1. a)Implementation of LAN using suitable multiuser Windows operating System and demonstrating client-server and peer to peer mode of configuration.
b) Compare TCP/IP and OSI Reference model using 8 points

Point	**OSI Reference Model**	**TCP/IP Model**
1	Developed by ISO (International Standard Organization).	Developed by the U.S. Department of Defense (DoD).
2	Has **7 layers**.	Has **4 layers**.
3	Conceptual and more theoretical model.	Practical and protocol-oriented model.
4	Distinguishes **session** and **presentation** layers.	Combines them within the **application layer**.
5	Strictly follows the layer approach (each layer independent).	Layers are not as strictly separated.
6	Protocols were defined after model creation.	Model was defined after protocols were developed.
7	Provides a clear standard for manufacturers.	Provides actual communication protocols (TCP, IP, UDP, etc.).
8	Example protocols: FTAM, X.400, etc.	Example protocols: HTTP, FTP, SMTP, DNS, TCP, IP.

a) Simulate various Networks (LAN, WAN) using relevant network devices on Simulator i) Ping ii) ipconfig / ifconfig iii) Host name iv) Whois

b) Compare between TCP and UDP. Under what circumstances you will use them.

1️⃣ Ping Command

```
ping 192.168.1.2
```

2️⃣ IP Configuration Command (Windows)

```
ipconfig
```

3️⃣ Host Name Command

```
hostname
```

4️⃣ Whois Command (if installed / on Linux CMD)

```
whois google.com
```

1(b) Comparison between TCP and UDP

Point	**TCP (Transmission Control Protocol)**	**UDP (User Datagram Protocol)**
1	Connection-oriented protocol	Connectionless protocol
2	Provides reliable data transfer (error checking, acknowledgements)	Unreliable – no guarantee of delivery
3	Data transmitted in sequence (ordered delivery)	Packets may arrive out of order
4	Slower due to overhead of connection setup	Faster with less overhead
5	Suitable for large or critical data transfer	Suitable for real-time

Poin t	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
		applications
6	Example: HTTP, FTP, SMTP	Example: DNS, DHCP, VoIP, Video Streaming
7	Performs flow control and congestion control	No flow or congestion control
8	Establishes connection using 3-way handshake	No handshake; data sent directly

. a)Simulate various Networks (LAN, WAN) using relevant network devices on Simulator i) Netstat ii) Route iii) NSlookup iv) ARP v) Finger

b) Give general format of ICMP and explain different types of error reporting messages used in ICMP.

1️⃣Netstat Command

Displays active network connections and listening ports.

```
netstat
```

👉 For detailed info (protocols, addresses, ports, and states):

```
netstat -a
```

2️⃣Route Command

Displays or modifies the IP routing table.

```
route print
```

👉 To add or delete a route:

```
route add 192.168.1.0 mask 255.255.255.0 192.168.1.1
```

```
route delete 192.168.1.0
```

3️⃣NSlookup Command

Displays DNS information (to find IP address of a domain).

```
nslookup google.com
```

4️⃣ARP Command

Displays and modifies the IP-to-MAC address mapping table.

```
arp -a
```

5️⃣Finger Command

Displays information about a user on a remote system (if service is enabled).

```
finger username@hostname
```

📘 General Format of ICMP Packet:

Field	**Size (bits)**	**Description**
Type	8	Specifies the type of ICMP message (e.g., 0 = Echo Reply, 8 = Echo Request).
Code	8	Provides further details about the type of message.
Checksum	16	Used for error-checking of the ICMP header and data.

Field	Size (bits)	Description
Rest of Header	32	Varies depending on the type and code (e.g., for Echo — includes Identifier and Sequence Number).
Data	Variable	Contains additional data (often part of the original IP packet that caused the error).

🧩 ICMP Packet Structure (Diagrammatically)

```
+--------+--------+-----------------+-------------------+
| Type  | Code |    Checksum    |  Rest of Header  |
+--------+--------+-----------------+-------------------+
|            Additional Data / Payload           |
+---------------------------------------------------+
```

📡 Types of ICMP Error Reporting Messages

Type	Message Name	Meaning / Purpose
3	**Destination Unreachable**	Sent when a packet cannot reach its destination (e.g., no route, port unreachable).
4	**Source Quench** *(Deprecated)*	Informs sender to reduce transmission rate (for congestion control).
5	**Redirect Message**	Informs sender of a better route to the destination.
11	**Time Exceeded**	Sent when a packet's TTL (Time to Live) value reaches zero (used in traceroute).
12	**Parameter Problem**	Indicates an invalid or missing field in the IP header.

4. a) Observe and note the details of the live type of traffic for ARP, (Frame analysis, ethernet) from interface using packet capture and analysis tool wireshark.

b) Write a short note TCP congestion control.

c) Compare ARP and RARP protocol.

⚙️ 8. ARP Packet Capture (Step-by-Step Execution)

1️⃣ **Start Wireshark** → select Wi-Fi → Start Capture
2️⃣ **Filter**

```
arp
```

3️⃣ **Generate Traffic**

```
arp -d *
```

```
ping 192.168.1.1
```

4️⃣ **Observe**

- Who has 192.168.1.1? Tell 192.168.1.5
- 192.168.1.1 is at xx:xx:xx:xx:xx:xx

5️⃣ **Stop & Save**
File → **Save As → arp_analysis.pcapng**

(b) Short Note on TCP Congestion Control

📘 Definition:

TCP Congestion Control is a mechanism used by the **Transmission Control Protocol (TCP)** to prevent **network congestion** by controlling the rate at which data is sent.

Congestion occurs when the **network is overloaded**, causing packet loss and delays. TCP automatically adjusts its sending rate based on feedback from the network.

⚙️ Phases of TCP Congestion Control:

1. **Slow Start:**

- Transmission begins with a small congestion window (**cwnd** = **1 MSS**).
- Window size increases **exponentially** (doubles every RTT) until a threshold (**ssthresh**) is reached or packet loss occurs.

2. **Congestion Avoidance:**
   - After reaching the threshold, TCP increases **cwnd linearly** (by 1 MSS per RTT).
   - This helps to avoid congestion gradually.
3. **Fast Retransmit:**
   - If **three duplicate ACKs** are received, TCP assumes a packet loss and retransmits immediately (without waiting for timeout).
4. **Fast Recovery:**
   - Instead of going back to slow start, TCP reduces **cwnd** to half and continues linear growth.

Comparison between ARP and RARP Protocol

Feature	ARP (Address Resolution Protocol)	RARP (Reverse Address Resolution Protocol)
Full Form	Address Resolution Protocol	Reverse Address Resolution Protocol
Purpose	Converts **IP address → MAC address**	Converts **MAC address → IP address**
Used By	Sender (to find receiver's hardware address)	Diskless systems or clients (to know their own IP address)
Layer	Network Layer *(works between IP and Data Link)*	Network Layer
Direction of Mapping	Logical → Physical	Physical → Logical
Initiated By	Host sending data	Host without IP address

Feature	ARP (Address Resolution Protocol)	RARP (Reverse Address Resolution Protocol)
Message Type	ARP Request & ARP Reply	RARP Request & RARP Reply
Modern Replacement	Still in use *(ARP table maintained)*	Replaced by **BOOTP** and **DHCP**

5. a) Observe and note the details of the live type of traffic for RARP, (Frame analysis, ethernet) from interface using packet capture and analysis tool wireshark.

b) Explain ARP and RARP protocols in detail.

⚙️ **Procedure (Step-by-Step Execution):**

1. **Open Wireshark**
   - Launch Wireshark on your system.
2. **Select Interface**
   - Choose your active network adapter (e.g., *Wi-Fi* or *Ethernet0*).
   - Click **Start Capturing Packets**.
3. **Set Display Filter**
   - In the filter bar, type:
   - rarp
   - Press **Enter** to apply the filter.
4. **Generate or Wait for RARP Traffic**
   - (RARP is rarely seen on modern networks; it may appear in legacy systems or virtual lab setups.)

- If no RARP packets appear, simulate using an emulator or review a saved .pcap sample file.

5. **Stop Capture**
   - Click the **Red Square (Stop)** button after a few seconds of capture.
6. **Analyze Captured RARP Frame**
   - Select any **RARP packet** → Expand **Ethernet II** and **RARP** sections in the packet details pane

Comparison between ARP and RARP Protocol

Feature	ARP (Address Resolution Protocol)	RARP (Reverse Address Resolution Protocol)
Full Form	Address Resolution Protocol	Reverse Address Resolution Protocol
Purpose	Converts **IP address → MAC address**	Converts **MAC address → IP address**
Used By	Sender (to find receiver's hardware address)	Diskless systems or clients (to know their own IP address)
Layer	Network Layer *(works between IP and Data Link)*	Network Layer
Direction of Mapping	Logical → Physical	Physical → Logical
Initiated By	Host sending data	Host without IP address
Message Type	ARP Request & ARP Reply	RARP Request & RARP Reply
Modern Replacement	Still in use *(ARP table maintained)*	Replaced by **BOOTP** and **DHCP**

6. a)Capture and note the packet of HTTP Protocol using wireshark TCP-stream learn sequence of packets being sent and received.

b) Explain Hyper Text Transfer Protocol

🌐 5. HTTP Packet Capture (Step-by-Step Execution)

1️⃣ **Open Wireshark** → select Wi-Fi → Start Capture
2️⃣ **Set Filter**

```
http
```

3️⃣ **Generate HTTP Traffic**

```
curl http://example.com
```

or open http://neverssl.com in your browser.

4️⃣ **Observe Packets**

- GET / HTTP/1.1
- HTTP/1.1 200 OK

5️⃣ **Inspect Details**
Expand *Hypertext Transfer Protocol*:

- Method: GET
- Host: example.com
- Response Code: 200 OK
- Content-Type: text/html

6️⃣ **Follow TCP Stream**
Right-click → *Follow* → *TCP Stream*

7️⃣ **Stop & Save**
File → **Save As → http_capture.pcapng**

5(b) Hyper Text Transfer Protocol (HTTP)

📘 Definition:

HTTP (Hyper Text Transfer Protocol) is an **application layer protocol** used for **transferring hypertext documents** (like web pages) on the **World Wide Web (WWW)**.
It defines how **clients (browsers)** and **servers (web servers)** communicate and exchange information.

⚙️ Working Principle:

- HTTP follows a **client–server model**.
- The **client** sends an HTTP **request** to the server, and the **server** sends back an HTTP **response**.
- Communication takes place over **TCP/IP** (usually port **80** for HTTP and **443** for HTTPS).

📡 Features of HTTP:

1. **Connectionless:** Each request/response is independent.
2. **Stateless:** The server does not retain client information between requests.
3. **Media Independent:** Can transfer any type of data (text, image, video, etc.).
4. **Extensible:** New methods and headers can be added easily.

7. a)Capture and note the packet of DHCP Protocol using wireshark TCP-stream learn sequence of packets being sent and received.

b) Write short note on DHCP.

🌐 1. DHCP Packet Capture (Step-by-Step Execution)

1️⃣Open Wireshark

- Launch Wireshark.
- Select your network interface (e.g., Wi-Fi).
- Click **Start Capturing Packets**.

2️⃣Set Display Filter

- In the filter bar, type:
- dhcp

(or use bootp if no DHCP packets appear)

3️⃣Generate DHCP Traffic

- Open Command Prompt → type:
- ipconfig /release

(Releases your current IP)

- Then type:

- ipconfig /renew

(Requests a new IP from DHCP server)

4️⃣**Observe Packets in Wireshark**
You will see the 4-step DHCP process:

- DHCP Discover — Client broadcasts IP request
- DHCP Offer — Server offers IP address
- DHCP Request — Client requests offered IP
- DHCP ACK — Server assigns IP

5️⃣**Inspect a Packet**

- Expand the *Bootstrap Protocol (BOOTP)* section.
- Observe:
    - Message Type (Discover/Offer/Request/ACK)
    - Client MAC Address
    - Your (client) IP Address
    - Server Identifier

6️⃣**Follow UDP Stream**

- Right-click a DHCP packet → *Follow* → *UDP Stream*

7️⃣**Stop & Save**

- Click Stop (red square).
- File → **Save As → dhcp_capture.pcapng**
- **Definition:**
  DHCP stands for **Dynamic Host Configuration Protocol**. It is a **network layer protocol** used to **automatically assign IP addresses** and other network configuration parameters such as **subnet mask, default gateway, and DNS server** to client devices in a network.

  Purpose:
  The main purpose of DHCP is to **reduce manual configuration** of IP addresses and to ensure that **each device receives a unique IP address** dynamically, avoiding duplication or conflicts.

Working Principle:
DHCP works on the **client–server model**.
When a device connects to a network, the **DHCP client** on that device sends a request to the **DHCP server**, which provides an available IP address and other configuration information.

The process involves four key steps (called the **DORA process**):

1. **Discover** – The client broadcasts a message to locate a DHCP server.
2. **Offer** – The server responds with an available IP address offer.
3. **Request** – The client requests to use the offered IP address.
4. **Acknowledge** – The server confirms and assigns the IP address.

8. Observe and note the working of protocols using PING / TRACEROUTE / PATHPING and capture packets in LAN using packet capture and analysis tool Cisco Packet Tracer.

b) Give general format of ICMP and explain different types of error reporting messages used in ICMP.

PING (ICMP) Packet Capture (Step-by-Step Execution)

1️⃣Open Wireshark

- Launch Wireshark.
- Select active interface (e.g., Wi-Fi).
- Start Capturing Packets.

2️⃣Set Display Filter
Type:

```
icmp
```

and press Enter.

3️⃣Generate ICMP Traffic
Open Command Prompt → type:

```
ping -4 -n 4 google.com
```

This sends 4 ICMP Echo Requests and gets Echo Replies.

4️⃣**Observe in Wireshark**
You’ll see:

- Echo Request — from your PC
- Echo Reply — from destination

5️⃣**Inspect Packet Details**

- Expand *Internet Control Message Protocol* section:
    - Type: 8 (Request) / 0 (Reply)
    - Code: 0
    - Checksum
    - Identifier & Sequence Number

6️⃣**Follow ICMP Stream**
Right-click → *Follow* → *ICMP Stream*

7️⃣**Stop & Save**
File → **Save As → ping_icmp_capture.pcapng**

🛰️ 3. TRACEROUTE Packet Capture (Step-by-Step Execution)

1️⃣**Open Wireshark**

- Launch Wireshark → select Wi-Fi interface → Start Capture.

2️⃣**Set Display Filter**

```
icmp
```

(Traceroute uses ICMP Time Exceeded messages.)

3️⃣**Generate Traceroute Traffic**
Open Command Prompt → type:

```
tracert google.com
```

4️⃣**Observe in Wireshark**
You’ll see packets from your system with increasing **TTL (Time To Live)** values and ICMP responses:

- ICMP “Time Exceeded” from intermediate routers

- ICMP “Echo Reply” from final destination

5️⃣Inspect Packets

- Expand ICMP layer → check *Type* (11 = Time Exceeded, 0 = Echo Reply)
- Note *TTL values* increasing hop by hop

6️⃣Stop & Save
File → **Save As → tracert_capture.pcapng**

🌍 4. PATHPING Packet Capture (Step-by-Step Execution)

1️⃣Open Wireshark

- Start capture on your network interface.

2️⃣Set Filter

```
icmp
```

3️⃣Generate PATHPING Traffic
In Command Prompt, type:

```
pathping google.com
```

(This runs traceroute + ping for each hop.)

4️⃣Observe Packets
You’ll see:

- ICMP Echo Requests & Replies
- Time Exceeded messages from routers
- Multiple pings to each hop (round-trip time calculation)

5️⃣Inspect Details

- Expand ICMP section → view Type and Sequence numbers
- Compare responses from each hop

6️⃣Stop & Save
File → **Save As → pathping_capture.pcapng**

1️⃣Introduction:

ICMP (Internet Control Message Protocol) is a **Network Layer protocol** used by network devices like **routers and hosts** to send **error messages** and **diagnostic information** about IP packet delivery.
It works along with the **IP protocol** and helps in **network troubleshooting** (for example, commands like ping and traceroute use ICMP).

2️⃣General Format of ICMP Packet:

Field	Size (in bits)	Description
Type	8	Identifies the type of ICMP message (e.g., 0 = Echo Reply, 3 = Destination Unreachable).
Code	8	Provides further information about the message type.
Checksum	16	Used for error-checking of the ICMP header and data.
Rest of Header	32	Varies depending on the type and code (e.g., Identifier, Sequence Number, etc.).
Data	Variable	Additional data such as part of the original IP packet.

3️⃣ICMP Packet Structure (Diagrammatically):

```
+--------+--------+----------------+------------------+
|  Type  |  Code  |     Checksum    |  Rest of Header  |
+--------+--------+----------------+------------------+
|              Additional Data / Payload             |
+---------------------------------------------------+
```

4️⃣Types of ICMP Error Reporting Messages:

ICMP Type	Message Name	Explanation / Usage
3	**Destination Unreachable**	Sent when a packet cannot reach its destination due to routing failure, port unreachable, or network

ICMP Type	Message Name	Explanation / Usage
		issues.
4	**Source Quench** *(deprecated)*	Used to inform the sender to slow down transmission rate to avoid congestion.
5	**Redirect Message**	Sent by a router to inform a host of a better route for packet delivery.
11	**Time Exceeded**	Sent when the **TTL (Time To Live)** field of a packet reaches zero — used by traceroute.
12	**Parameter Problem**	Sent when there is an invalid field in the IP header (e.g., error in options or checksum).

13. a) Capture and note the packet of HTTP Protocol using wireshark TCP-stream learn sequence of packets being sent and received.
b) Explain the TCP Connection management in Client/Server model.

🌐 5. HTTP Packet Capture (Step-by-Step Execution)

1️⃣**Open Wireshark** → select Wi-Fi → Start Capture
2️⃣**Set Filter**

```
http
```

3️⃣**Generate HTTP Traffic**

```
curl http://example.com
```

or open http://neverssl.com in your browser.

4️⃣**Observe Packets**

- GET / HTTP/1.1
- HTTP/1.1 200 OK

5️⃣**Inspect Details**
Expand *Hypertext Transfer Protocol*:

- Method: GET
- Host: example.com

- Response Code: 200 OK
- Content-Type: text/html

6️⃣Follow TCP Stream
Right-click → *Follow* → *TCP Stream*

7️⃣Stop & Save
File → **Save As → http_capture.pcapng**

Answer:

1️⃣Introduction:

TCP (Transmission Control Protocol) is a **connection-oriented**, **reliable** transport layer protocol.
Before data transfer begins, a **connection** must be established between the **client** and the **server**.
TCP manages this connection using a **three-way handshake** for connection establishment and a **four-step process** for connection termination.

2️⃣TCP Connection Management Phases:

TCP connection management involves two major stages:

1. **Connection Establishment (Three-Way Handshake)**
2. **Connection Termination (Four-Way Handshake)**

3️⃣Connection Establishment (Three-Way Handshake):

The purpose of the three-way handshake is to **synchronize sequence numbers** and **acknowledgment numbers** between client and server to start reliable communication.

Step	Description	Flag Used
Step 1: SYN	The **client** sends a **SYN** (synchronize) packet to the server to initiate a connection and synchronize sequence numbers.	SYN = 1
Step 2: SYN + ACK	The **server** responds with a **SYN-ACK** packet, acknowledging the client's SYN and sending its own sequence number.	SYN = 1, ACK = 1

Step	Description	Flag Used
Step 3: ACK	The **client** sends an **ACK** packet back to the server, acknowledging the connection establishment.	ACK = 1

✅ Now the TCP connection is established and ready for data transfer.

Diagram: Three-Way Handshake

```
Client                    Server
-------------------------------------------------------
SYN ---------------------------------->  (Step 1)
      <------------------ SYN + ACK   (Step 2)
ACK ----------------------------------->  (Step 3)
```

9. a)Configure servers like HTTP and understand packet sequence and data flowing between client-server using packet analysis tools Cisco Packet Tracer.

b) Give the classification of commonly used Unicast Routing protocols and explain Distance Vector Routing protocol with an appropriate example

🌐 STEP-BY-STEP: HTTP Configuration in Cisco Packet Tracer

🧩 Step 1: Add Devices

1. Open **Cisco Packet Tracer**.
2. From the bottom device bar, **drag and drop:**
   - **1 PC**
   - **1 Switch (e.g., 2960)**
   - **1 Server**

🔌 Step 2: Connect Devices

1. Click on the **Lightning bolt (⚡)** icon (Connections).

2. Select **Copper Straight-Through Cable**.
3. Connect the devices as follows:
    - **PC → Switch (FastEthernet0)**
    - **Server → Switch (FastEthernet0)**
4. Wait until both links turn **green** (active connection).

⚙️ Step 3: Assign IP Addresses

We'll use this simple IP plan:

Device	**Interface**	**IP Address**	**Subnet Mask**
PC	Fa0	192.168.1.10	255.255.255.0
Server	Fa0	192.168.1.2	255.255.255.0

➤ On the PC:

1. Click the **PC** → *Desktop* → *IP Configuration*
2. Enter:
    - IP Address: 192.168.1.10
    - Subnet Mask: 255.255.255.0
    - Leave Gateway blank (for now)

➤ On the Server:

1. Click the **Server** → *Config* → *FastEthernet0*
2. Enter:
    - IP Address: 192.168.1.2
    - Subnet Mask: 255.255.255.0
    - Gateway: (leave blank or add if using router later)

🌍 Step 4: Enable HTTP Service on Server

1. On the **Server**, go to **Services** tab.
2. Click **HTTP** on the left panel.
3. Turn **HTTP: ON** (you can turn HTTPS off for now).
4. Optional: Edit the **index.html** page — write something like:
5. <h1>Welcome to my Packet Tracer Web Server!</h1>

🧠 Step 5: Test Connectivity

1. On the **PC**, open *Desktop* → *Command Prompt*.
2. Type:
3. ping 192.168.1.2

✅ If you get replies — connection is fine.

🌐 Step 6: Access the Webpage

1. On the **PC**, open *Desktop* → *Web Browser*.
2. In the URL bar, type:
3. http://192.168.1.2
4. You should see the web page you edited earlier!

🎉 Congratulations! You've successfully configured **HTTP in Packet Tracer**.

1️⃣Introduction

Routing protocols are used by routers to determine the best path for forwarding data packets across networks.
When data is sent from **one source to one destination**, it is known as **Unicast Routing**.

2️⃣ Classification of Unicast Routing Protocols

Unicast routing protocols are broadly classified into two main types:

Category	Description	Examples
1. Static Routing	Routes are manually configured by the network administrator.	No specific protocol (manually set)
2. Dynamic Routing	Routes are automatically learned and updated by routers using routing algorithms.	

4️⃣ Distance Vector Routing Protocol

Definition:

Distance Vector Routing Protocols determine the best path to a destination based on **distance (hop count)** and **direction (vector)**.

Each router maintains a **routing table** that contains:

- Destination network
- Next hop router
- Distance (number of hops)

Routers **periodically exchange** their routing tables with their **direct neighbors** to update routes.

Key Characteristics:

- Uses **Bellman–Ford algorithm**
- **Hop count** used as a metric
- Updates are sent **periodically** and when changes occur
- **Slow convergence** (compared to link-state)
- May cause **routing loops** (prevented using Split Horizon, Hold-down timers, etc.)

5️⃣ Example:

Consider three routers connected as follows:

R1 ---- R2 ---- R3

Route r	**Initial Knowledge**	**After Exchange (RIP)**
R1	Knows only its own network (N1)	Learns N2 and N3 from R2
R2	Knows N1 and N3	Acts as intermediate router
R3	Knows only N3	Learns N1 and N2 from R2

Example Metric Calculation:

- R1 to R2 → 1 hop
- R1 to R3 → 2 hops (via R2)
  Hence, **R1 chooses R2 as next hop** for reaching R3.

10.a)Configure servers like FTP and understand packet sequence and data flowing between client-server using packet analysis tool Cisco Packet Tracer.

b) What is the purpose of FTP? What are the three FTP transmission modes? Explain.

🌐 STEP-BY-STEP: FTP Configuration in Cisco Packet Tracer

🧩 Step 1: Add Devices

- Open **Cisco Packet Tracer**.
- From the bottom device bar, drag and drop:
  - **1 PC**
  - **1 Switch (e.g., 2960)**
  - **1 Server**

🔌 Step 2: Connect Devices

1. Click the **Lightning bolt (⚡)** icon (Connections).
2. Choose **Copper Straight-Through Cable**.
3. Connect the devices as follows:
   - **PC → Switch (FastEthernet0)**
   - **Server → Switch (FastEthernet0)**
4. Wait for both links to turn **green** (active).

⚙️ Step 3: Assign IP Addresses

We'll use the same simple IP plan:

Device	**Interface**	**IP Address**	**Subnet Mask**
PC	Fa0	192.168.1.10	255.255.255.0
Server	Fa0	192.168.1.2	255.255.255.0

➤ On the PC:

- Click **PC → Desktop → IP Configuration**
  - IP Address: 192.168.1.10
  - Subnet Mask: 255.255.255.0
  - Leave Gateway blank

➤ On the Server:

- Click **Server → Config → FastEthernet0**
  - IP Address: 192.168.1.2
  - Subnet Mask: 255.255.255.0
  - Gateway: *(leave blank or add if router used later)*

🌍 Step 4: Enable FTP Service on Server

1. On the **Server**, go to the **Services** tab.
2. Click **FTP** on the left menu.
3. Toggle **FTP: ON**.
4. In the **User Accounts** section, add a new account:
   - **Username:** student
   - **Password:** 123
   - **Directory:** (default /)
5. You can leave the default file list or upload new files in the **FTP Root Folder** (optional).

🧠 Step 5: Test Connectivity

1. On the **PC**, open **Desktop → Command Prompt**.
2. Type:
3. ping 192.168.1.2

✅ If you get replies, your connection is working properly.

📁 Step 6: Access the FTP Server

Option 1 — Using Command Prompt

1. On the PC's Command Prompt, type:
2. ftp 192.168.1.2
3. When prompted:
4. Username: student
5. Password: 123
6. Once logged in, you can type:
7. dir   (to list files)
8. get file.txt   (to download file)
9. put file.txt   (to upload file)

quit

1️⃣Purpose of FTP

FTP (File Transfer Protocol) is an **application layer protocol** used to **transfer files** between a **client** and a **server** over a **TCP/IP network**.

It provides a reliable and efficient way to **upload**, **download**, and **manage files** remotely.

📘 Main Purposes of FTP:

1. **File Transfer:** Upload and download files between local and remote systems.
2. **File Access:** View and manage files (rename, delete, or move) on a remote server.
3. **Data Sharing:** Facilitate file exchange among users in different locations.
4. **Remote Storage:** Access and maintain files on web or FTP servers.

2️⃣FTP Transmission Modes

FTP supports **three modes of data transmission**, depending on how data is formatted and sent between client and server.

Mode	Name	Description
1. Stream Mode	Default mode	Data is transmitted as a continuous stream of bytes. TCP handles segmentation and reassembly.
2. Block Mode	Data is sent in blocks	Each block of data is preceded by a 3-byte header describing the block size and type. Used for efficient data transfer.
3. Compressed Mode	Data is compressed	Data is compressed before transmission using algorithms like run-length encoding to save bandwidth.

11a).Configure servers for DNS and understand data flowing between client-server using packet analysis tool Cisco Packet Tracer.

b) Write short on DNS in Internet.

🧩 Devices and Tools Required:

- 1 × DNS Server
- 1 × Client PC
- 1 × Switch
- Cisco Packet Tracer (latest version)

⚙️ Step-by-Step Configuration

Step 1: Network Setup

1. Open **Cisco Packet Tracer**.
2. Drag and drop the following devices:
   - **1 Server** (for DNS)
   - **1 PC** (client)
   - **1 Switch** (to connect both)
3. Connect:
   - **PC → Switch → Server** using **Copper Straight-Through Cables**.

Step 2: Assign IP Addresses

Device	Interface	IP Address	Subnet Mask	Gateway
Server (DNS)	FastEthernet0	192.168.1.2	255.255.255.0	192.168.1.1
PC (Client)	FastEthernet0	192.168.1.3	255.255.255.0	192.168.1.1

Step 3: Configure DNS Server

1. Click on the **Server** → go to **Services tab**.
2. Turn **ON** the **DNS service**.
3. In the **DNS table**, click **Add** and enter:
    - **Name:** www.example.com
    - **Address:** 192.168.1.2 *(or any web server IP you'll host)*
4. Click **Save**.

Step 4: Configure Client DNS Settings

1. Click the **PC** → go to **Desktop tab → IP Configuration**.
2. Enter:
    - **IP Address:** 192.168.1.3
    - **Subnet Mask:** 255.255.255.0
    - **Default Gateway:** 192.168.1.1
    - **DNS Server:** 192.168.1.2 *(same as your DNS server IP)*

Step 5: Verify DNS Resolution

1. On the **PC**, open **Command Prompt (Desktop → Command Prompt)**.
2. Type:
3. ping www.example.com
4. The DNS server should resolve the domain to its IP address (192.168.1.2), and ping replies should be received.

✅ **Successful Resolution Output Example:**

```
Pinging www.example.com [192.168.1.2] with 32 bytes of data:

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

Step 6: Observe Data Flow (Packet Analysis)

1. Switch to **Simulation Mode** (bottom-right corner of Packet Tracer).

2. In the **Add Simple PDU** tool, send a ping from the **PC** to www.example.com.
3. Observe packets moving between the **Client** and **DNS Server**:
    - **DNS Query Packet:** Client → DNS Server
    - **DNS Response Packet:** DNS Server → Client
4. You can click on each packet to view detailed **Encapsulation / Decapsulation** and protocol layers.

📘 Definition:

DNS (Domain Name System) is a **hierarchical and distributed naming system** used on the **Internet** to translate **human-readable domain names** (like www.google.com) into **IP addresses** (like 142.250.183.100), which computers use to identify each other on the network.

🔍 Purpose of DNS:

- Humans find it easier to remember names rather than IP addresses.
- DNS acts as the **"phonebook of the Internet"**, converting domain names to corresponding IP addresses automatically.
- It allows users to access websites using names instead of numeric IPs.

⚙️ Working of DNS:

When a user enters a website name in a browser:

1. The request goes to a **DNS Resolver** (usually provided by the ISP).
2. The resolver contacts:
    - **Root DNS Server** → points to TLD servers (.com, .org, etc.)
    - **TLD Server** → directs to the authoritative server for that domain.
    - **Authoritative DNS Server** → returns the IP address of the requested domain.
3. The browser then connects to that IP to load the website.

🧩 Components of DNS:

1. **DNS Resolver (Client Side):** Sends query on behalf of user.
2. **Root Server:** The top-level DNS server that directs queries to TLD servers.
3. **TLD Server:** Handles domains like .com, .org, .edu, etc.
4. **Authoritative Server:** Contains actual domain-to-IP mappings.

12a) Configure servers for Email and understand data flowing between client-server using packet analysis tool Cisco Packet Tracer.

b) Explain transition strategies between IPv4 to IPv6.

🧩 Devices and Tools Required:

- 1 × **Server** (Mail Server)
- 2 × **PCs** (Client A and Client B)
- 1 × **Switch**
- Cisco **Packet Tracer**

⚙️ Step-by-Step Configuration

Step 1: Network Topology Setup

1. Open **Cisco Packet Tracer**.
2. Drag and drop:
   - **1 Server**

   - **2 PCs**
   - **1 Switch**

3. Connect all devices using **Copper Straight-Through Cables**.
   - PC0 → Switch
   - PC1 → Switch
   - Server → Switch

Step 2: Assign IP Addresses

Device	Interface	IP Address	Subnet Mask	Gateway
Server (Mail Server)	FastEthernet0	192.168.1.2	255.255.255.0	192.168.1.1
PC0 (User A)	FastEthernet0	192.168.1.3	255.255.255.0	192.168.1.1
PC1 (User B)	FastEthernet0	192.168.1.4	255.255.255.0	192.168.1.1

Step 3: Configure the Email Server

1. Click on the **Server** → go to the **Services tab**.
2. Turn **ON** both:
   - **SMTP (Simple Mail Transfer Protocol)**
   - **POP3 (Post Office Protocol)**
3. Click on **Email Service** → Add users:

Username	**Password**	**Email ID**
user1	123	user1@mail.com
user2	123	user2@mail.com

4. Save settings and make sure both services show **ON (green)**.

Step 4: Configure Client PCs (Email Application)

On PC0 (User A):

1. Click **PC0 → Desktop → Email**.
2. Configure as:
   - **Name:** User1
   - **Email:** user1@mail.com
   - **Incoming Mail Server:** 192.168.1.2
   - **Outgoing Mail Server:** 192.168.1.2
   - **Username:** user1
   - **Password:** 123
3. Click **Save**.

On PC1 (User B):

1. Click **PC1 → Desktop → Email**.
2. Configure as:
   - **Name:** User2
   - **Email:** user2@mail.com
   - **Incoming Mail Server:** 192.168.1.2

- **Outgoing Mail Server:** 192.168.1.2
- **Username:** user2
- **Password:** 123

3. Click **Save**.

Step 5: Send and Receive Email

1. On **PC0 (User1)** → Open **Email Application**.
2. Click **Compose** →
   - **To:** user2@mail.com
   - **Subject:** Test Mail
   - **Message:** “Hello User2 – this is a test email.”
   - Click **Send**.
3. On **PC1 (User2)** → Open **Email Application** → Click **Receive**.
   - You should see the mail from **user1@mail.com**.

✅ **Mail Transfer Successful!**

Step 6: Observe Data Flow (Packet Analysis)

1. Switch to **Simulation Mode** (bottom right).
2. In the **event list**, observe the following protocols:
   - **SMTP (Port 25)** – Used when sending email from client to server.
   - **POP3 (Port 110)** – Used when retrieving email from server to client.
3. Click on each packet → view **Encapsulation Details** to analyze:
   - Source and destination IPs
   - Protocol used (SMTP/POP3)

  - Message direction (Client → Server or Server → Client)

📘 Introduction:

The Internet was originally built using **IPv4 (Internet Protocol version 4)**, which uses **32-bit addresses**.
However, due to the **exhaustion of IPv4 addresses**, a new protocol — **IPv6 (Internet Protocol version 6)** with **128-bit addresses** — was developed.

Since the entire Internet cannot switch to IPv6 instantly, **transition strategies** are used to allow **IPv4 and IPv6 to coexist** and **communicate** during the migration period.

⚙️ Major IPv4 to IPv6 Transition Strategies:

There are **three main strategies** used for transitio

1️⃣Dual Stack

- Devices run **both IPv4 and IPv6 protocols** simultaneously.
- Communication occurs using either version depending on the destination.
- **Advantage:** Simple and supports gradual migration.
- **Example:** A router configured with both IPv4 and IPv6 addresses.

2️⃣Tunneling

- **Encapsulates IPv6 packets inside IPv4 packets** to pass through an IPv4 network.
- Used when two IPv6 networks are separated by an IPv4 infrastructure.
- **Types:** 6to4, Teredo, and Manual Tunneling.

3️⃣ Translation (NAT64 / DNS64)

- Converts **IPv6 packets into IPv4** and vice versa for communication between incompatible systems.
- **Useful** when IPv6-only and IPv4-only devices must interact.

✅ Conclusion:

13.a) Execute DHCP Server using simulator Cisco Packet Tracer.
b) Explain 3 way handshake during connection establishment at Transport layer?

🌐 STEP-BY-STEP: DHCP Configuration in Cisco Packet Tracer

🧩 Step 1: Add Devices

- Open **Cisco Packet Tracer**.
- From the bottom device bar, drag and drop:
  - **1 Server**
  - **1 Switch (e.g., 2960)**
  - **1 or more PCs**

🔌 Step 2: Connect Devices

1. Click the **Lightning bolt (⚡)** icon (Connections).
2. Choose **Copper Straight-Through Cable**.
3. Connect:
   - **Server → Switch (FastEthernet0)**
   - **PC(s) → Switch (FastEthernet0)**
4. Wait until all links turn **green** (active connection).

⚙️ Step 3: Assign IP to Server

We'll make the server a DHCP server, so only **it** gets a static IP.

➤ On the Server:

1. Click **Server → Config → FastEthernet0**
2. Set:
   - IP Address: 192.168.1.1
   - Subnet Mask: 255.255.255.0
3. Leave **Gateway blank** (for simple LAN setup).

🌍 Step 4: Enable and Configure DHCP Service

1. On the **Server**, go to **Services → DHCP**.
2. Make sure the **DHCP service is ON**.
3. In the **Service Configuration** area, fill in details:
   - **Pool Name:** LAN
   - **Default Gateway:** 192.168.1.1
   - **DNS Server:** 192.168.1.1 *(optional)*
   - **Starting IP Address:** 192.168.1.2
   - **Subnet Mask:** 255.255.255.0
   - **Maximum Number of Users:** 10 *(or more as needed)*
4. Click **Save**.

✅ Now, your DHCP server is ready to assign IPs automatically.

🧠 Step 5: Configure PCs to Receive IP Automatically

1. Click on **PC → Desktop → IP Configuration**
2. Select **DHCP** instead of Static.
3. Wait a few seconds.
4. You'll see:

- **IP Address:** auto-assigned (e.g., 192.168.1.2)
- **Subnet Mask:** 255.255.255.0
- **Default Gateway:** 192.168.1.1

💡 Repeat this for all PCs if you've added more than one.

🧾 Step 6: Verify IP Allocation

1. On any **PC → Desktop → Command Prompt**, type:
2. ipconfig
3. You should see an IP in the **192.168.1.x** range — assigned by the server.

✅ That confirms your DHCP is working properly!

📶 Step 7: Test Network Connectivity

1. From a PC, open the **Command Prompt** and type:
2. ping 192.168.1.1

(Ping the DHCP Server)

- If you get replies, everything is set up correctly.

🎉 Step 8: Save the Project

Once working →
File → Save As → dhcp_configuration.pkt

✅ **Congratulations!** You have successfully configured and verified **DHCP (Dynamic Host Configuration Protocol)** in Cisco Packet Tracer.
Your server is now automatically assigning IP addresses to all connected PCs!

The **3-way handshake** is the process used by the **Transmission Control Protocol (TCP)** at the **transport layer** to **establish a reliable connection** between a client and a server before data transmission begins.

It ensures that **both sides are ready to communicate** and that **sequence numbers** are synchronized.

⚙️ Steps of TCP 3-Way Handshake:

Step	Message	Description
1️⃣ **SYN (Synchronize)**	Client → Server	The **client** sends a **SYN** packet to the server to initiate a connection and synchronize sequence numbers.
2️⃣ **SYN + ACK (Synchronize + Acknowledge)**	Server → Client	The **server** responds with a **SYN-ACK** packet, acknowledging the client's request and sending its own sequence number.
3️⃣ **ACK (Acknowledge)**	Client → Server	The **client** sends an **ACK** packet back to confirm the connection. Connection is now established.

🧩 Diagram:

Client   Server

SYN -------------------------------->   (Step 1)

      <------------------- SYN + ACK   (Step 2)

ACK -------------------------------->   (Step 3)

13.a) Execute Proxy, web Server using simulator Cisco Packet Tracer.
b) List the typical QoS parameters in the Transport Layer and explain each one.

🌐 STEP-BY-STEP: "Proxy" via HTTP + DNS Control (Packet Tracer)

🧩 Step 1: Add Devices

- Open **Cisco Packet Tracer**.
- Drag and drop:
    - **1 Switch (2960)**
    - **1 Server** → *Server1 (HTTP)*

- **1 Server** → *Server2 (Control/DNS)*
- **1 PC** → *Client PC*

🔌 Step 2: Connect Devices

1. Click ⚡ **Connections** → **Copper Straight-Through**.
2. Connect:
   - **PC → Switch (Fa0)**
   - **Server1 → Switch (Fa0)**
   - **Server2 → Switch (Fa0)**
3. Ensure all link lights turn **green**.

🗺️ Step 3: IP Plan (Static)

Device	**Interface**	**IP Address**	**Subnet Mask**
Server 1	Fa0	192.168.1.2	255.255.255.0
Server 2	Fa0	192.168.1.3	255.255.255.0
PC	Fa0	192.168.1.10	255.255.255.0

(Gateway not required for single LAN demo.)

⚙️ Step 4: Configure Interfaces

➤ Server1 (HTTP)

- **Server1 → Config → FastEthernet0**
  - IP: 192.168.1.2
  - Mask: 255.255.255.0

➤ Server2 (Control/DNS)

- **Server2 → Config → FastEthernet0**
  - IP: 192.168.1.3
  - Mask: 255.255.255.0

➤ **PC**

- **PC → Desktop → IP Configuration**
  - IP: 192.168.1.10
  - Mask: 255.255.255.0
  - **DNS Server:** 192.168.1.3 ← (this forces all name lookups to "proxy" server)

🌍 Step 5: Enable Services

➤ **On Server1 (HTTP backend)**

1. **Services → HTTP → Turn ON**
2. (Optional) Edit **index.html**:
3. <h1>Welcome from Server1 (192.168.1.2)</h1>

➤ **On Server2 (Control/DNS)**

1. **Services → DNS → Turn ON**
2. Add an **A record**:
   - **Name:** www.site.com
   - **Address:** 192.168.1.2 *(points to Server1)*
3. (Optional "policy control" via DNS):
   - **Block a site** by mapping it to nowhere or local:
     - www.blocked.com → 0.0.0.0 *(or leave absent to return NXDOMAIN)*
   - **Redirect** a site:
     - www.redirect.com → 192.168.1.2 *(lands on your local page)*

This is the "proxy-like" control: Server2 decides what names resolve.

🧪 Step 6: Verify Basic Connectivity

On **PC → Desktop → Command Prompt**:

ping 192.168.1.2

ping 192.168.1.3

✅ Both should reply.

🌐 Step 7: Test Name Resolution via Control/DNS Server

On **PC → Command Prompt**:

nslookup www.site.com

- **Server** should show 192.168.1.3 (your DNS) as the resolver.
- **Address** returned should be 192.168.1.2.

(If nslookup isn't available in PT, test via browser in next step.)

🧭 Step 8: Browse Through the "Proxy" Path

On **PC → Desktop → Web Browser**:

http://www.site.com

✅ You should see Server1's web page (served at 192.168.1.2), but **the decision to reach it came from Server2 (DNS)**.

(b) Typical QoS Parameters in the Transport Layer

Quality of Service (QoS) parameters at the **Transport Layer** define how effectively data is transmitted between sender and receiver.
The main parameters are:

QoS Parameter	Description
1. Bandwidth (Throughput)	The amount of data transmitted per unit time. Higher bandwidth means more data transfer capacity.
2. Delay (Latency)	The time taken for a packet to travel from source to destination. Lower delay = better performance.

QoS Parameter	Description
3. Jitter	Variation in packet arrival times. Low jitter ensures smooth real-time communication (e.g., voice/video).
4. Packet Loss	Percentage of packets lost during transmission. Reliable transport (TCP) minimizes packet loss.
5. Reliability	Ensures data integrity and delivery using acknowledgment, retransmission, and sequencing.
6. Flow Control	Prevents sender from overwhelming the receiver (e.g., TCP sliding window mechanism).
7. Congestion Control	Regulates data flow when the network is overloaded to prevent packet drops.

.