | 192.0.1.1 to 223.255.254.254 | Supports 254 hosts on each of 2 million networks. |
| Class D | 224.0.0.0 to 239.255.255.255 | Reserved for multicast groups. |
| Class E | 240.0.0.0 to 254.255.255.254 | Reserved. |

**Sub netting,** why we Develop sub netting and How to calculate subnet mask and how to identify subnet address.

**Default subnet masks:**

- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class C: 255.255.255.0

**Super netting,** why we develop super netting and how to calculate super net mask and how to identify super net address.

## Task-1: Configure Host IP, Subnet Mask and Default Gateway in a System in LAN (TCP/IP Configuration).

## Procedure:

1. Click on the Start button and select Control Panel then Network and Internet Connections.
2. Click Network and Internet Connections.
3. Right click on the Local Area Connection icon and select Properties.
4. Select Internet Protocol (TCP/IP).
5. Click on the Properties button.
6. Uncheck that Obtain an IP address automatically and Obtain DNS server address automatically and put IP, Subnet mask & Default Gateways. (write your IP)
    - IP Address: 172.50.30.62
    - Subnet mask: 255.255.224.0
    - Default gateway: 172.50.5.203
    - Preferred DNS server: 172.50.5.203
7. Click on the advanced button and select the DNS tab in the Advanced TCP/IP Settings window.
8. Ensure that register this connection's addresses in DNS is not selected.
9. Click OK, OK, then Close to close all boxes.

## Task-2: Establish a simple LAN using Switch.

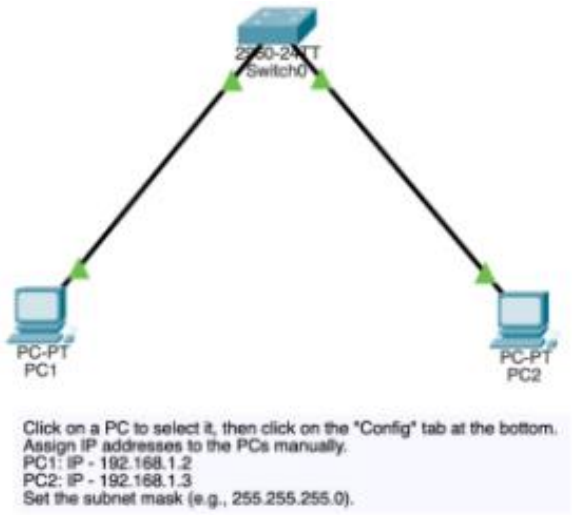
### 1. Creating a New Packet Tracer Project

*Launch Packet Tracer:*
Open the Packet Tracer application on your computer.

*Create a New Project:*
Click on “File” and then select “New” to start a new Packet Tracer project.

### 2. Designing the LAN Topology

In this example, we’ll configure a simple LAN consisting of two computers connected to a switch. Follow these steps to design the topology:




*Add Devices:*
From the left panel, drag and drop devices onto the workspace. For our basic LAN, use PCs as end devices and a switch to connect them.

*Connect Devices:*
Use copper straight-through cables to connect the PCs to the switch. Click on the “Connections” button on the left panel, select

the copper straight-through cable, and connect the devices by clicking on the desired interfaces on the switch and PCs.

*Arrange Devices:*
Organize the devices on the workspace for a clear and logical layout. This step is optional but can enhance the visual representation of your network.

**3. Configuring IP Addresses on PCs**

For devices in a LAN to communicate, each device must have a unique IP address within the same subnet. Follow these steps to configure IP addresses on the PCs:

*Select a PC:*
Click on one of the PCs to select it.

*Access Configuration Tab:*
Click on the "Config" tab at the bottom of the screen to access the configuration options for the selected PC.

*Assign IP Address:*
Manually assign an IP address to PC 1. For example:

- PC1: IP – 192.168.1.2
  Set the subnet mask, such as 255.255.255.0.

*Repeat for Other PCs:*
Repeat the process for the other PCs, ensuring that each PC has a unique IP address within the same subnet.

- PC2: IP – 192.168.1.3
  Set the subnet mask, such as 255.255.255.0.

**4. Testing Connectivity**

Once the configuration is complete, it's essential to test connectivity to ensure that devices within the LAN can communicate effectively. Follow these steps to test connectivity:

*Open Command Prompt:*
Open the command prompt on one of the PCs.

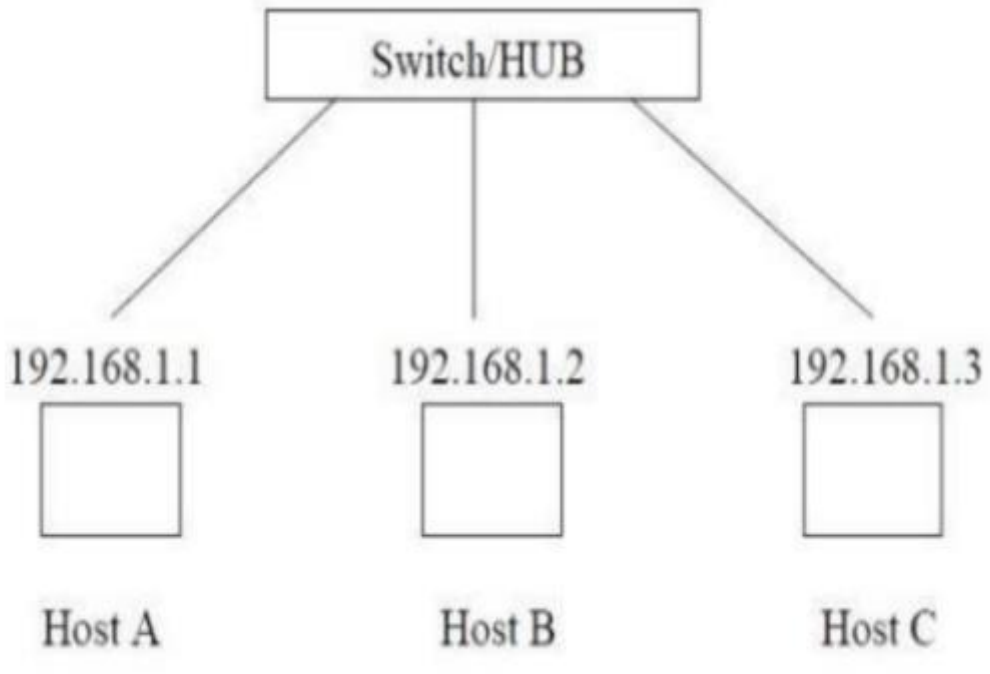
*Ping Another PC:*
Use the "ping" command to test connectivity to another PC. For example: `ping 192.168.1.3`
This command sends a series of packets to the specified IP address and checks for a response, verifying that the devices can communicate within the LAN.

**5. Saving the Packet Tracer Project**

After completing the configuration and testing, it's crucial to save your Packet Tracer project to preserve your work. Follow these steps to save your project:

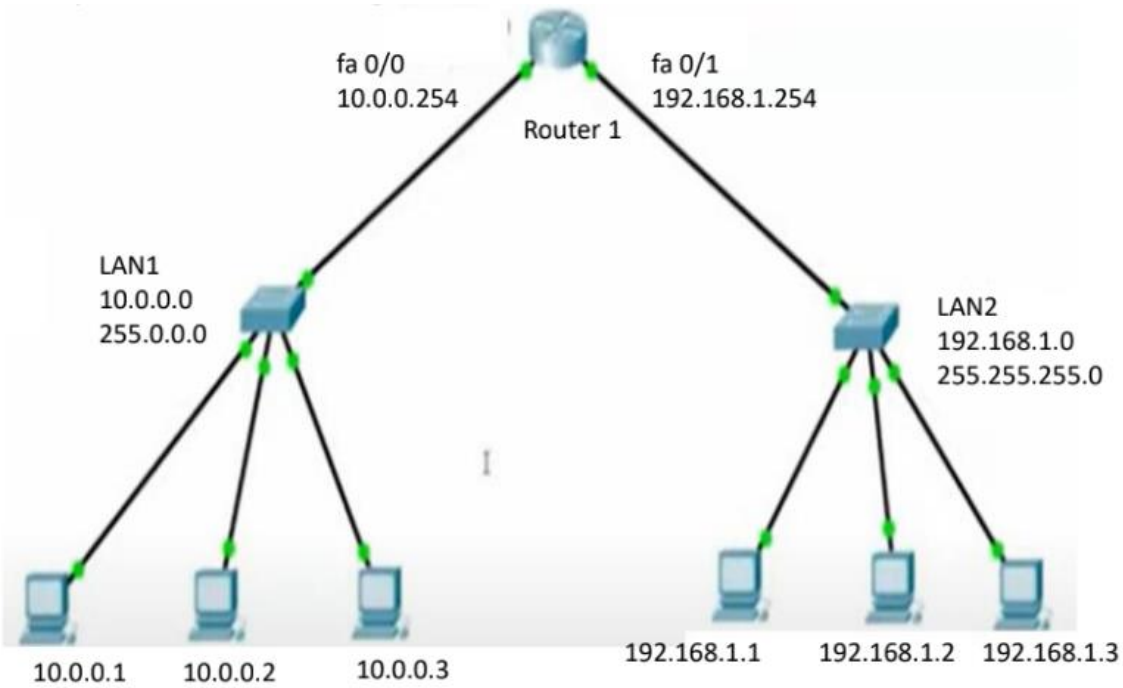
*Save the Project:*
Click on “File” and then select “Save” to save your Packet Tracer project.

## Task-3: Establish a simple LAN using Switch.




## Task-4: Establish Peer to Peer network connection using two systems using Switch and Router in a LAN.

**Step 1:** Create the following network in Packet Tracer



## 2. Configure a Cisco Router

**Step 1: Access Command Line Interface**
– Access the CLI (Command Line Interface) by clicking on Router 1.
– Say no when asked if you would like to enter the initial configuration dialog of the router.
– **Good To Remember:**

- There are various command modes, and the following are some of the main command modes:
    1. User EXEC Mode: `Router>`
    2. Privileged EXEC Mode: `Router#`
    3. Global Configuration Mode: `Router(config)#`
    4. Interface Configuration Mode: `Router(config-if)#`
    5. Sub Interface Configuration Mode: `Router(config-subif)#`

**Step 2: Enter the Privileged EXEC Mode**
– To enter the Privileged EXEC Mode, use the command `enable` in the CLI of Router 1.

- `Router>enable`
- `Router#`

**Step 3: Switch to Global Configuration Mode**
– Switch to the Global Configuration Mode by using the command `configure terminal` in the Privileged EXEC Mode.

- `Router#configure terminal`
- `Enter configuration commands, one per line. End with CNTL/Z.`
- `Router(config)#`

**Step 4: Change the Router Name**
– Change the default router name from 'Router' to 'R1'.

- You can configure any desired name on the router as per the topology you're creating. This helps you to differentiate the device from other devices in the network.
- The command to change the name is `hostname name`.
- `Router(config)#hostname R1`

**Step 5: Assign IP Addresses**
– Assign IP Addresses to the interfaces of Router 1.

- Assigning IP addresses to the router is very important and is required to make your router able to forward packets to/from networks.
- Configure `192.168.1.254` and `10.0.0.254` on interface `fa 0/1` and `fa 0/0`, respectively.
- **Commands to configure IP Address and make the interface `fa 0/1` up:**
    - `%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up`
- **Commands to configure IP Address and make the interface `fa 0/0` up:**
    - `%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up`

- After assigning the IP Addresses and turning on the ports, you'll observe the link lights turning green. This indicates that the port is now in forwarding mode and can be used to transmit data.

**Step 6: Assign IP Address to All PCs**
– Click on PC 0 and open the Desktop Tab.
– Go to the IP Configuration and assign IP Address `10.0.0.1` with the default subnet mask `255.0.0.0` and gateway `10.0.0.254` on PC 0.
– Assign IP addresses to other PCs similarly.

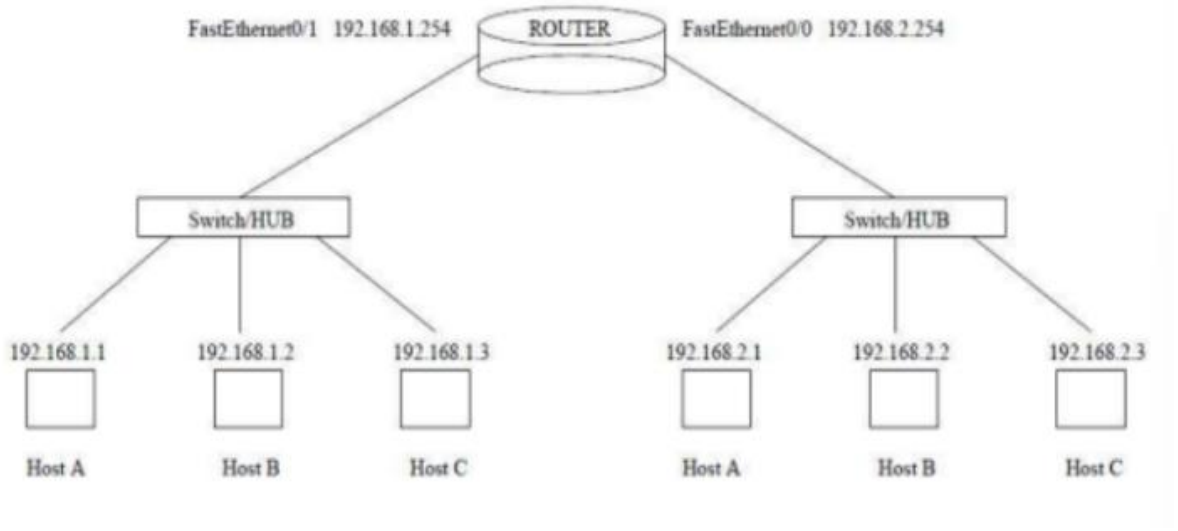
**Step 7: Check Connectivity**
– Check the connectivity between devices.

- We will now ping from PC 0 to PC 1 to check if they can communicate.
- Click on PC 0, go to the Desktop Tab, and click on Command Prompt.
- Use the command `ping IP Address of Destination`.
    - `Packet Tracer PC Command Line 1.0`
    - `C:\>ping 192.168.1.2`
    - `Pinging 192.168.1.2 with 32 bytes of data:`
    - `Request timed out.`
    - `Request timed out.`
    - `Reply from 192.168.1.2: bytes=32 time<1ms TTL=126`
    - `Reply from 192.168.1.2: bytes=32 time<1ms TTL=126`
    - `Ping statistics for 192.168.1.2:`
    - `Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),`
    - `Approximate round trip times in milliseconds:`
    - `Minimum = 0ms, Maximum = 0ms, Average = 0ms`

**Step 8: Save the Packet Tracer Project**
– Save the Packet Tracer project to preserve your work.

- Click on “File” and then select “Save” to save your Packet Tracer project.

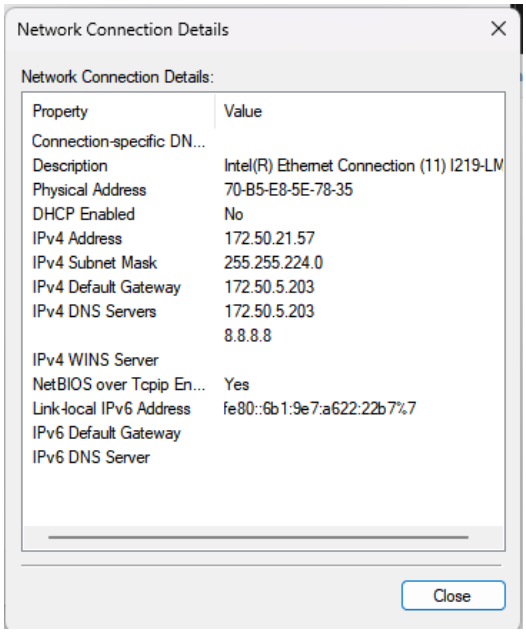
## Task-5: Establish two LANs via Router

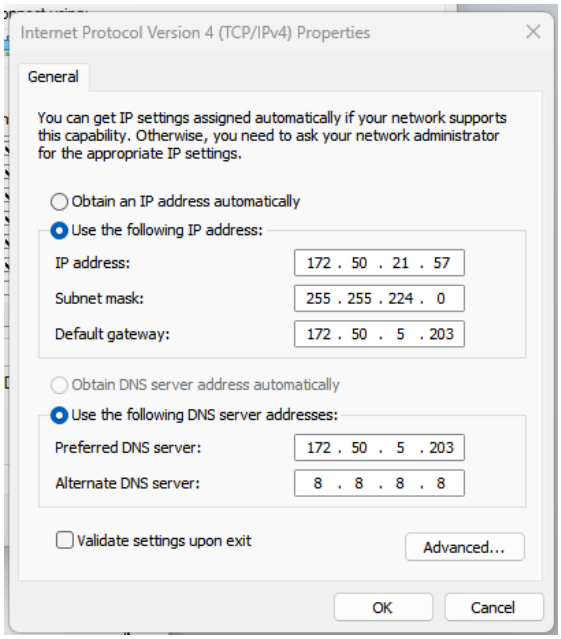


## Output Description:

### Task – 1:

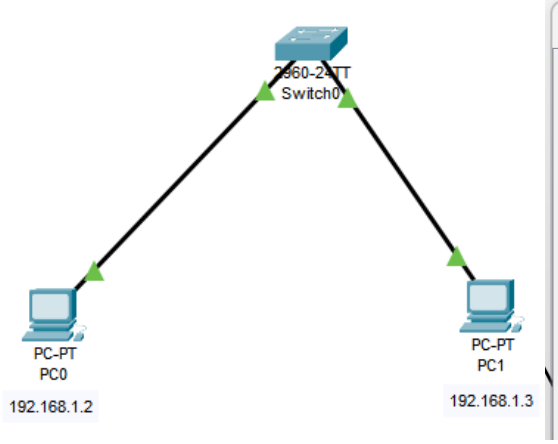
The images depict the configuration of a static IP address, subnet mask, and default gateway for a system in a LAN. The IP address is set to `172.50.21.57` with a subnet mask of `255.255.224.0` and a default gateway of `172.50.5.203`. DNS settings are manually configured with a preferred DNS server of `172.50.5.203` and an alternate DNS server of `8.8.8.8`.

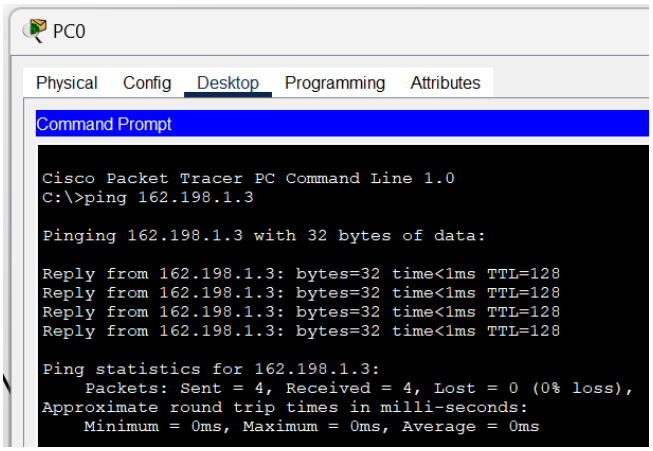



### Task – 2:

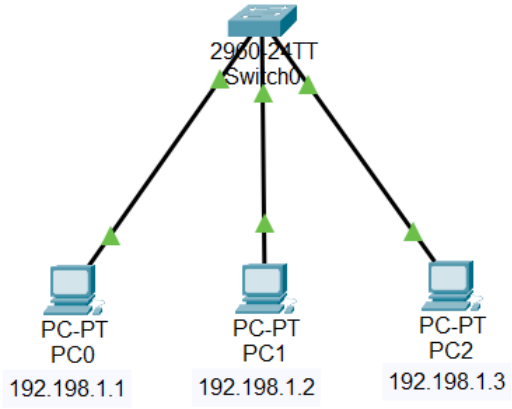
I opened Cisco Packet Tracer, created a new project, added PCs and a switch, connected them with copper cables, manually assigned unique IP addresses (e.g., PC1: 192.168.1.2, PC2: 192.168.1.3), and verified connectivity using the "ping" command.



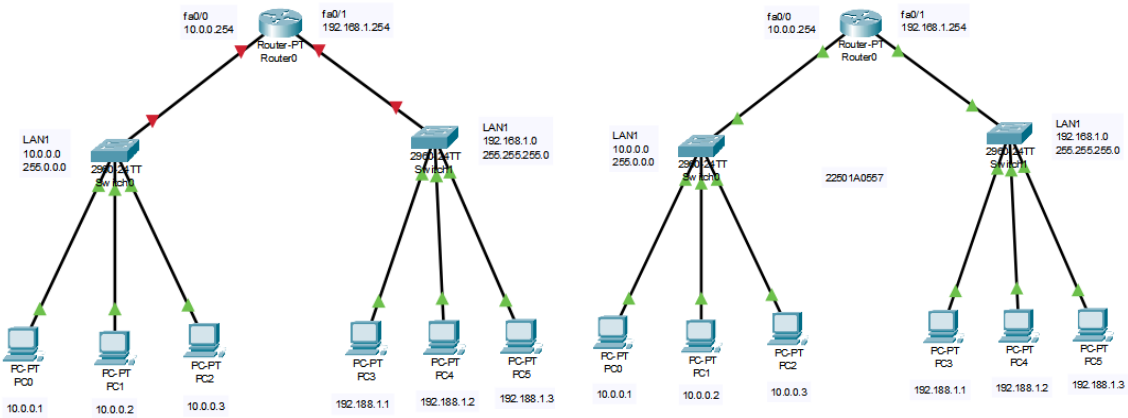

### Task – 3:

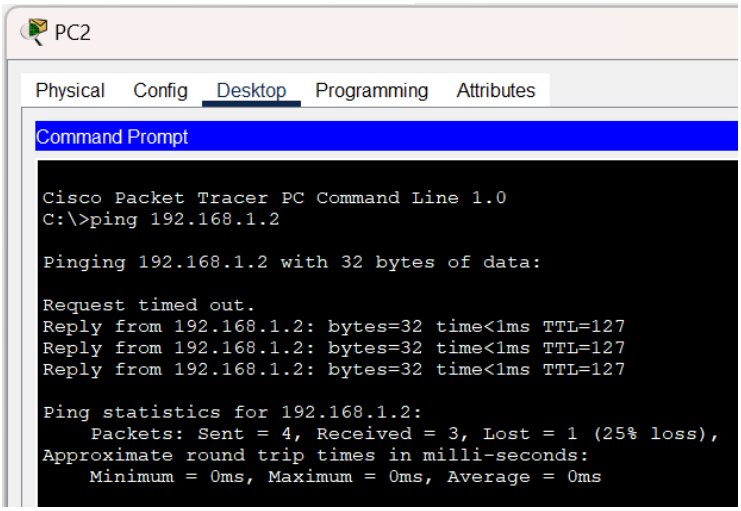
A simple LAN using Switch is established.




**Task – 4:**

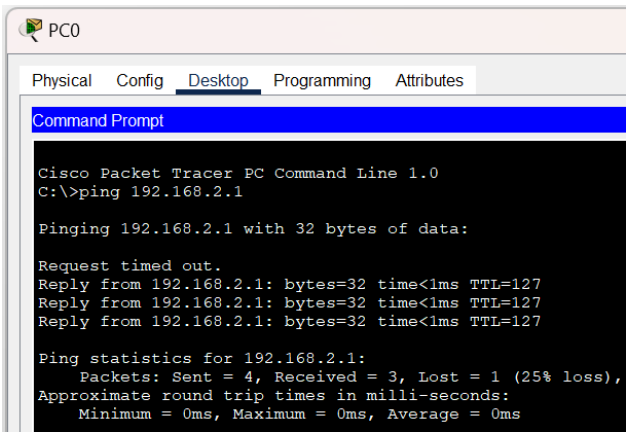
I successfully established a peer-to-peer network connection using a switch and a Cisco router in a LAN. Initially, the link lights were red (indicating disabled interfaces), turned yellow (interfaces up but not fully operational), and finally green (successful communication). I configured the router, assigned IP addresses, checked connectivity, and saved the Packet Tracer project.

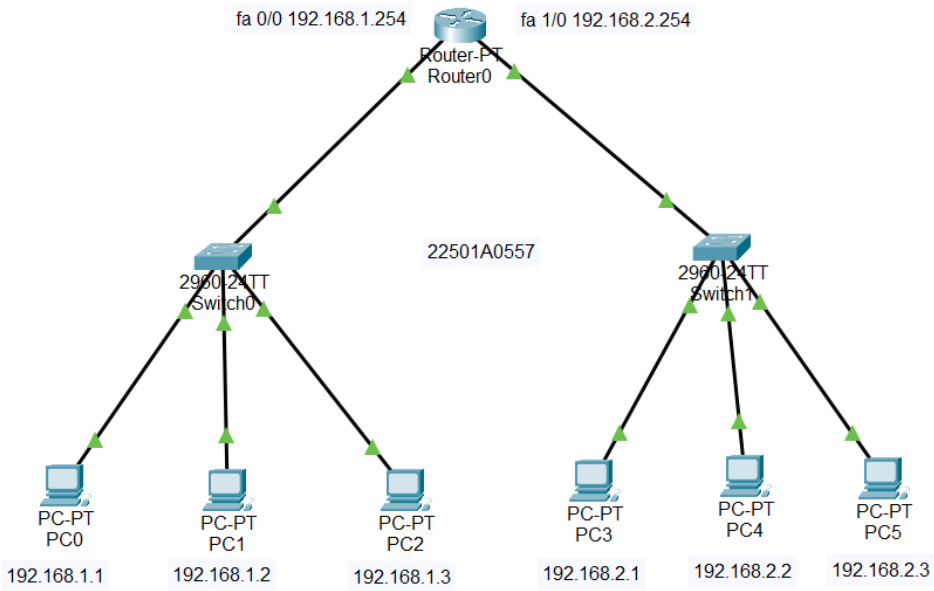


**Task – 5:**

I successfully set up a peer-to-peer network connection using a switch and a Cisco router in a LAN. I observed that initially, the link lights were red, indicating disabled interfaces. After configuring the router and assigning IP addresses, the lights turned yellow (interfaces up but not fully operational). Finally, they turned green, signifying successful communication.




## Viva-Voice Questions:

**Q) What command is used to enter Privileged EXEC Mode on a Cisco router?**
**A)** The command `enable` is used to enter Privileged EXEC Mode.

**Q) How do you access Global Configuration Mode from Privileged EXEC Mode?**
**A)** Use the command `configure terminal` to access Global Configuration Mode**.**

**Q) What command is used to change the router's hostname?**
**A)** Use the command `hostname` followed by the desired name to change the router's hostname.

**Q) How do you assign an IP address to an interface on a Cisco router?**
**A)** Use the command `ip address` followed by the IP address and subnet mask to assign an IP address to an interface.

**Q) What is the purpose of the `no shutdown` command in interface configuration?**
**A)** The `no shutdown` command is used to enable an interface, ensuring it is operational.

Date:

# Experiment No. 8

## Aim:

To configure static IP Addresses.

To configure Automatic IP Addressing using DHCP.

Test Connectivity using ICMP.

## Components and Tools:

| | |
|---|---|
| System | : Desktop Computer/Laptop |
| Operating system | : Windows/Linux |
| Tool | : Packet Tracer |
| Components | : PCs (4), Switches (2), Routers (2) |

## Description:

**Static Routing**:

- **What It Is**: A network administrator manually sets up specific paths (routes) in a router's routing table. These paths don't change unless the administrator manually updates them.
- **Another Name**: Non-adaptive Routing
- **How It Works**: The routes are fixed, and if there's a problem with a route, it doesn't adjust automatically.

**Dynamic Routing**:

- **What It Is**: The router automatically updates its routes based on the current network conditions. It uses routing protocols to find the best paths to different destinations.
- **Examples**: RIP and OSPF are popular dynamic routing protocols.
- **How It Works**: If a route fails, dynamic routing protocols quickly find a new path to the destination.

**Dynamic IP Addressing**:

- **What It Is**: A system that assigns IP addresses to devices automatically.
- **Protocol Used**: Dynamic Host Configuration Protocol (DHCP).
- **Components**:
  - **DHCP Server**: Assigns IP addresses to devices.
  - **DHCP Clients**: Devices that receive IP addresses from the DHCP server.

**Another Name for Dynamic Routing**: Adaptive Routing

## Procedure:

### Part 1: Configure Static IP Addresses

1. **Deploy the Components**:
   - Create the network as shown in the provided figure.
2. **Assign Static IP Addresses to Each PC**:
   - Set a unique static IP address for each PC in the network.
3. **Configure Both Routers**:
   - Access the configuration settings for each router.
4. **Configure Static Routing on Router 0**: a. Click on Router 0. b. Click on the **Config** tab. c. Select **Static** from the routing options. d. Enter the network IP address: 192.168.3.0 (destination network address). e. Enter the subnet mask: 255.255.255.0. f. Enter the Next Hop IP address: 192.168.1.4. g. Click **Add**. h. Go to **Settings** and click **Save**.
5. **Configure Static Routing on Router 1**:
   - Repeat the steps for Router 0, adjusting settings as necessary for Router 1.
6. **Test Connectivity**:
   - Use the **ping** command to verify connectivity between PC1 and PC3.

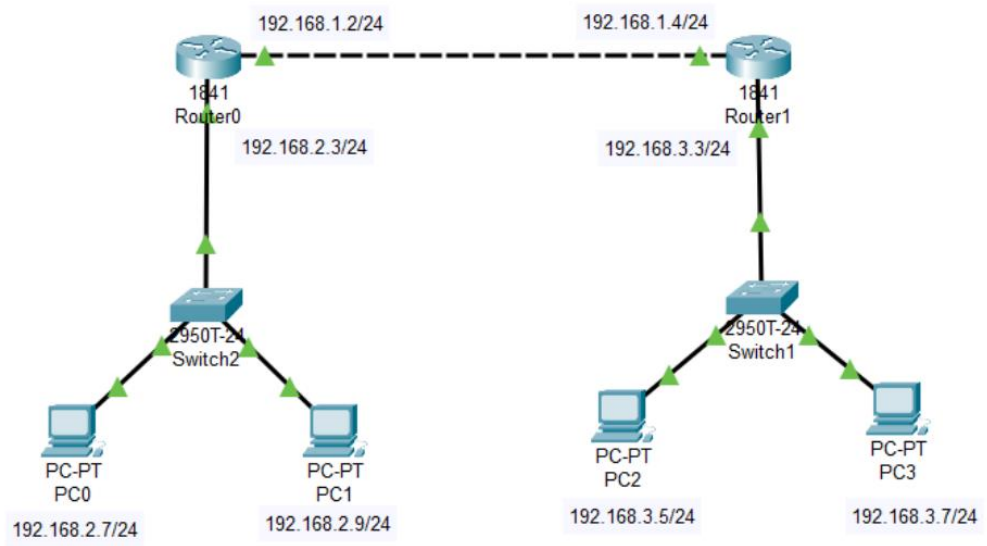



**Fig-8.1:** Configuring static IP addresses

**Part 2: Configure Automatic IP Addressing Using DHCP**

1. **Build the Network Topology**:
   - Set up the network in Packet Tracer as shown in the provided figure.
2. **Configure Static IP Address on the Server**:
   - Set the server's IP address to 192.168.1.2/24.
3. **Configure DHCP Service on the Generic Server**: a. Click on the server. b. Go to the **Services** tab. c. Select **DHCP** from the menu. d. Define the DHCP network parameters as follows:
   - **Pool Name**: MY_LAN
   - **Default Gateway**: 192.168.1.1
   - **DNS Server**: 192.168.1.2
   - **Start IP Address**: 192.168.1.0
   - **Subnet Mask**: 255.255.255.0
   - **Maximum Number of Users**: 256 e. Click **Add** and then **Save**. The DHCP entry will be included in the list.
4. **Enable DHCP Service**:
   - Turn on the DHCP service on the server.
5. **Configure PCs for DHCP**:
   - Enable DHCP configuration on each PC to allow them to receive automatic IP addresses.

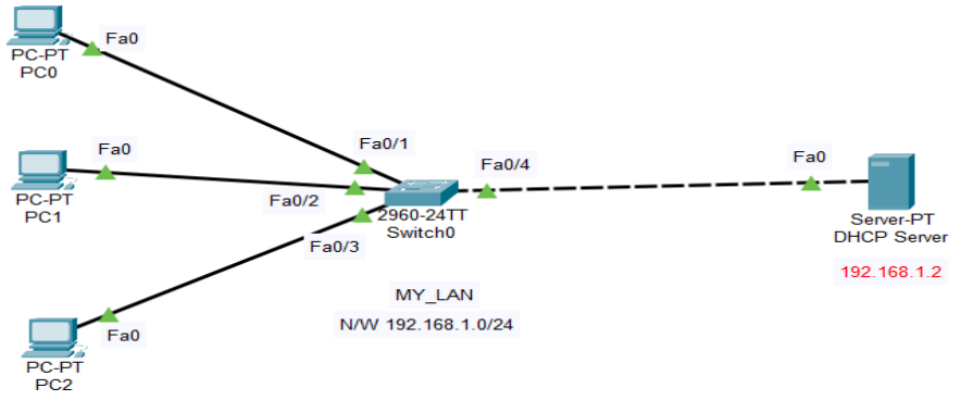



**Fig-8.2:** Configuring Dynamic IP addresses

## Observation:

```
C:\>ping 192.168.3.7

Pinging 192.168.3.7 with 32 bytes of data:

Reply from 192.168.3.7: bytes=32 time<1ms TTL=126
Reply from 192.168.3.7: bytes=32 time<1ms TTL=126
Reply from 192.168.3.7: bytes=32 time<1ms TTL=126
Reply from 192.168.3.7: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.3.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

**Fig-8.3.1:** Output of ping command for task-1

The image shows the result of a successful ping to the IP address `192.168.3.7`. Four packets were sent, and all were received with no loss, showing that the connection is stable. The round-trip time was very low, averaging 0 milliseconds, which means the network connection to `192.168.3.7` is excellent.

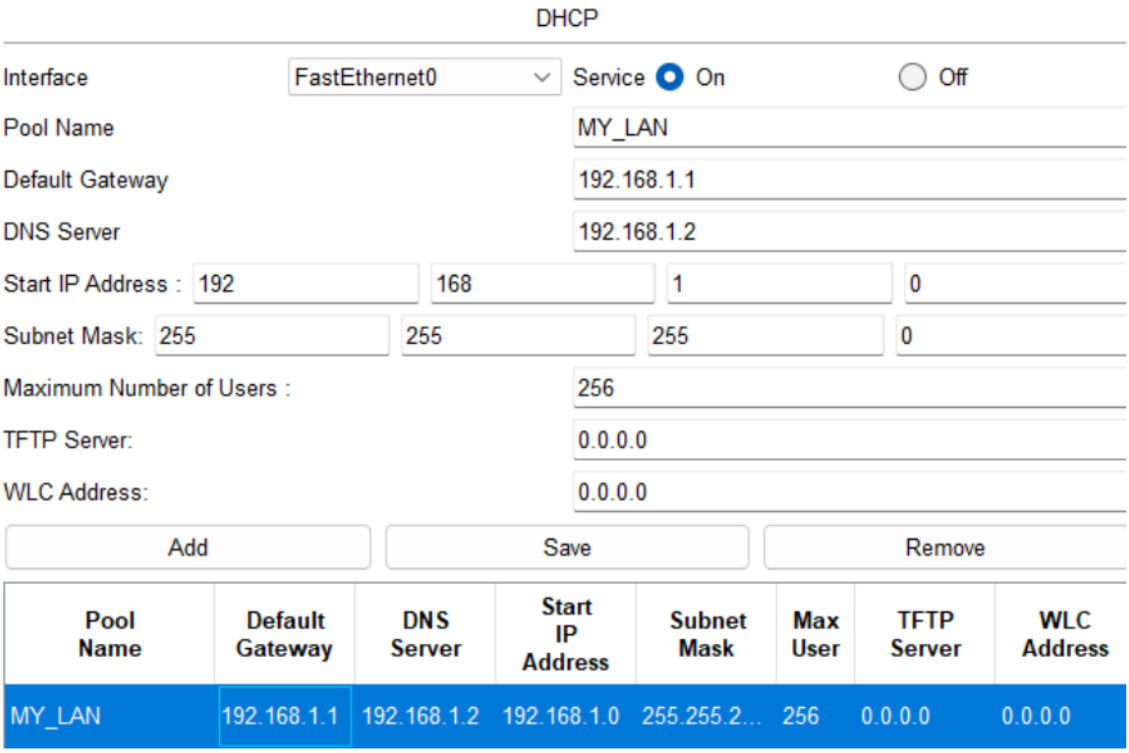



**Fig-8.3.2:** Configuring DHCP server

**Result:**

The configuration of static IP addresses, automatic IP addresses using DHCP, and connectivity testing with ICMP has been completed successfully.

**Viva-Voice Questions:**

**Q) What is a static IP address?**

**A)** A static IP address is a fixed IP address manually assigned to a device, which does not change over time.

**Q) How do you configure a static IP address on a device?**

**A)** You configure a static IP address by manually entering the IP address, subnet mask, and default gateway in the network settings of the device.

**Q) What does DHCP stand for and what is its role?**

**A)** DHCP stands for Dynamic Host Configuration Protocol, and it automatically assigns IP addresses to devices on a network.

**Q) What is the purpose of the ping command in network testing?**

**A)** The ping command tests network connectivity by sending packets to a destination IP address and measuring the response time.

**Q) How can you verify that a router's static routing is set up correctly?**

**A)** You can verify static routing by checking the routing table and using the ping command to test connectivity to the destination network.

Date:

# Experiment No. 9

## Aim:

Implement Unicast Routing Algorithms using any Programming Language

## Components and Tools:

| | |
|---|---|
| System | : Desktop Computer/Laptop |
| Operating System | : Windows/Linux |
| IDE | : Spyder |
| Language | : Python |

## Description:

**Unicast Routing**:

Unicast routing is a basic networking concept where data packets travel from one specific source to one specific destination. The goal is to find the fastest and most reliable route for the data to reach its target. Routers use different methods to decide the best path for these packets based on the network's layout and current conditions.

**Distance Vector Routing**:

Distance Vector Routing is a type of routing protocol where each router keeps a table showing the best-known distances to every destination and the next hop to get there. The idea is to ensure that packets take the shortest or least expensive route to their destination. This method helps routers make decisions on the best path for sending data.

**Dijkstra's Algorithm**:

Dijkstra's Algorithm is used to find the shortest path from one node to all other nodes in a network. It's commonly used in link-state routing protocols, where routers have complete information about the network. By running this algorithm, routers can determine the best route to send packets efficiently. Like Distance Vector Routing, Dijkstra's Algorithm helps in finding the most optimal path for unicast communication.

## Algorithm: (Distance Vector Routing)

- Initialize **Data Structures**:

  - Create a distance matrix `dist` initialized to `INF` (infinity) to represent the initial distances between routers.
  - Create a `next_hop` matrix initialized to `-1` to track the next hop for each path.

- Set **Up Initial Values**:

  - For each router, set the distance to itself as `0`.
  - For each direct connection (edge) between routers, set the distance in the `dist` matrix and update the `next_hop` to point directly to the destination.

- Update **Distances**:

  - Repeat until no more updates occur:
    - For each router `i`:
      - For each router `j`:
        - For each router `k`:
          - If a shorter path from `i` to `k` through `j` is found (i.e., `dist[i][k]` can be reduced):
            - Update the distance `dist[i][k]`.
            - Update the `next_hop` for the path from `i` to `k` to be the same as the path from `i` to `j`.

- Print **Routing Tables**:

  - For each router, print the routing table that shows the shortest distance and next hop to every other router.

- Handle **Input/Output**:

  - Read the number of routers and edges.
  - Read each edge's details and update the `graph` with these edges.
  - Call the `distance_vector_routing` function to compute the shortest paths and print the results.

**Program:**

```
import sys


INF = float('inf')


def print_routing_table(dist, next_hop, n, src):
    print(f"\nRouting table for router {chr(ord('A') +
src)}:")
    print("Destination\tDistance\tNext Hop")
    for i in range(n):
        destination = chr(ord('A') + i)
        hop = destination if next_hop[i] == -1 else
chr(ord('A') + next_hop[i])
        if dist[i] == INF:
            print(f"{destination}\t\tinf\t\t-")
        else:
            print(f"{destination}\t\t{dist[i]}\t\t{hop}")


def distance_vector_routing(graph, n):
    dist = [[INF] * n for _ in range(n)]
    next_hop = [[-1] * n for _ in range(n)]

    for i in range(n):
        dist[i][i] = 0
        for j in range(n):
            if graph[i][j] != INF:
                dist[i][j] = graph[i][j]
                next_hop[i][j] = j

    updated = True
    while updated:
```

```
        updated = False
        for i in range(n):
            for j in range(n):
                for k in range(n):
                    if dist[i][j] < INF and dist[j][k] < INF
and dist[i][k] > dist[i][j] + dist[j][k]:
                        dist[i][k] = dist[i][j] + dist[j][k]
                        next_hop[i][k] = next_hop[i][j]
                        updated = True

    for i in range(n):
        print_routing_table(dist[i], next_hop[i], n, i)

if __name__ == "__main__":
    n = int(input("Enter the number of routers: "))
    graph = [[INF] * n for _ in range(n)]

    for i in range(n):
        graph[i][i] = 0

    edges = int(input("Enter the number of edges: "))
    for _ in range(edges):
        srcChar, destChar, weight = input("Enter edge (source
destination weight): ").split()
        weight = int(weight)
        src = ord(srcChar) - ord('A')
        dest = ord(destChar) - ord('A')
        graph[src][dest] = weight
        graph[dest][src] = weight

    distance_vector_routing(graph, n)
```

**Output:**

```
G VENKATA SAI RAM@GVSR-Laptop MINGW64 /d/OneDrive/La
$ python -u "d:\OneDrive\Language Specific Projects'
Enter the number of routers: 4
Enter the number of edges: 5
Enter edge (source destination weight): A B 2
Enter edge (source destination weight): B C 3
Enter edge (source destination weight): C D 11
Enter edge (source destination weight): A D 1
Enter edge (source destination weight): B D 7

Routing table for router A:
Destination     Distance        Next Hop
A               0               A
B               2               B
C               5               B
D               1               D

Routing table for router B:
Destination     Distance        Next Hop
A               2               A
B               0               B
C               3               C
D               3               A

Routing table for router C:
Destination     Distance        Next Hop
A               5               B
B               3               B
C               0               C
D               6               B

Routing table for router D:
Destination     Distance        Next Hop
A               1               A
B               3               A
C               6               A
D               0               D
```

The output displays the routing tables generated for four routers (A, B, C, and D) in a network using the Distance Vector Routing algorithm implemented in our program. Each table shows the shortest distance (or cost) and the next hop required for a given router to reach every other router in the network.

- **Router A** has direct connections to **B** with a distance of 2 and to **D** with a distance of 1. The path to **C** is indirect, routed through **B** with a total distance of 5.
- **Router B** shows its direct connections to routers **A** and **C** and calculates the shortest paths to other routers, using the next hop to indicate the optimal route.
- **Router C** maintains direct connections to **B** and **D**; the path to **A** is routed through **B** based on the shortest calculated distance.
- **Router D** has a direct link to **A** and uses intermediate routes through **B** or **C** to reach other routers.

These routing tables are the result of our program's execution of the Distance Vector Routing algorithm, which optimally computes the shortest paths between all routers in the network.

## Algorithm: (Dijkstra's)

• Initialize:

- Set the number of vertices `V`.
- Create a `graph` to store edge weights between the vertices.
- Set the source vertex `src` from which to calculate shortest paths.

• Set **up the distance and visited lists**:

- Initialize the `dist` list with infinity (`sys.maxsize`) to represent the initial distances from the source to all other vertices.
- Set the distance from the source to itself as 0.
- Initialize the `visited` list to keep track of visited vertices, all initially set to `False`.

• Find **the nearest unvisited vertex**:

- Use the `min_distance` function to find the vertex `u` with the smallest distance from the source that has not been visited yet.

• Update **distances**:

- Mark vertex `u` as visited.
- For each vertex `v`, if there is an edge from `u` to `v` and `v` is unvisited, and if the total distance to reach `v` through `u` is less than the current known distance to `v`, update the distance to `v`.

• Repeat:

- Repeat steps 3 and 4 until all vertices have been visited.

• Output **the shortest distances**:

- Print the shortest distances from the source to all vertices.

**Program:**

```
import sys


def min_distance(dist, visited, V):
    # Initialize minimum distance for next node
    min_val = sys.maxsize
    min_index = -1

    # Search for the nearest unvisited vertex
    for v in range(V):
        if not visited[v] and dist[v] <= min_val:
            min_val = dist[v]
            min_index = v

    return min_index


def dijkstra(graph, src, V):
    # Initialize distance values and visited set
    dist = [sys.maxsize] * V
    visited = [False] * V
    dist[src] = 0

    for _ in range(V - 1):
        # Find the nearest unvisited vertex
        u = min_distance(dist, visited, V)
```

```
        visited[u] = True


        # Update the distance value of the adjacent vertices

        for v in range(V):

            if not visited[v] and graph[u][v] and dist[u] !=
sys.maxsize and dist[u] + graph[u][v] < dist[v]:

                dist[v] = dist[u] + graph[u][v]


    # Print the result

    print("Vertex \t Distance from Source")

    for i in range(V):

        print(f"{i} \t\t {dist[i]}")


if __name__ == "__main__":

    V = int(input("Enter the number of vertices: "))

    graph = [[0] * V for _ in range(V)]


    E = int(input("Enter the number of edges: "))

    for _ in range(E):

        print("Enter the directed edges (source destination
weight):",end=" ")

        src, dest, weight = map(int, input().split())

        graph[src][dest] = weight


    src = int(input("Enter the source vertex: "))

    dijkstra(graph, src, V)
```

## Output:

```
G VENKATA SAI RAM@GVSR-Laptop MINGW64 /d/OneDrive/Language Spe
$ python -u "d:\OneDrive\Language Specific Projects\03 Python\
Enter the number of vertices: 5
Enter the number of edges: 7
Enter the directed edges (source destination weight): 0 1 3
Enter the directed edges (source destination weight): 0 3 8
Enter the directed edges (source destination weight): 0 4 7
Enter the directed edges (source destination weight): 3 4 3
Enter the directed edges (source destination weight): 1 3 4
Enter the directed edges (source destination weight): 1 2 1
Enter the directed edges (source destination weight): 2 3 2
Enter the source vertex: 0
Vertex   Distance from Source
0                0
1                3
2                4
3                6
4                7
```

The output displayed is the result of running Dijkstra's algorithm on a graph with 5 vertices (numbered from 0 to 4) and 7 edges. The user provided the directed edges along with their weights, and the algorithm calculated the shortest distances from the chosen source vertex (vertex 0) to all other vertices in the graph.

The resulting table shows each vertex and its corresponding shortest distance from the source vertex 0. According to the output:

- Vertex 0 has a distance of 0 (since it's the source vertex).
- Vertex 1 has a shortest distance of 3 from vertex 0.
- Vertex 2 has a shortest distance of 4 from vertex 0.
- Vertex 3 has a shortest distance of 6 from vertex 0.
- Vertex 4 has a shortest distance of 7 from vertex 0.

These values represent the minimum cost required to reach each vertex from the source vertex 0, as computed by Dijkstra's algorithm based on the given edge weights.

## Result:

Unicast routing algorithms, Distance vector routing algorithm and Dijkstra algorithm are implemented and executed successfully.

## Viva-Voce Questions:

**Q) What is the primary difference between Dijkstra's algorithm and the Distance Vector Routing algorithm?**
A) Dijkstra's algorithm finds the shortest path from a single source to all other nodes, while Distance Vector Routing updates paths between routers in a distributed network to find the shortest path.

**Q) How does Dijkstra's algorithm select the next vertex to process?**
A) Dijkstra's algorithm selects the vertex with the minimum distance from the source that has not yet been visited.

**Q) In Distance Vector Routing, what is meant by "distance vector"?**
A) A "distance vector" is a table that each router maintains, which contains the shortest distance to every other router in the network and the next hop to reach them.

**Q) Why is Dijkstra's algorithm suitable for link-state routing protocols?**
A) Because Dijkstra's algorithm requires complete knowledge of the network topology to calculate the shortest path, which is provided by link-state protocols.

**Q) What happens when a router detects a change in a network topology using Distance Vector Routing?**
A) The router updates its distance vector table and shares the new information with its neighbours to propagate the changes throughout the network.

Date:

# Experiment No. 10

## Aim:

Demonstrate Network Address Translation (NAT)

## Components and Tools:

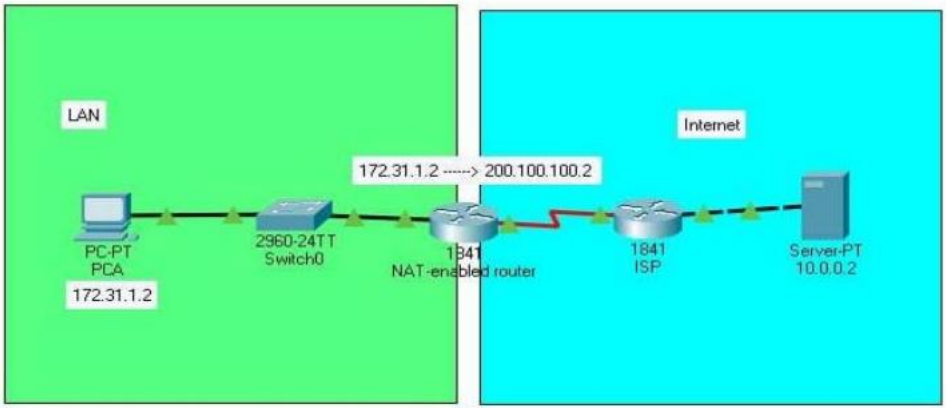
| | |
|---|---|
| System | : Desktop Computer/Laptop |
| Operating system | : Windows/Linux |
| Tool | : Packet Tracer |
| Components | : Laptops (3), Switch (1), Routers (2), Server (1) |

## Description:

### IP Address Classification

### Internet Protocol Versions

- **Internet Protocol version 4 (IPv4)**: Defines an IP address as a 32-bit number.
- **Internet Protocol version 6 (IPv6)**: Defines an IP address as a 128-bit number.




### Public and Private IP Addresses

- **Public Addresses**: Routable addresses used on the internet, allowing users to access resources on a computer network located anywhere in the world.
- **Private Addresses**: Not routable, and no traffic can be sent to or from them over the internet. These addresses are within the range of:
  - 10.0.0.0 to 10.255.255.255
  - 172.16.0.0 to 172.31.255.255
  - 192.168.0.0 to 192.168.255.255

**IPv4 Address Exhaustion**

- **Limitation**: IPv4 faces an address exhaustion issue.
- **Short-term Solutions**: Private addressing and Network Address Translation (NAT).

**Network Address Translation (NAT)**

- **Purpose**: Allows organizations to assign private addresses to their internal network while translating them to public addresses for internet connectivity.
- **Types of NAT**:
  - **Static Network Translation (Static NAT)**
  - **Dynamic Network Address Translation (Dynamic NAT)**
  - **Port Address Translation (PAT)**

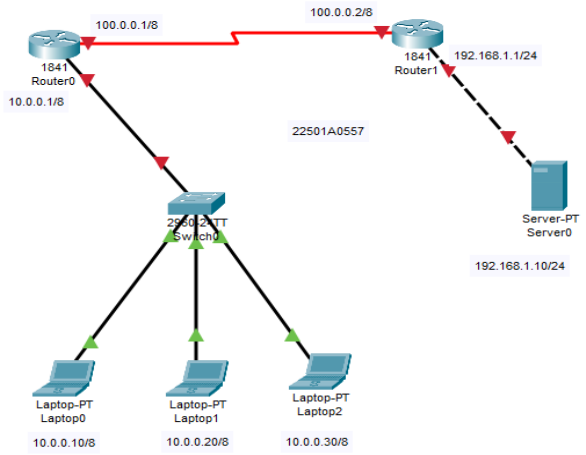
**Static NAT Configuration**

**Diagrams / Experimental Set-up**

- **Static NAT**: Creates a one-to-one mapping between private and public addresses, usually configured by a network administrator.

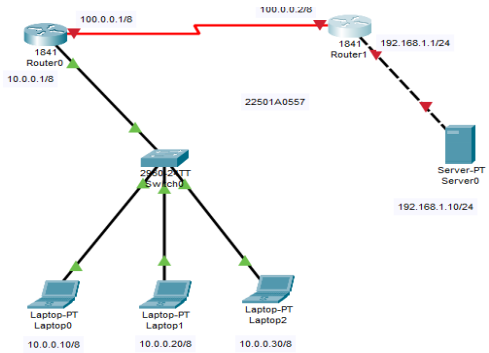
## Procedure:

1. **Initial IP Configuration**:
   - Assign IP addresses to devices as per the table below:
     - **Laptop0**: 10.0.0.10/8 (Connected to Fa0/0 of R0)
     - **Laptop1**: 10.0.0.20/8 (Connected to Fa0/0 of R0)
     - **Laptop2**: 10.0.0.30/8 (Connected to Fa0/0 of R0)
     - **Server0**: 192.168.1.10/24 (Connected to Fa0/0 of R1)
     - **Serial 0/0/0 of R1**: 100.0.0.1/8 (Connected to Serial 0/0/0 of R2)
     - **Serial 0/0/0 of R2**: 100.0.0.2/8 (Connected to Serial 0/0/0 of R2)

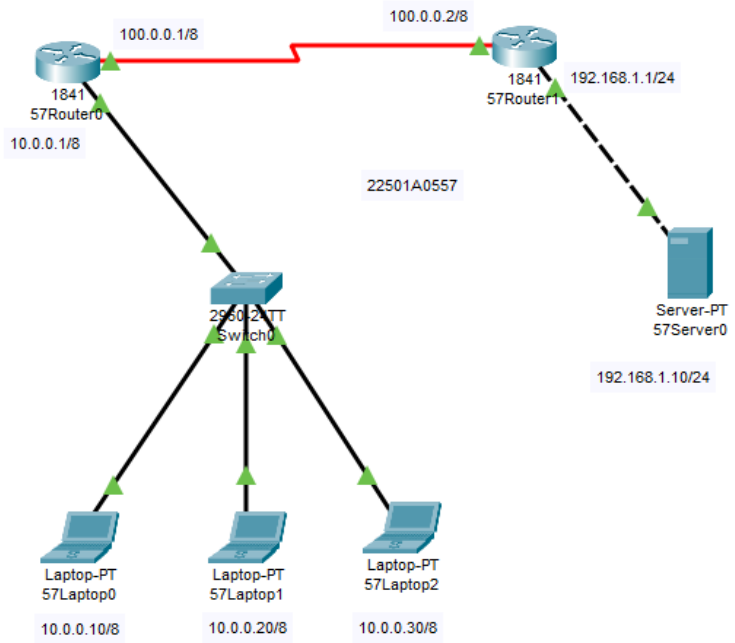



2. **Set IP Address to Laptop**:
   - Click on the laptop, go to Desktop, select IP configuration, choose Static, and set the IP address as given in the table. Repeat for other laptops.

3. **Set IP Address to Server**:
   - Follow the same steps to configure the IP address on the server.




4. **Configure IP Address in Routers**:
   - **Router1**:
     - Enter CLI mode and configure the IP addresses:
     - Router>enable
     - Router# configure terminal
     - Router(config)#hostname R1
     - R1(config)#interface FastEthernet0/0
     - R1(config-if)#ip address 10.0.0.1 255.0.0.0
     - R1(config-if)#no shutdown
     - R1(config-if)#exit
     - R1(config)#interface Serial0/0/0
     - R1(config-if)#ip address 100.0.0.1 255.0.0.0
     - R1(config-if)#clock rate 64000
     - R1(config-if)#bandwidth 64
     - R1(config-if)#no shutdown
     - R1(config-if)#exit
     - R1(config)#



**Good to Remember**

- **Command Modes**:
    1. **User EXEC Mode**: Router>
    2. **Privileged EXEC Mode**: Router#
    3. **Global Configuration Mode**: Router(config)#
    4. **Interface Configuration Mode**: Router(config-if)#
    5. **Sub Interface Configuration Mode**: Router(config-subif)
- **Show Controllers Command**:
- R1#show controllers serial 0/0/0
- Interface Serial0/0/0
- Hardware is PowerQUICC MPC860
- DCE V.35, clock rate 2000000

**Step-2.2: Configure IP Address in Router2**

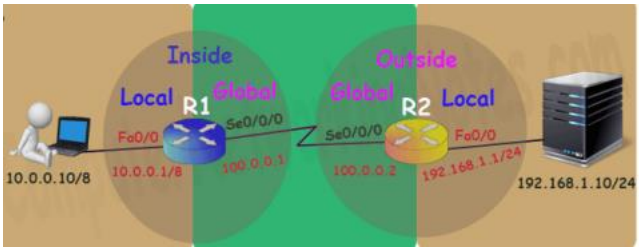
1. Click Router2 and select CLI, then press Enter.
2. Enter the following commands:
3. Router>enable
4. Router#configure terminal
5. Router(config)#hostname R2
6. R2(config)#interface FastEthernet0/0
7. R2(config-if)#ip address 192.168.1.1 255.255.255.0
8. R2(config-if)#no shutdown
9. R2(config-if)#exit
10. R2(config)#interface Serial0/0/0
11. R2(config-if)#ip address 100.0.0.2 255.0.0.0
12. R2(config-if)#no shutdown
13. R2(config-if)#exit
14. R2(config)#

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R2
R2(config)#interface FastEthernet0/0
R2(config-if)#ip address 192.168.1.1 255.255.255.0
R2(config-if)#no shutdown

R2(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
exit
R2(config)#interface Serial0/0/0
R2(config-if)#ip address 100.0.0.2 255.0.0.0
R2(config-if)#no shutdown

R2(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
exit
R2(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up
```

**Step-3: Configure Static NAT**

Static NAT configuration requires three steps:

- Define IP address mapping.
- Define inside local interface.
- Define inside global interface.




**Syntax**:

Router(config)#ip nat inside source static [inside local IP address] [inside global IP address]

**Step-3.1: R1 Static NAT Configuration**

1. Enter the following commands on Router1:
2. Router>enable
3. Router# configure terminal
4. R1(config)#ip nat inside source static 10.0.0.10 50.0.0.10
5. R1(config)#interface FastEthernet 0/0
6. R1(config-if)#ip nat inside
7. R1(config-if)#exit
8. R1(config)#interface Serial 0/0/0
9. R1(config-if)#ip nat outside
10. R1(config-if)#exit

```
R1(config)#ip nat inside source static 10.0.0.10 50.0.0.10
R1(config)#interface FastEthernet0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface Serial0/0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

**Device IP Address Mapping**:

| Device | Inside Local IP Address | Inside Global IP Address |
|---|---|---|
| Laptop0 | 10.0.0.10 | 50.0.0.10 |
| Laptop1 | 10.0.0.20 | 50.0.0.20 |
| Laptop2 | 10.0.0.30 | 50.0.0.30 |
| Server | 192.168.1.10 | 200.0.0.10 |

Configure the translation for the remaining addresses:

R1(config)#ip nat inside source static 10.0.0.20 50.0.0.20
R1(config)#ip nat inside source static 10.0.0.30 50.0.0.30

**Step-3.2: R2 Static NAT Configuration**

1. Enter the following commands on Router2:
2. Router>enable
3. Router# configure terminal

4. R2(config)#ip nat inside source static 192.168.1.10 200.0.0.10
5. R2(config)#interface FastEthernet 0/0
6. R2(config-if)#ip nat inside
7. R2(config-if)#exit
8. R2(config)#interface Serial 0/0/0
9. R2(config-if)#ip nat outside
10. R2(config-if)#exit

**Step-4: Configure Static Routing in Routers**

**Step-4.1: Configure Static Routing in R1**

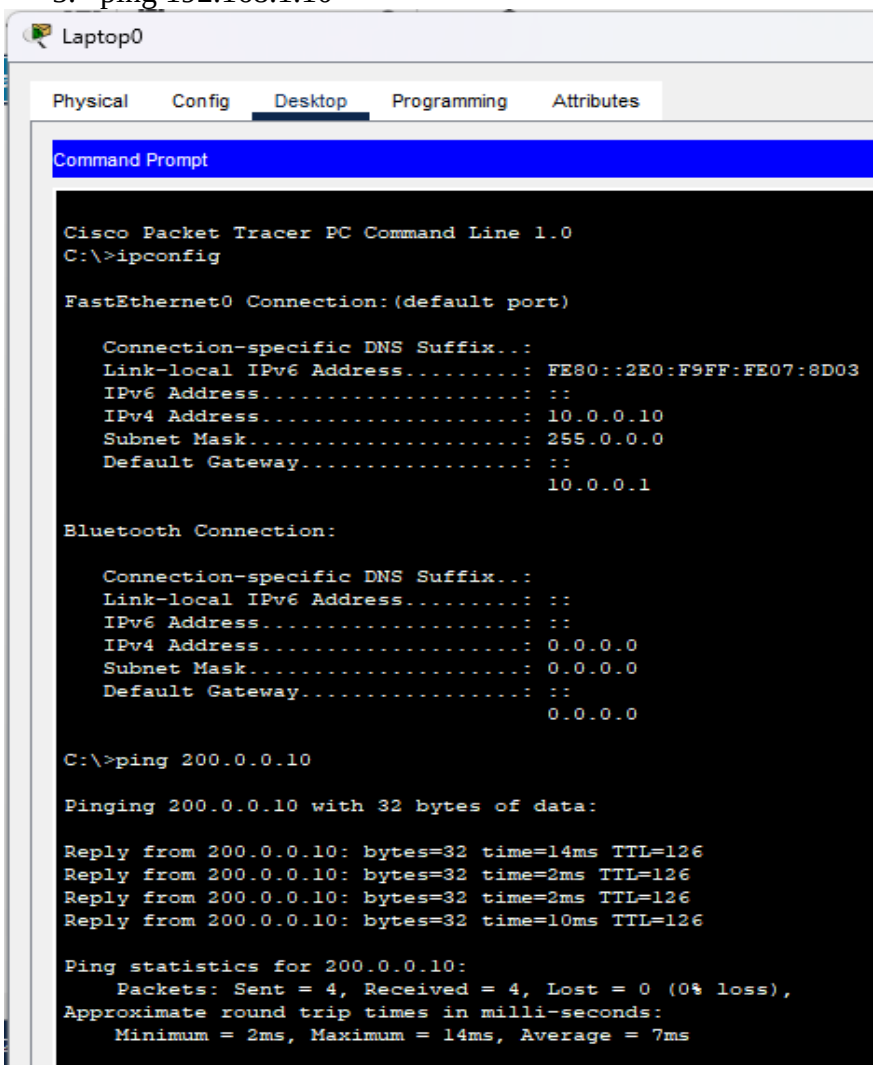
1. Enter the following command on Router1:
2. R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2

**Step-4.2: Configure Static Routing in R2**

1. Enter the following command on Router2:
2. R2(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1

**Step-5: Testing Static NAT Configuration**

1. To test this setup, click Laptop0, go to Desktop, and open Command Prompt.
2. Run the following commands:
3. ipconfig
4. ping 200.0.0.10
5. ping 192.168.1.10

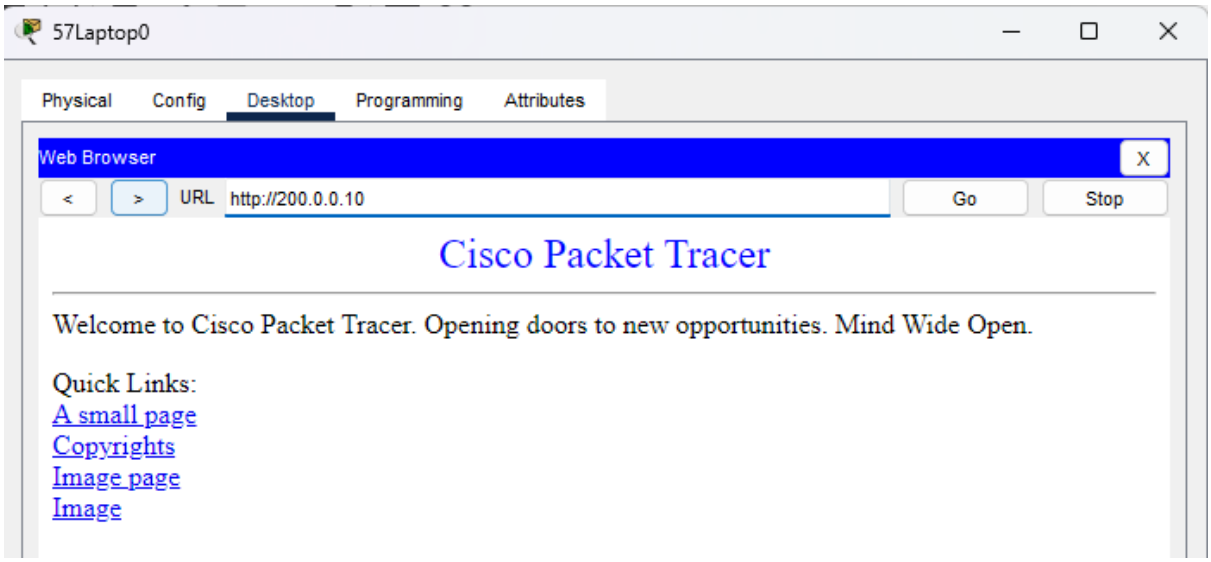


```
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

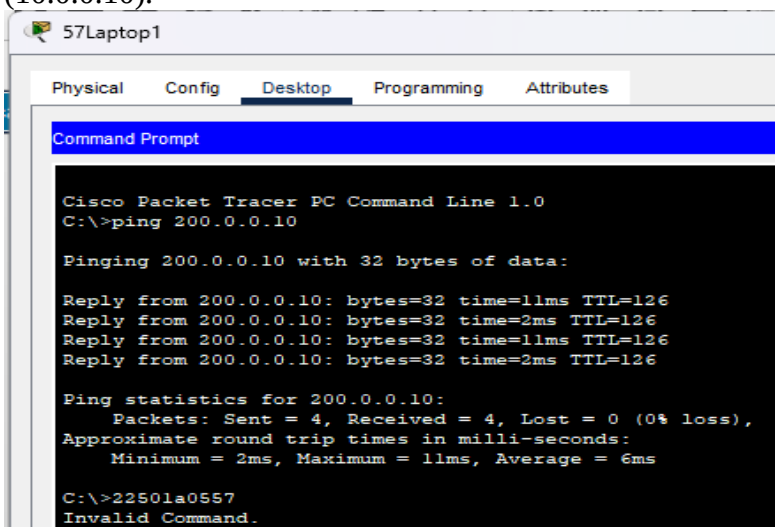
**Explanation**:

- The first command verifies that we are testing from the correct NAT device.
- The second command checks whether we can access the remote device. A ping reply confirms connectivity.
- The third command checks whether we can access the remote device on its actual IP address. A ping error confirms that we cannot connect to the remote device on this IP address.

3. For further testing, click Laptop0, go to Desktop, open Web Browser, and access 200.0.0.10. This confirms that host 10.0.0.10 can access 200.0.0.10.




4. Now, run the ping 200.0.0.10 command from Laptop1. If you cannot connect to the remote device from this host, it is because NAT was configured only for Laptop0 (10.0.0.10).



**Verifying NAT Translation**

1. You can verify this translation on the router with the show ip nat translation command.

**Example**:

R1#show ip nat translation

This command will illustrate the translation on Router R1 and Router R2.

57Router0

```
R1(config)#ip nat inside source static 10.0.0.20 50.0.0.20
R1(config)#ip nat inside source static 10.0.0.30 50.0.0.30
R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 50.0.0.20:2       10.0.0.20:2        200.0.0.10:2       200.0.0.10:2
icmp 50.0.0.20:3       10.0.0.20:3        200.0.0.10:3       200.0.0.10:3
icmp 50.0.0.20:4       10.0.0.20:4        200.0.0.10:4       200.0.0.10:4
icmp 50.0.0.20:5       10.0.0.20:5        200.0.0.10:5       200.0.0.10:5
---  50.0.0.10         10.0.0.10          ---                ---
---  50.0.0.20         10.0.0.20          ---                ---
---  50.0.0.30         10.0.0.30          ---                ---
tcp 50.0.0.10:1025     10.0.0.10:1025     200.0.0.10:80      200.0.0.10:80
tcp 50.0.0.10:1026     10.0.0.10:1026     200.0.0.10:80      200.0.0.10:80

R1#
```

57Router1

```
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.10:1026     50.0.0.10:1026

R2#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
---  200.0.0.10        192.168.1.10       ---                ---
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.10:1025     50.0.0.10:1025
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.10:1026     50.0.0.10:1026
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1025     50.0.0.30:1025
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1026     50.0.0.30:1026
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1027     50.0.0.30:1027
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1028     50.0.0.30:1028
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1029     50.0.0.30:1029
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1030     50.0.0.30:1030
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1031     50.0.0.30:1031
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1032     50.0.0.30:1032
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1033     50.0.0.30:1033
tcp 200.0.0.10:80      192.168.1.10:80    50.0.0.30:1034     50.0.0.30:1034
```

## Result:

Successfully configured NAT for the given network setup, enabling seamless communication between private and public IP addresses.

## Viva-Voice Questions:

**Q) What is the primary purpose of NAT?**
**A.** The primary purpose of NAT is to translate private IP addresses to public IP addresses, allowing devices within a private network to access the internet.

**Q) What are the three types of NAT?**
**A.** The three types of NAT are Static NAT, Dynamic NAT, and Port Address Translation (PAT).

**Q) Why is IPv4 address exhaustion a problem?**
**A.** IPv4 address exhaustion is a problem because the limited number of available IP addresses cannot accommodate the growing number of internet-connected devices.

**Q) What is the difference between public and private IP addresses?**
**A.** Public IP addresses are routable on the internet, while private IP addresses are not routable and are used within local networks.

**Q) How does Static NAT differ from Dynamic NAT?**
**A.** Static NAT creates a one-to-one mapping between a private and a public IP address, while Dynamic NAT maps private IP addresses to a pool of public IP addresses on a first-come, first-served basis.

Date:

# Experiment No. 11

## Aim:

Demonstrate the working of Application Layer Protocols – DNS, HTTP, Email, FTP, and Telnet.

## Components and Tools:

| | |
|---|---|
| System | **:** Desktop Computer/Laptop |
| Operating System | **:** Windows/Linux |
| Tool | **:** Packet Tracer |
| Components | **:** Laptops (3), Switch (1), Router (1), Servers: DNS Server, HTTP Server, Email Server, FTP Server |

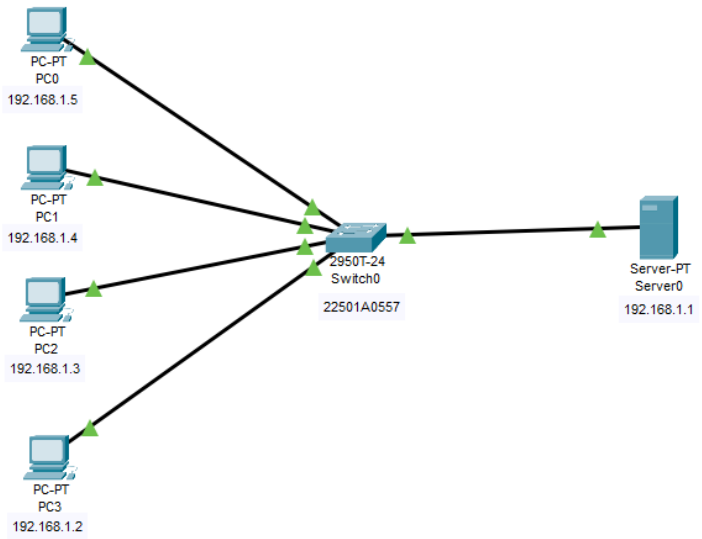
## Description:

### Application Layer Protocols:

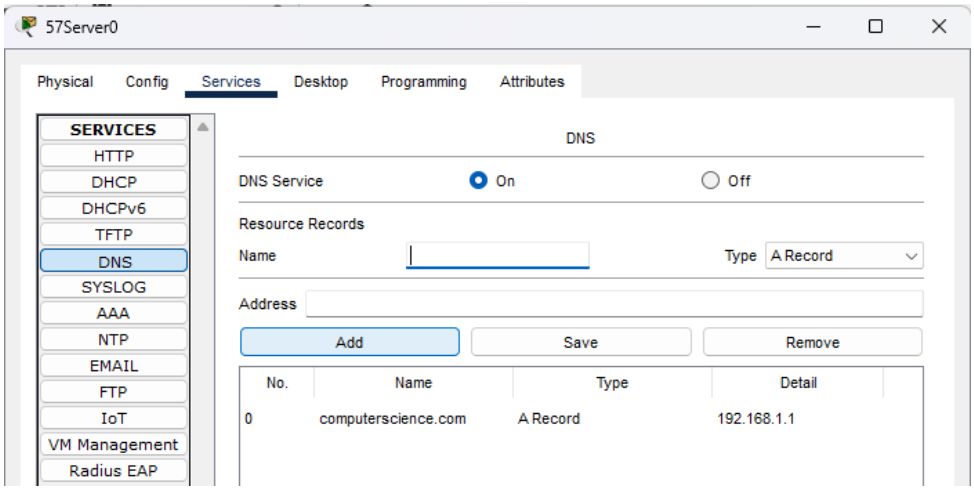
- **DNS (Domain Name System):** Resolves domain names to IP addresses, allowing users to access websites by name instead of numerical IP addresses.
- **HTTP (Hypertext Transfer Protocol):** Used for transmitting web pages over the internet, enabling the communication between web browsers and servers.
- **Email Protocols:** Such as SMTP (Simple Mail Transfer Protocol) for sending emails and POP3/IMAP for retrieving emails.
- **FTP (File Transfer Protocol):** Facilitates the transfer of files between computers on a network.
- **Telnet:** A protocol used for remotely managing devices over a network.

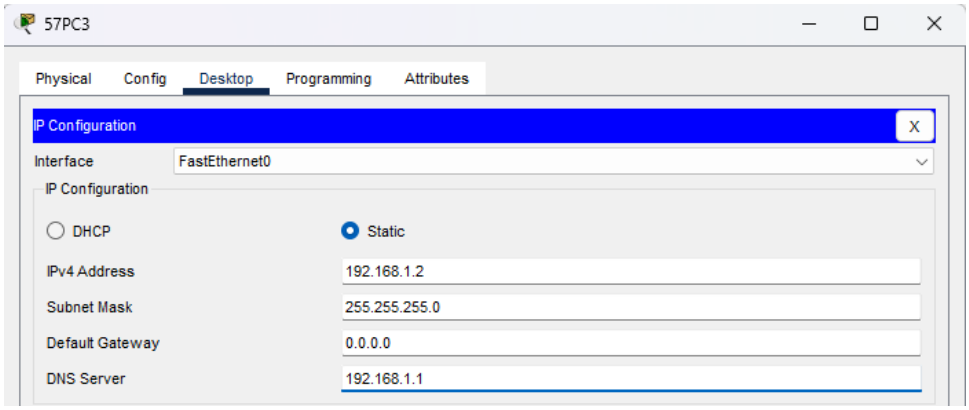
## Procedure:

### Task 11.1: Setup DNS, HTTP, and Email Server in Cisco Packet Tracer



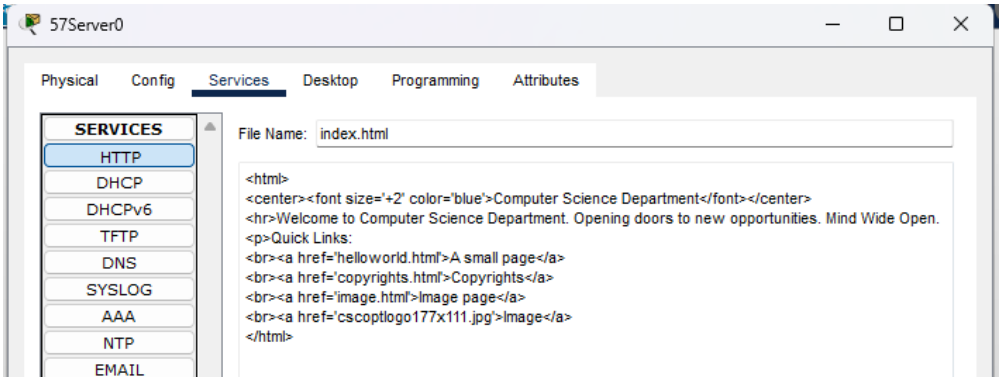
1. **Configure DNS Server:**
   - Click on the DNS Server.
   - Go to the Desktop tab, select DNS.
   - Add a new host, entering the hostname (e.g., example.com) and its corresponding IP address.



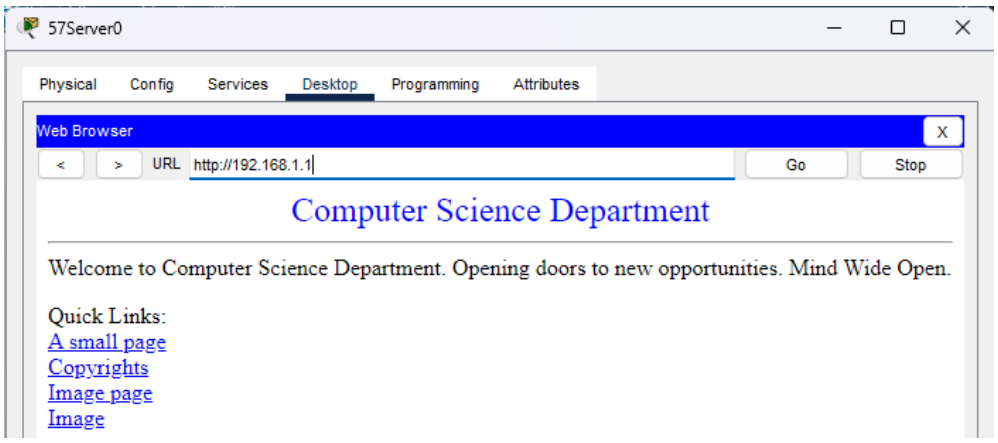




Configuring PCs with DNS Server

2. **Configure HTTP Server:**
   - Click on the HTTP Server.
   - Go to the Desktop tab, select HTTP.
   - Enable the HTTP service and configure a web page by editing the default page.

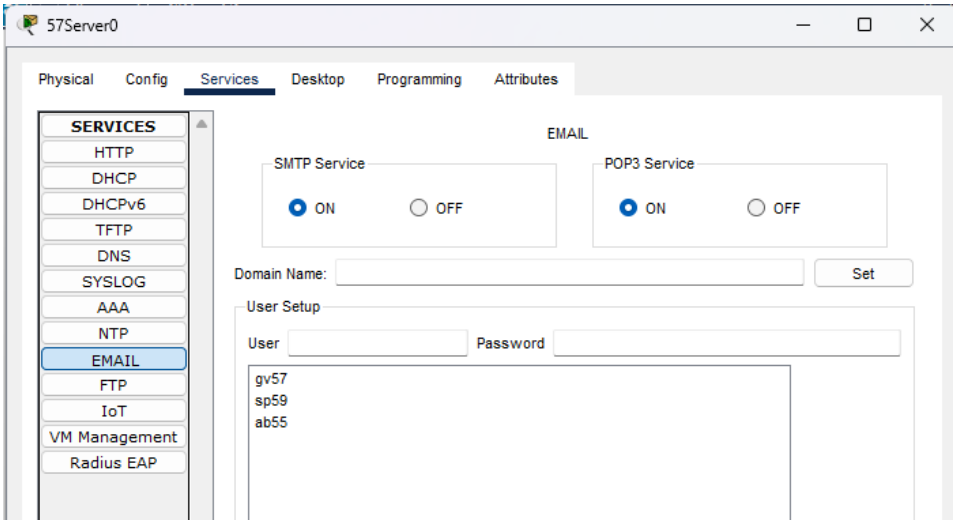



Editing original HTML to view changes

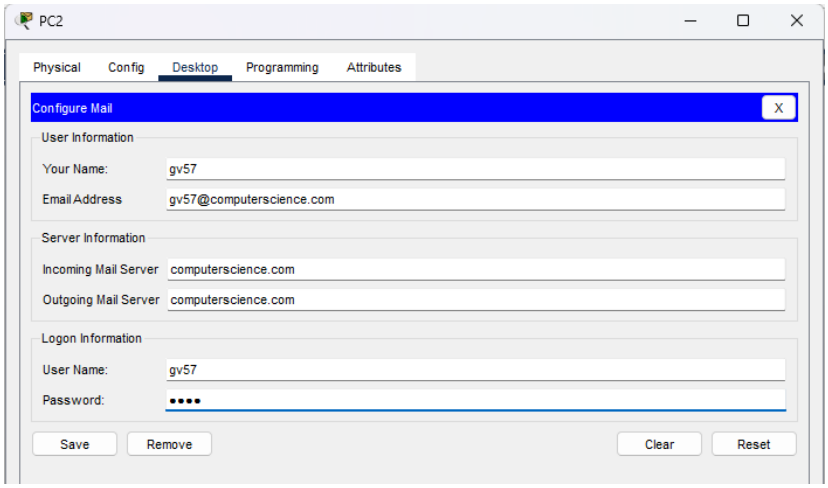



Verifying the changes in the website

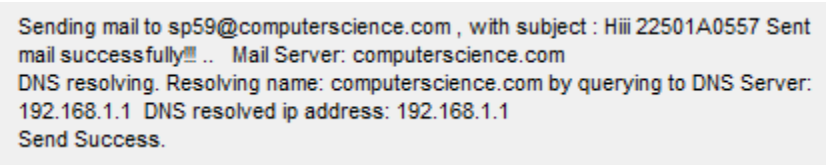
3. **Configure Email Server:**
   - Click on the Email Server.
   - Go to the Desktop tab, select Email.
   - Configure user accounts with usernames and passwords



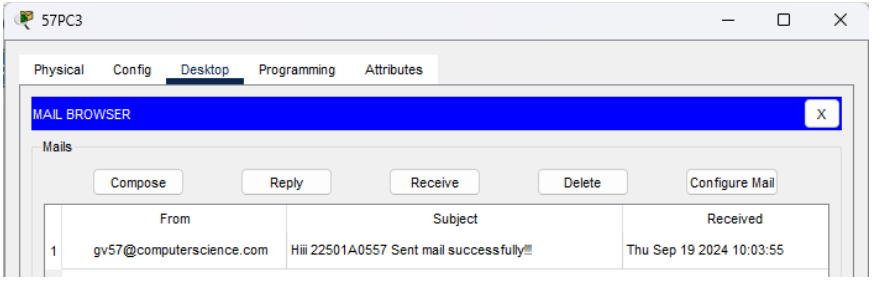

Creating New users for each PC



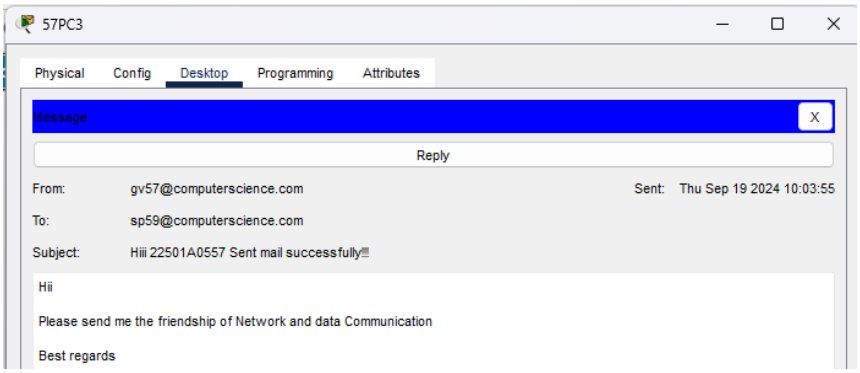

Configuring mail in PC2




Sent a mail from PC2




Successfully received mail on PC3

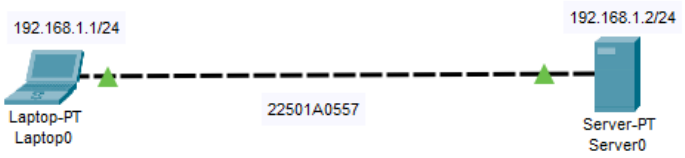



Viewing the mail on PC3

**Task 11.2: Setup FTP Server in Packet Tracer**

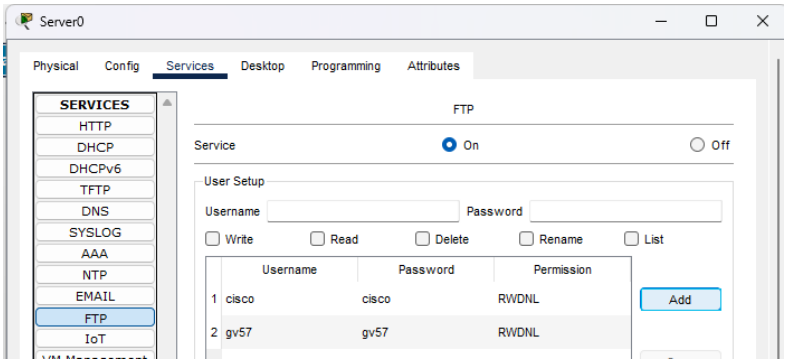
1. Build the network topology
2. Configure static IP addresses on the laptop and the server.

   Laptop: IP address: 192.168.1.1 Subnet Mask: 255.255.255.0
   Server: IP address: 192.168.1.2 Subnet Mask: 255.255.255.0




Configuring the FTP Server

3. Configure FTP in the server.
   - Go to services and open FTP Service.
   - Go to user setup and create a username (gv57) and password (gv57).
   - Select all the permissions (Write, Read, Delete, Rename, List) and add the user



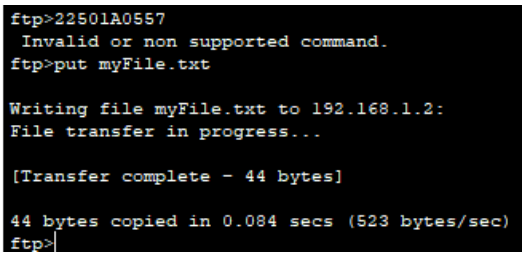

Adding a new User in FTP Server

4. Connect the FTP server from Laptop
   - From the Laptop's command prompt, FTP the server using the server IP address by typing:
     ftp 192.168.1.2
   - Provide the username(gv57) and password(gv57) for ftp login.
   - You are now in the FTP prompt.

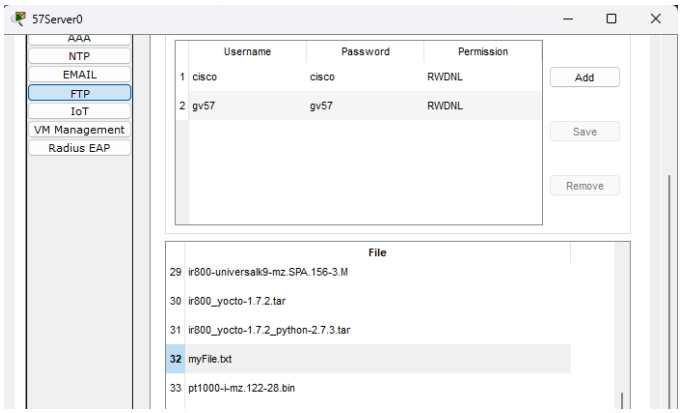
```
Cisco Packet Tracer PC Command Line 1.0
C:\>22501A0557
Invalid Command.

C:\>ftp 192.168.1.2
Trying to connect...192.168.1.2
Connected to 192.168.1.2
220- Welcome to PT Ftp server
Username:gv57
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

5. Create a file in the laptop then upload it to the server using FTP.
    - Open the Text Editor in the Laptop, create a file and give it your name of choice.
    - Type any text in the editor then save your file. e.g. myFile.txt.
6. Now upload the file from the laptop to the server using FTP (An FTP connection has to be started first. But this is what we've done in step 4) So to do an FTP upload, we'll type:

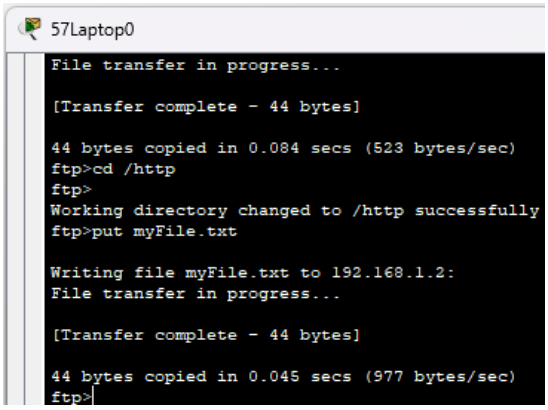
    put MyFile.txt



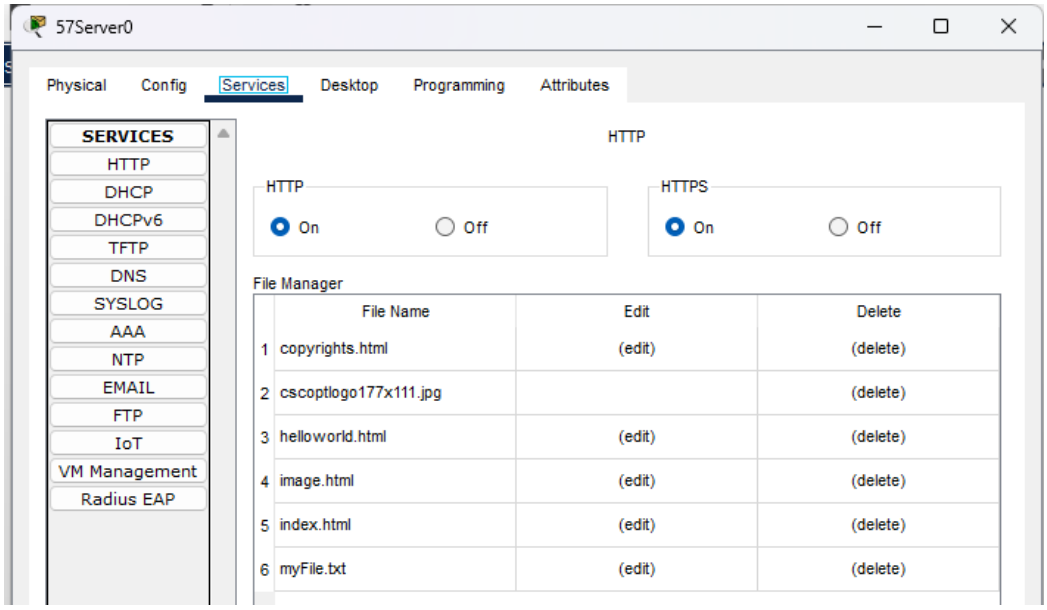

7. Once file upload is successful, go to the Server FTP directory to verify if the file sent has been received. To do this, go to Server-> Services->FTP. Here look for MyFile.txt sent from the laptop.




8. we can open an HTTP directory in the server by typing: cd /http.
    - This will change the current directory from FTP directory to HTTP directory
    - Once the http directory is open, you can upload a file to the HTTP server.
    - You're now uploading a file to an HTTP folder(directory) using FTP.
    - For example: put MyFile.txt • let's open an HTTP directory and upload(put) a file to it using FTP:



- You can now check up in the HTTP directory in the server and verify that the file uploaded from the Laptop(MyFile.txt) is well received:



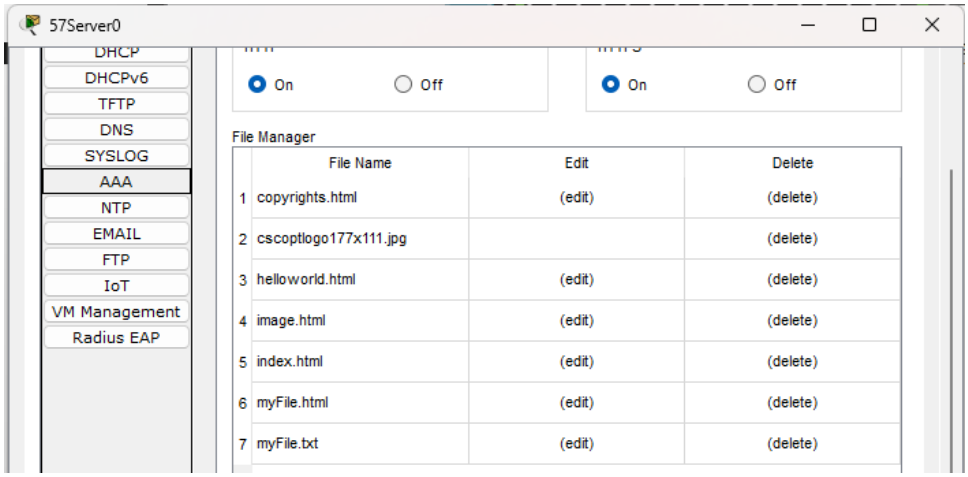

**Task 11.2.1:**

- **Create an HTML File:**
  - Open a text editor and type the markup for your HTML file.
  - Save the file as `myFile.html`.

- **Upload to HTTP Server Using FTP:**
  - Open Command Prompt.
  - Connect to the server using the IP address.
  - Enter the username and password.
  - Change the directory to HTTP.
  - Upload the file `myFile.html`.

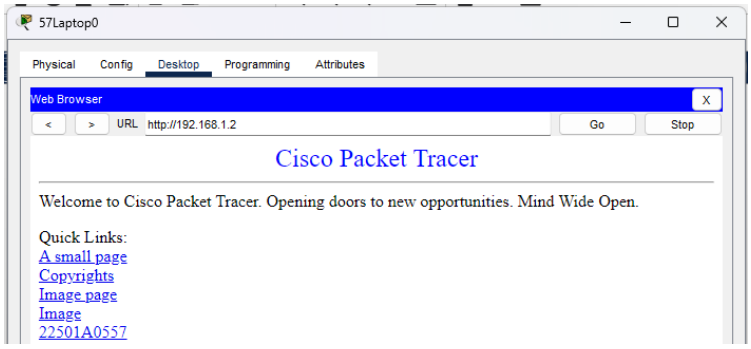
```
ftp>cd /http
ftp>
Working directory changed to /http successfully
ftp>put myFile.html

Writing file myFile.html to 192.168.1.2:
File transfer in progress...

[Transfer complete - 103 bytes]

103 bytes copied in 0.062 secs (1661 bytes/sec)
ftp>
```

- **Verify Upload:**
  - Go to **Server** > **Services** > **HTTP** and check for `myFile.html`.

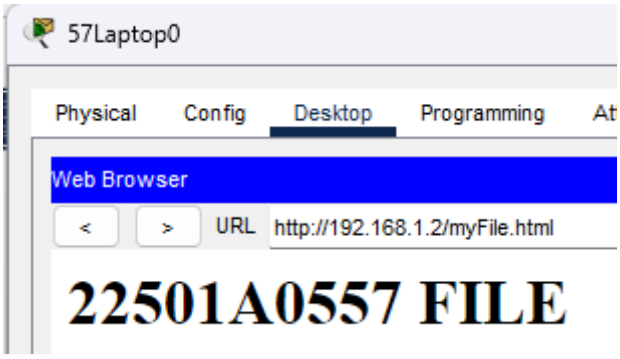



- **Edit `index.html`:**
  - Add a link to `myFile.html`.
  - Save the changes.

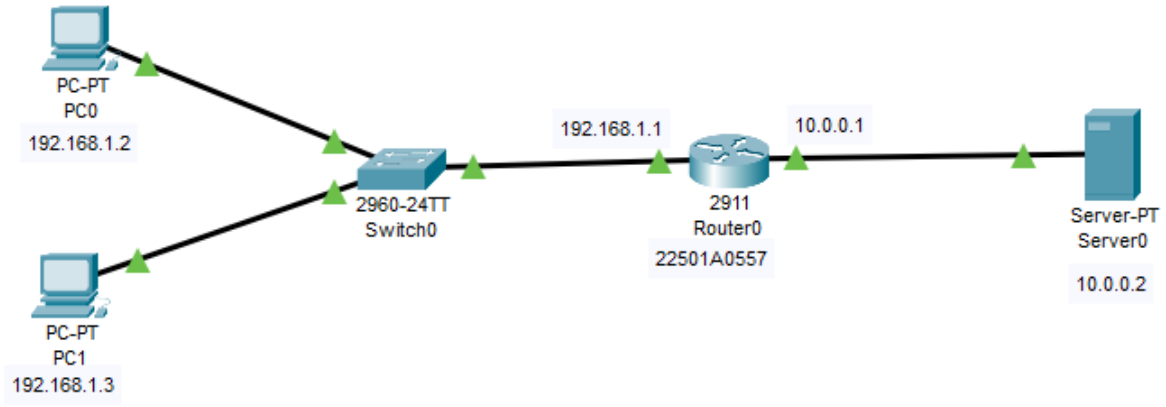
- **Access File from Laptop's Browser:**
  - Open the browser and enter the server's IP address.




  - Click the link to access `22501A0557`.



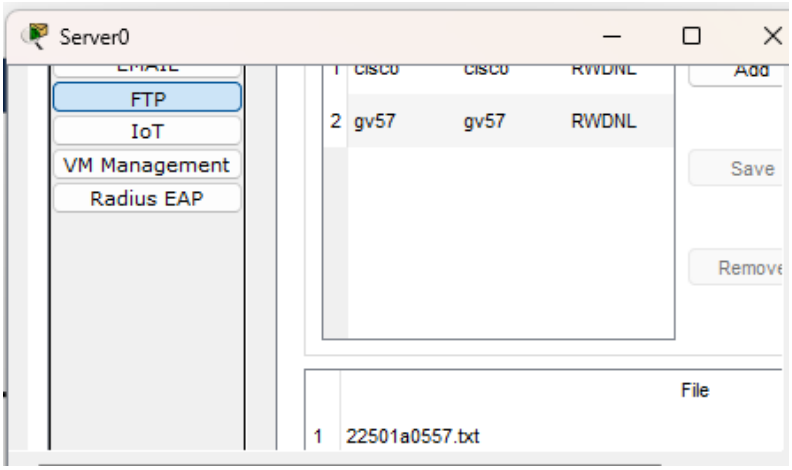
**Task of FTP Server:**




Creating an FTP Server

```
C:\>ftp 10.0.0.2
Trying to connect...10.0.0.2
Connected to 10.0.0.2
220- Welcome to PT Ftp server
Username:gv57
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put 22501a0557.txt

Writing file 22501a0557.txt to 10.0.0.2:
File transfer in progress...

[Transfer complete - 7 bytes]

7 bytes copied in 0.079 secs (88 bytes/sec)
ftp>
```

Sending the text file to FTP Server




Verifying the text file sent to FTP Server

## Telnet vs. SSH

### What is Telnet?

- Telnet is a protocol that allows you to connect to remote computers (called hosts) over a TCP/IP network. It provides a way to remotely control a device by sending commands.

### What is SSH?

- SSH (Secure Shell) is a secure remote login protocol that encrypts the password and other transmitted information, preventing it from being intercepted ("sniffed") by others.

### Enabling Telnet and SSH

- **VTY Ports**: To enable Telnet or SSH, we configure VTY (Virtual Teletype) ports. VTY ports are not physical but virtual ports used for remote access through Telnet or SSH.

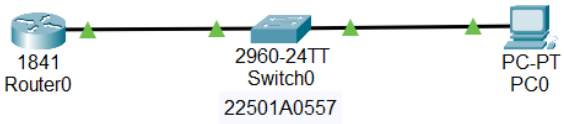
### Telnet vs. SSH

| Feature | Telnet | SSH |
|---|---|---|
| Purpose | Remote login | Secure remote login |
| Security | Less secure, sends data in plain text | More secure, encrypts all transmitted data |
| Authentication | No authentication | Uses public key for authentication |
| Data Encryption | No encryption | Encrypts data |
| Bandwidth Overhead | Lower overhead | Slightly higher overhead |
| Current Usage | Mostly replaced by SSH | Widely used due to security |

### Key Differences:

1. **Security**: SSH is more secure, encrypting all communication, while Telnet sends data in plain text.
2. **Authentication**: SSH uses public key authentication, whereas Telnet has no built-in authentication mechanism.
3. **Usage**: SSH is widely used and has replaced Telnet in most scenarios due to its security features.
4. **Bandwidth**: SSH adds some overhead due to encryption, while Telnet is lighter on bandwidth.

## Configuring Telnet:




| Device | Interface | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| R1 | Fa0/0 | 192.168.0.245 | /24 | N/A |
| PC1 | NIC | 192.168.0.15 | /24 | 192.168.0.245 |

1. Draw the above network in packet tracer and configure it with the addresses.
2. Open the Router's CLI and write the following lines to enable the telnet protocol.

```
Router(config)#line vty 0 4
```

3. Now open the PCs terminal and write the following lines to (We will try to remotely access the router from the PC using the telnet protocol):

```
Cisco Packet Tracer PC Command Line 1.0
C:\>telnet 192.168.0.245
Trying 192.168.0.245 ...Open

[Connection to 192.168.0.245 closed by foreign host]
```

4. As we were not able to access the router. Go back to Router's CLI and write the following code to set Line connection password (password for remote access Telnet):

```
Router(config-line)#password cisco
```

5. Return back to the PCs terminal and try accessing Router again:

```
C:\>telnet 192.168.0.245
Trying 192.168.0.245 ...Open


User Access Verification

Password:
Router>enable
% No password set.
```

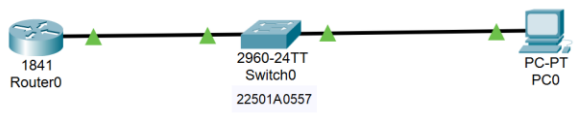
6. We were able to access the router now, but we were not able to reach the configuration mode of R1. So, now go back to Router's CLI and write the following code to set Router's Global Configuration mode password:

```
Router(config)#enable secret ct1306
```

7. Now we will gain full access to Router and we can configure it completely from PC

```
% No password set.
Router>telnet 192.168.0.245
Trying 192.168.0.245 ...Open


User Access Verification

Password:
Router>enable
Password:
Password:
Router#exit

[Connection to 192.168.0.245 closed by foreign host]
Router>
```

## Configuring SSH:




| Device | Interface | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| R1 | Fa0/0 | 10.0.0.1 | /8 | N/A |
| PC1 | NIC | 10.0.0.2 | /8 | 10.0.0.1 |

1. Draw the above network in packet tracer and configure it with the addresses.
2. Open the Router's CLI and write the following lines to enable the SSH protocol.

```
Router(config)#hostname R1
R1(config)#ip domain-name gv57.com
R1(config)#crypto key generate rsa
The name for the keys will be: R1.gv57.com
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 800
% Generating 800 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#username gv57 password gv57
*Mar 1 0:2:52.381: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#ip ssh time-out 30
R1(config)#ip ssh authentication-retries 5
R1(config)#line bty 0 4
                 ^
% Invalid input detected at '^' marker.

R1(config)#line vty 0 4
R1(config-line)#transport input ssh
R1(config-line)#login local
R1(config-line)#exit
```

3. Now open the PCs terminal and write the following lines to (We will try to remotely access the router from the PC using the ssh protocol):
   a. Use the password that has been set to ssh connection gv57

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ssh -l gv57 10.0.0.1

Password:



R1>enable
% No password set.
```

4. Although we were able to access the router, we were not able to reach the configuration mode of R1. So, now go back to R1's CLI and write the following code to set R1's Global Configuration mode password:

```
R1(config)#enable secret gv57
R1(config)#
```

5. Return back to the PCs terminal and try accessing R1 again, use the password that has been set for the Router's Global Configuration (gv57)

```
R1>ssh -l gv57 10.0.0.1

Password:



R1>enable
Password:
R1#
```

As we can see, we can Finally configure the router R1 with the help of SSH.

**Result:**

Successfully configured and tested the application layer protocols (DNS, HTTP, Email, FTP, Telnet and SSH) in the given network setup, enabling effective communication and file transfer across the network.

**Viva-Voice Questions:**

**Q) What is the function of DNS?**

**A:** DNS translates human-readable domain names into IP addresses for locating resources on the internet.

**Q) How does HTTP work?**

**A:** HTTP facilitates communication between web browsers and servers for transferring web content.

**Q) What are the main email protocols used?**

**A:** The main email protocols are SMTP for sending emails and POP3/IMAP for retrieving emails.

**Q) What is the purpose of FTP?**

**A:** FTP is used to transfer files between computers over a network.

**Q) How is Telnet useful?**

**A:** Telnet allows remote management of network devices through a command-line interface.

Date:

# Experiment - 12

## Aim:

Analyse and build a network for an organization using Network Simulation tool.

## Apparatus:

1. Tool: Packet Tracer
2. Components: As per the network setup

## Description:

This case study analyzes the network infrastructure of PVP Siddhartha Institute of Technology (PVPSIT), a technical education institute. PVPSIT is a medium-sized organization located in Vijayawada, with approximately 250 staff members supporting up to 1000 students in each department.

PVPSIT has 8 departments in total on the campus, plus 5 sections for administration. The organization is comprised of a main building having three floors on both sides, with four wiring closets per floor. The both sides of the main building are connected to the central network administrative (CNA) room; the switches from these three floors are connected with fiber connections to the CNA.

The Computer Science Department has three corridors, with three wiring closets per corridor. The three corridors totally comprise of with 4 laboratories, 10 class rooms, and 15 staff rooms. Within each of the three department corridors is a corridor switch. One main switch from each corridor connects to the main switch in CNA. The other switches connect either directly to each floor switch or via a daisy chain of switches, depending on which was most convenient at the time.

Currently, the LAN spans the entire campus. No Layer 3 switching is present. The address space is 172.16.0.0 /16. Addresses are coded sequentially into PCs as they are deployed. Staff members have been meaning to deploy DHCP but have not had the time. The applications that the organization is currently running include standard office applications, plus some specialized tools running over IP in each lab.

Web server, file server, and other servers are present in the CNA room located in computer science department. All the new servers are capable of using Gigabit or Gigabit Ether Channel connectivity. The WAN uses 56-kbps links to three floors of the college. The one router uses static routing that was configured by a previous network designer.

Each Lab in the computer science department has 80 computers. All staff rooms are to be equipped with Static LAN Connections and Wireless connectivity. The class rooms are to be connected wirelessly comprising of 80 students. The web server has a public IP. The file server has local IP within the department.

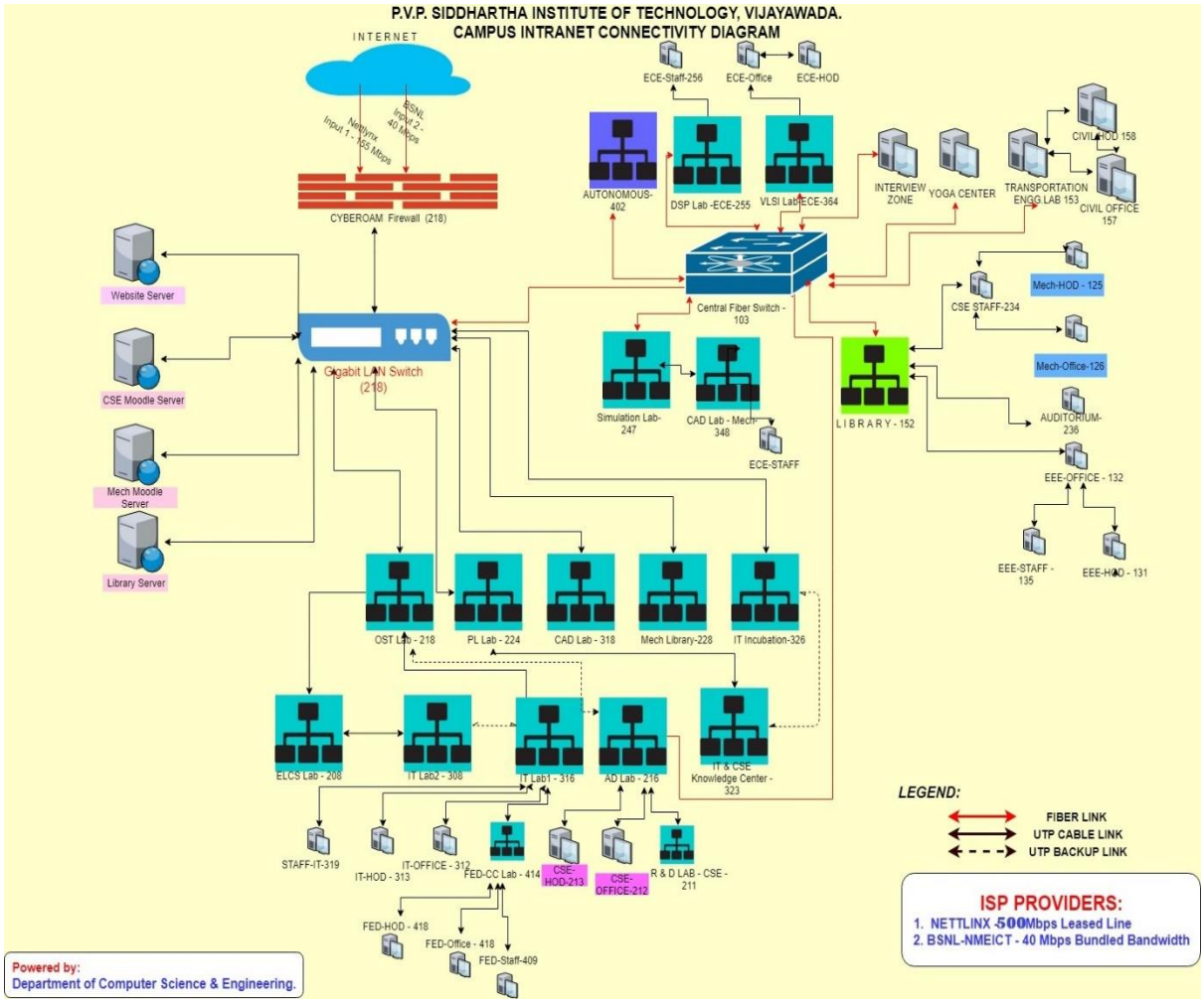



**Fig-12.1:** PVPSIT Network architecture

## Observations:

1. What kind of cable network does the college have and how is it configured?

A. The college uses **fiber connections** as the backbone for inter-building connectivity, especially from switches located on different floors to the Central Network Administrative (CNA) room. **UTP (Unshielded Twisted Pair)** cables are used for connecting the individual systems, classrooms, and labs to switches within departments.

2. List different types of connectors used in college?

A. **RJ-45 connectors** for UTP cables (Ethernet connections). **Fiber-optic connectors** such as **SC** or **LC** types for the fiber optic backbone connections.

3. What kinds of network interface cards (NICs) are used in the college's network, and how are they configured?

A. Most of the computers in labs and offices are likely equipped with **Gigabit Ethernet NICs** (1 Gbps) to ensure high-speed wired connections. The servers in the CNA room have **Gigabit EtherChannel capability**, meaning multiple gigabit Ethernet ports can be aggregated for higher bandwidth. NICs are configured to work with the campus' IP addressing scheme provided by the firewall and switch systems.

4. How fast is the college network, and what kind of bandwidth is available?

A. The college has two internet service providers (ISPs):

1. **NETLINX** providing a **500 Mbps leased line**.
2. **BSNL-NMEICT** providing **40 Mbps bundled bandwidth**.

5. How many switches does the college have and what kind of switches are they and their configuration?

A. The college has a significant number of switches, as follows:

- **Four switches per floor** across three floors in the main building.
- The **Computer Science Department** alone has a **main switch for each corridor** and switches for each laboratory, staff room, and classroom.
- The switches are primarily **Layer 2 (Data Link Layer)** switches, as **Layer 3 switching** is not yet present in the network.

Switches are connected in a **daisy chain** in some areas, while in other cases, they directly connect to the floor switches. The backbone switches are connected via **fiber** to the central CNA switch.

6. How many routers does the college have and what kind of routers are they?

A. The college has **one router**, which uses **static routing**. This suggests the router handles minimal dynamic routing, and all routing paths are manually configured.

7. What is the Host address, Network address, Subnet, Subnet mask, Subnet address of the college?

A. **Network address:** 172.16.0.0/16

**Host address:** Sequential assignment of IP addresses, with potential DHCP deployment in the future.

**Subnet Mask:** 255.255.0.0, allowing for over 65,000 host addresses.

**Subnet Address:** Since the entire network is under the 172.16.0.0/16 range, each department or floor likely gets a subset of this space.

8. How many servers does the college have and what are they used for and its configuration?

A. **Web Server:** Hosts the college's public-facing website with a **public IP address**.

**File Server:** Provides internal storage and file-sharing services within the department, using a **local IP**.

Other servers (like Moodle servers and library servers) are located in the CNA room. All servers are capable of **Gigabit Ethernet connectivity** and may use **EtherChannel** for load balancing and redundancy.

9. What kind of security measures does the college have in place to protect its network?

A. The college uses a **Cyberoam firewall** for network security. This firewall likely includes features such as:

- Intrusion Detection and Prevention (IDS/IPS)
- Content filtering
- Antivirus protection
- User authentication and access control
- Virtual Private Network (VPN) access for secure remote connections.

10. How does the College handle issues like network security or unauthorized access to the network?

A. **Access control policies** are managed via the **firewall** and switches, ensuring that only authorized users can access specific network resources. **Staff LAN connections** are static, while Wi-Fi networks likely use encryption protocols like **WPA2** to ensure secure access.

11. How does the college handle issues like network security breaches or hacking attempts?

A. The **Cyberoam firewall** likely provides real-time monitoring and alerts for security breaches. Additionally, logs from the firewall and servers can be analyzed to detect unusual traffic patterns, with IT staff responding to threats.

12. How does the college handle network during peak usage times?

A. The college may implement **Quality of Service (QoS)** and **Bandwidth management** policies through its switches or firewall to prioritize critical applications during peak hours. The significant bandwidth available (500 Mbps from the primary ISP) should be sufficient for handling heavy usage.

13. How does the college handle issues like network downtime or slow internet speeds?

A. **Dual ISP setup** provides redundancy in case of ISP failure. Network administrators may monitor the load and manually adjust resources or reroute traffic to ensure performance.

14. Are there any resources available to troubleshoot the network issues?

A. The college's IT team can use network diagnostic tools like **Wireshark** to analyze traffic at the packet level, and **firewall logs** to troubleshoot connectivity and security issues.

15. What is the virus detection software that is used in our college?

A. The **Cyberoam firewall** provides integrated antivirus scanning as part of its **Unified Threat Management (UTM)** functionality.

16. What kind of Wi-Fi network does the college have and how is it configured?

A. The diagram does not show specific Wi-Fi access points or wireless controllers, but Wi-Fi coverage is often implemented in institutions with centralized management systems (like Cisco or Aruba) using **dual-band** (2.4GHz and 5GHz) networks.

17. How many hubs does the college have, and what kind of hubs are they?

A. The diagram does not depict any **hubs** specifically, as modern networks typically use switches instead of hubs for better performance and security.

18. What kind of power supply does the college use for the network and how is it configured?

A. The diagram doesn't explicitly show the power supply details. However, switches and servers likely use **standard AC power** with possible **UPS (Uninterruptible Power Supply)** systems to ensure network reliability during power outages.

19.Analyze network traffic over PVPSIT network based on network layer, transport layer, and application layer protocols using Wireshark software.

A. This analysis would involve capturing packets using Wireshark to examine protocols at different layers:

- **Network Layer (IP)**: Check for routing and addressing.
- **Transport Layer (TCP/UDP)**: Analyze communication reliability and connection states.
- **Application Layer (HTTP/FTP)**: Evaluate web traffic, file transfers, etc.

20. Use Packet Tracer, create part of the Network/Topological Architecture of PVPSIT, configure certain application layer protocols used by the college, test and validate the communication over the network you designed.

A. Packet Tracer can simulate this network with VLANs, routing, and server-client communications for services like Moodle (HTTP/HTTPS), library (FTP/HTTP), etc.

## Result:

Network infrastructure and security configurations of PVPSIT have been successfully reviewed and assessed.