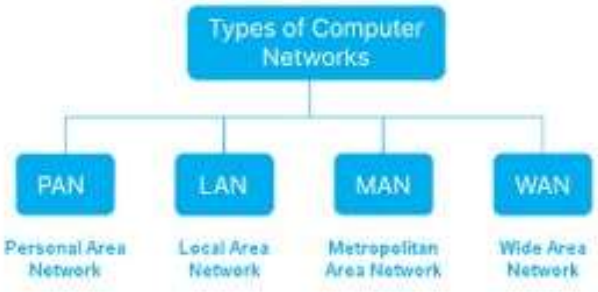
COMPUTER NETWORKS COURSE
UNIT WISE IMPORTANT QUESTIONS

UNIT-I

1. **What is computer network? And describe various types of networks?**

   A computer network is a system that connects two or more computing devices (such as computers, servers, routers, or phones) to share resources, exchange data, and communicate. These devices are linked using wired or wireless technologies, allowing users to access files, printers, internet, and applications from any connected system.

   Types of Computer Networks:



   LAN (Local Area Network):

   - Covers a small geographical area like a home, office, or school.
   - High speed and low cost.
   - Example: Connecting multiple computers in a computer lab.

   MAN (Metropolitan Area Network):

   - Spans a city or a large campus.
   - Connects multiple LANs within a region.
   - Example: Network in a university with different campuses across a city.

   WAN (Wide Area Network):

   - Covers large geographical areas like countries or continents.
   - Uses public networks like telephone lines or satellites.
   - Example*:* The Internet.

   PAN (Personal Area Network):

   - Very small network used for personal devices within a range of a few meters.
   - Example*:* Bluetooth connection between phone and headset.

   WLAN (Wireless LAN):

   - Similar to LAN but uses wireless technology like Wi-Fi.
   - Common in homes and cafes for internet access.

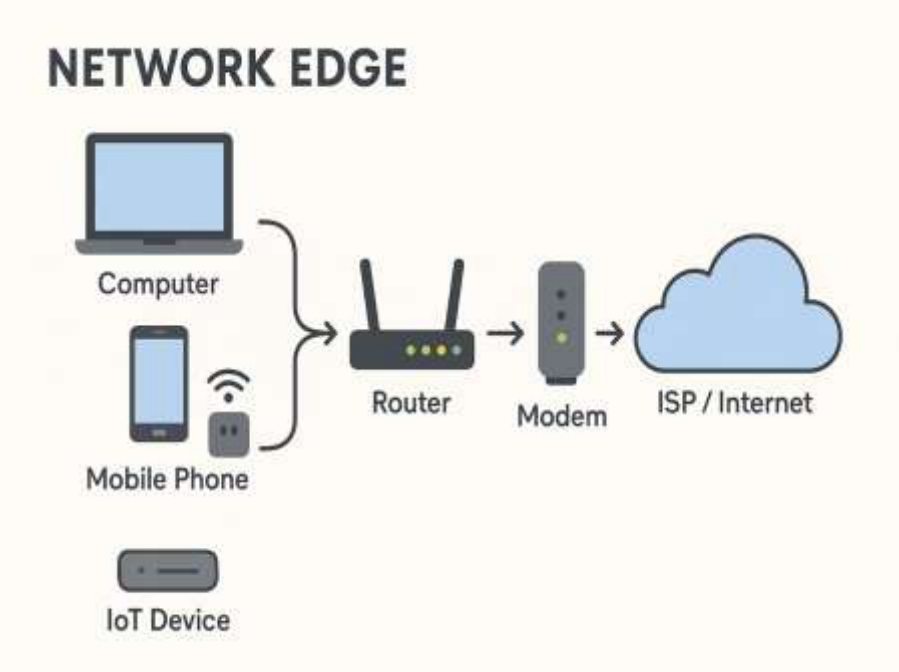
2. **Describe about network edge and mention the role of devices in the network edge?**

The network edge refers to the part of the computer network where end-user devices (clients) interact with the network. This is where communication originates or ends, such as when a user accesses a website or sends an email. The edge is the entry and exit point of data in the network.

Devices at the Network Edge and Their Roles

Device	Role
Computers/Laptops	Access applications, send/receive data.
Mobile Phones	Connect to the internet via Wi-Fi or mobile networks.
Routers	Connect local devices to the internet and manage data flow.
Modems	Convert digital signals to analog (and vice versa) for internet access.
IoT Devices	Smart sensors and devices that send/receive data to/from the network.
Access Points (Wi-Fi)	Provide wireless access to wired network resources.

- **User Devices:** Generate or consume data.
- **Router/Modem:** Transfer data to and from the internet.
- **ISP:** Internet Service Provider connects the edge to the global network.



3. **What is the difference between network edge and network core? And explain the role of devices in a network core?**

✅ Difference Between Network Edge and Network Core

Point	Network Edge	Network Core
What it is	Where users connect to the internet or network	The center part of the network that moves data everywhere
Main Job	Sends/receives data from users like phones, computers	Transfers data quickly between different networks
Devices used	Computers, phones, routers, modems	Big routers, switches, fiber cables
Speed	Normal speed, good for browsing or calling	Very fast, handles large amounts of data
Users	End-users (you and me)	No direct users, only used to send data across cities/world

Role of Devices in Network Core

Device	Role
Core Router	- It is a big and powerful router. - Sends data between different cities or countries very fast.
Switch	- Connects many devices in the core. - Helps to move data quickly from one point to another.
Fiber Cables	- Very fast cables made of glass. - Used to carry internet between places over long distances.
Data Centers	- Large buildings with many servers. - Store and send data (like YouTube videos, emails, websites).

---

4. **What is meant by packet switched network and describe virtual circuit network and data gram network?**

A packet switched network is a type of network where data is broken into small packets before being sent. Each packet travels independently through the network and is reassembled at the destination.

- It is used in the Internet.
- It is efficient and handles many users at the same time.

Types of Packet Switched Networks

There are two main types:

1. Virtual Circuit Network

- A temporary path is set up before data is sent.
- All packets follow the same path.
- Packets arrive in order.
- **Example:** ATM (Asynchronous Transfer Mode), Frame Relay

Advantages:

- Reliable
- Less chance of packet loss

Disadvantages:

- Takes time to set up the connection

2. Datagram Network

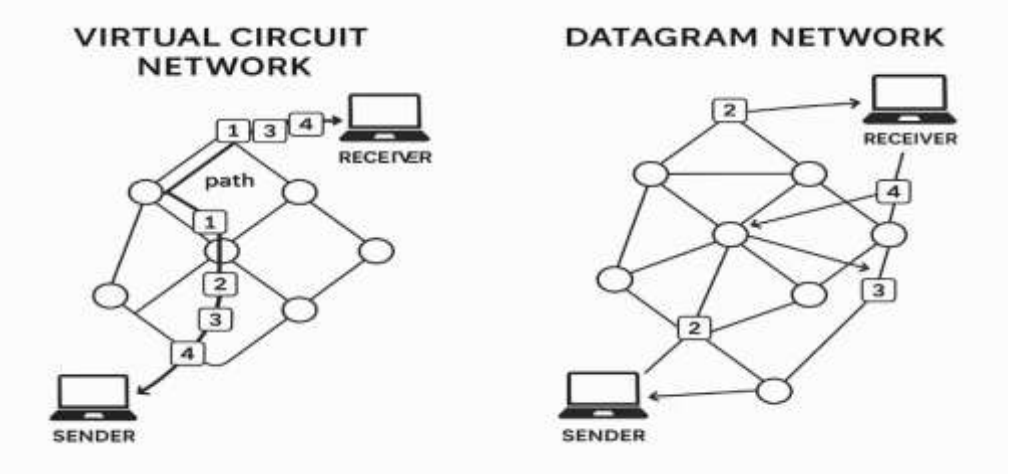
- No fixed path
- Each packet can take a different route to the destination
- Packets may arrive out of order
- **Example:** The Internet using IP protocol

Advantages:

- Fast
- No need to set up a path before sending

Disadvantages:

- May cause delay or out-of-order delivery



5. **Explain about delay, loss and throughput in packet switched networks?**

1. Delay: Delay is the time taken for a data packet to move from the sender to the receiver.

Types of Delay:

- **Processing Delay**: Time taken to check for errors and decide where to send the packet.
- **Queuing Delay:** Time a packet waits in a queue at the router.
- **Transmission Delay:** Time taken to push all bits of the packet onto the communication link.
- **Propagation Delay:** Time it takes for the data to travel through the wire or fiber.

Example: If you click a website link and it takes two seconds to open, that time is the delay.

2. Loss: Loss happens when data packets are dropped and do not reach the destination.

Reasons for Packet Loss:

- The router's memory is full due to heavy traffic.
- Network issues like link failure or congestion.
- Data gets damaged due to noise or errors during transmission.

Result: The sender may need to send the data again, which increases delay and reduces efficiency.

3. Throughput: Throughput is the amount of data successfully sent across the network in a given time.

Units of Measurement:

- It is measured in bits per second, such as Mbps or Gbps.

Types:

- **Instantaneous Throughput:** Speed at a specific moment.
- **Average Throughput:** Average speed calculated over time.
- **Example:** If you download a 100 MB file in 10 seconds, the throughput is 10 MB per second.

6. **What is OSI model and write about the functions of each layer?**
    - The OSI model stands for Open Systems Interconnection model.
    - It is a standard model used to understand how data moves from one computer to another in a network.
    - The OSI model has seven layers. Each layer has its own role and helps in sending or receiving data in a step-by-step process.
    - It helps different systems and devices to communicate with each other.

Functions of Each Layer

1. Physical Layer

- This is the first layer.
- It is responsible for sending raw bits over a physical connection like wires or cables.
- It deals with hardware like switches and cables.

2. Data Link Layer

- This layer helps in transferring data from one device to another without any errors.
- It adds information like the address of the sender and receiver to the data.

3. Network Layer

- This layer selects the best path for data to travel across networks.
- It uses addresses like IP addresses to find the destination.

4. Transport Layer

- This layer makes sure that data is delivered correctly and completely.
- It controls the flow of data and handles errors during transmission.

5. Session Layer

- This layer manages communication between two devices.
- It starts, manages, and ends the communication sessions.

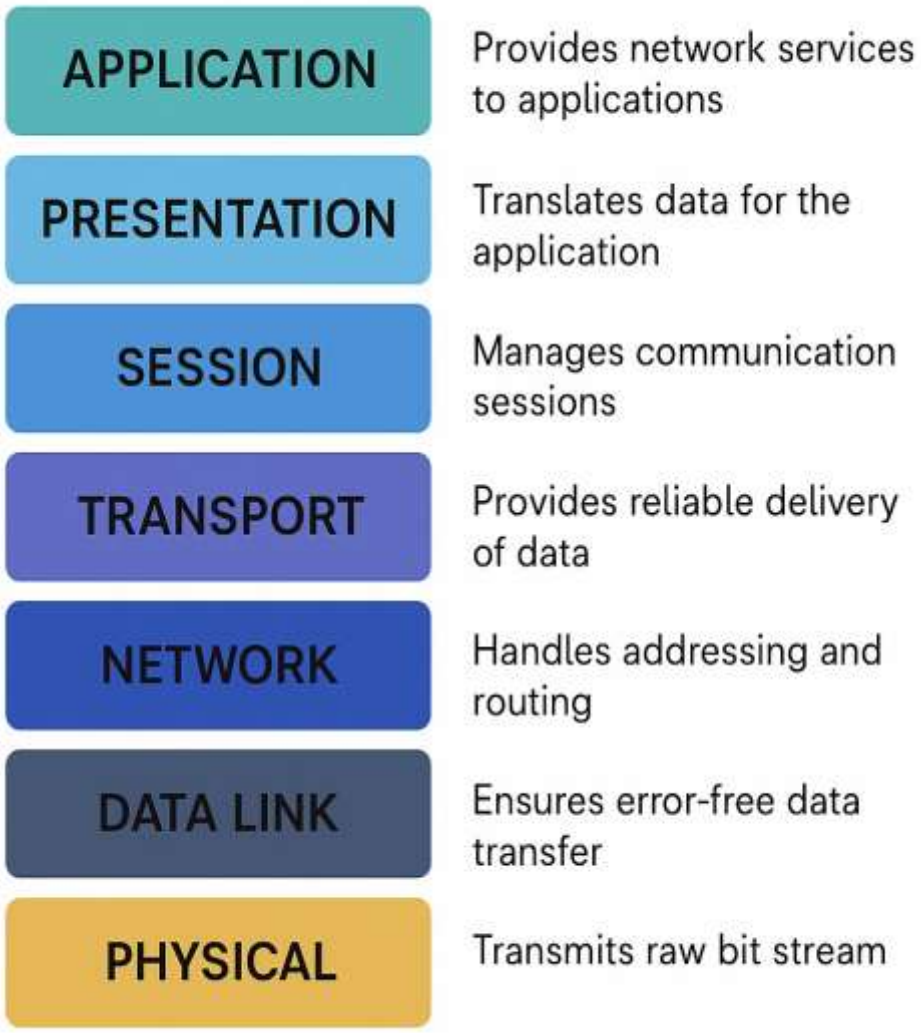
6. Presentation Layer

- This layer translates the data into a format that the receiving device can understand.
- It also handles data encryption and compression.

7. Application Layer

- This is the top layer and is closest to the user.
- It provides services like email, file transfer, and web browsing.

OSI Model Diagram:



---

7. **What is the full form of TCP/IP? And draw the TCP/IP protocol suit and write about each protocol?**

- TCP/IP stands for Transmission Control Protocol / Internet Protocol.
- It is a set of communication protocols used to connect devices on the Internet and other similar networks.
- TCP/IP defines how data should be packetized, addressed, transmitted, routed, and received.

Explanation of Each Layer in the TCP/IP Protocol Suite

1. Application Layer: Closest to the user.

Provides services and interfaces for applications.

Protocols:

- **HTTP (HyperText Transfer Protocol):** Used for web browsing.
- **FTP (File Transfer Protocol):** Used to transfer files.
- **SMTP (Simple Mail Transfer Protocol):** Used for sending emails.
- **DNS (Domain Name System):** Converts domain names into IP addresses.

2. Transport Layer: Responsible for reliable or fast delivery of data.

Ensures proper data transmission between two devices.

Protocols:

- **TCP (Transmission Control Protocol):** Provides reliable, connection-oriented transmission.
- **UDP (User Datagram Protocol):** Provides fast, connectionless communication.

3. Internet Layer: Handles addressing and routing of data across networks.

Ensures the packet reaches the correct destination.

Protocols:

- **IP (Internet Protocol):** Assigns IP addresses and routes data.
- **ICMP (Internet Control Message Protocol):** Used for error messages and diagnostics (like ping).
- **ARP (Address Resolution Protocol):** Maps IP addresses to MAC addresses.

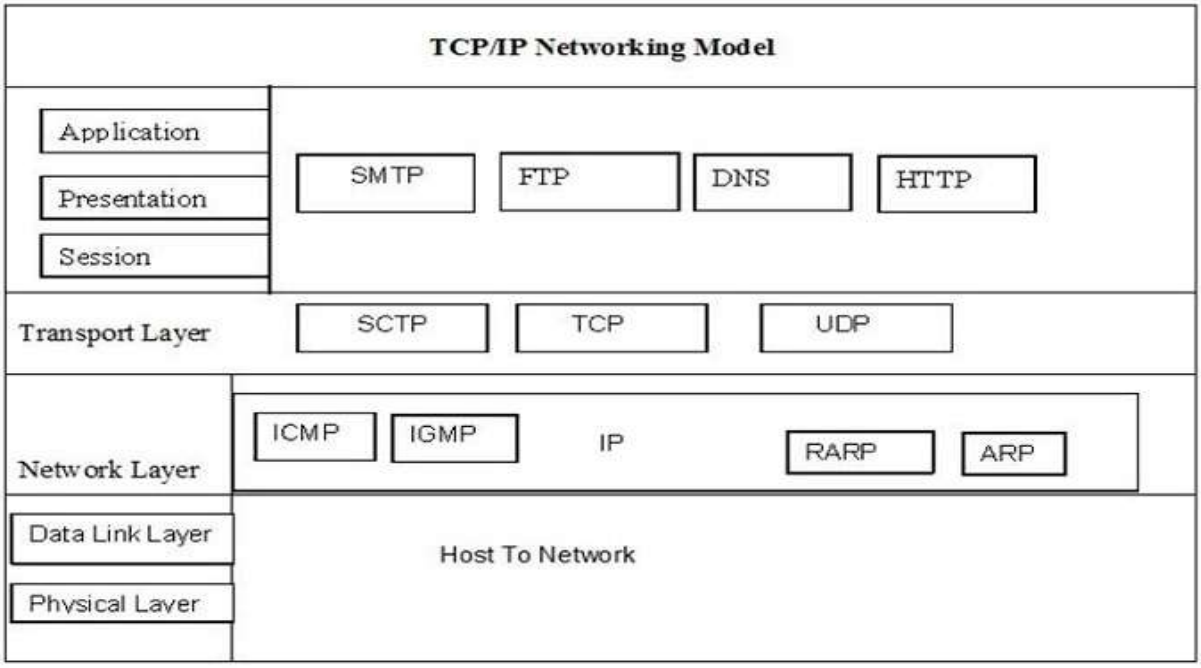
4. Network Access Layer (Link Layer): Deals with physical transmission of data over a network.

Controls how data is physically sent using hardware.

Technologies:

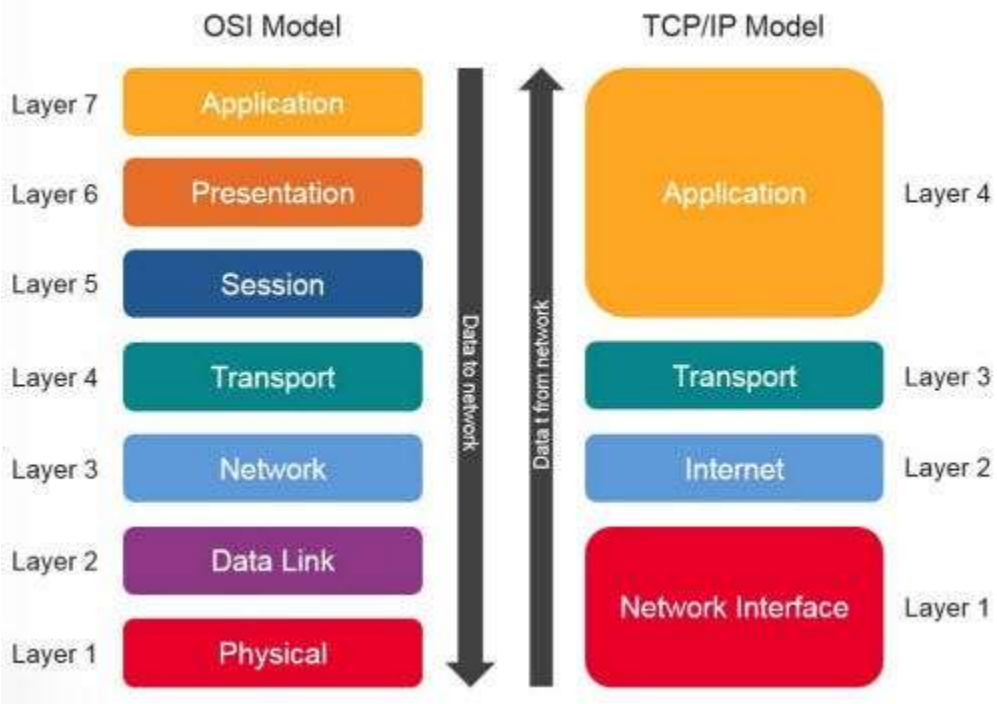
- **Ethernet:** Common wired network protocol.
- **Wi-Fi:** Wireless communication standard.
- **PPP (Point-to-Point Protocol):** Used for direct connections between two nodes.

TCP/IP Diagram:



---

8. Compare and contrast OSI and TCP/IP layers?



Comparison and Contrast of OSI and TCP/IP Models

Feature	OSI Model	TCP/IP Model
Full Form	Open Systems Interconnection	Transmission Control Protocol / Internet Protocol
Developed by	ISO (International Organization for Standardization)	DARPA (U.S. Department of Defense)
Year Introduced	1984	1970s
Model Type	Theoretical/Conceptual model	Practical/Implementation model
No. of Layers	7 layers	4 layers
Layers (Top to Bottom)	1. Application 2. Presentation 3. Session 4. Transport 5. Network 6. Data Link 7. Physical	1. Application 2. Transport 3. Internet 4. Network Interface
Function of Layers	Each layer has a distinct function (e.g., presentation handles encryption)	Layers are broader; some OSI layers are merged into one TCP/IP layer
Protocol Dependency	Protocol-independent; a generic model	Protocol-specific (built around TCP/IP protocols)
Use in Practice	Used as a reference model for teaching and design	Used for actual internet communication
Flexibility	Rigid boundaries between layers	More flexible and adaptable
Example Protocols	HTTP, FTP (Application); TCP, UDP (Transport); IP (Network)	Same, but organized under fewer layers

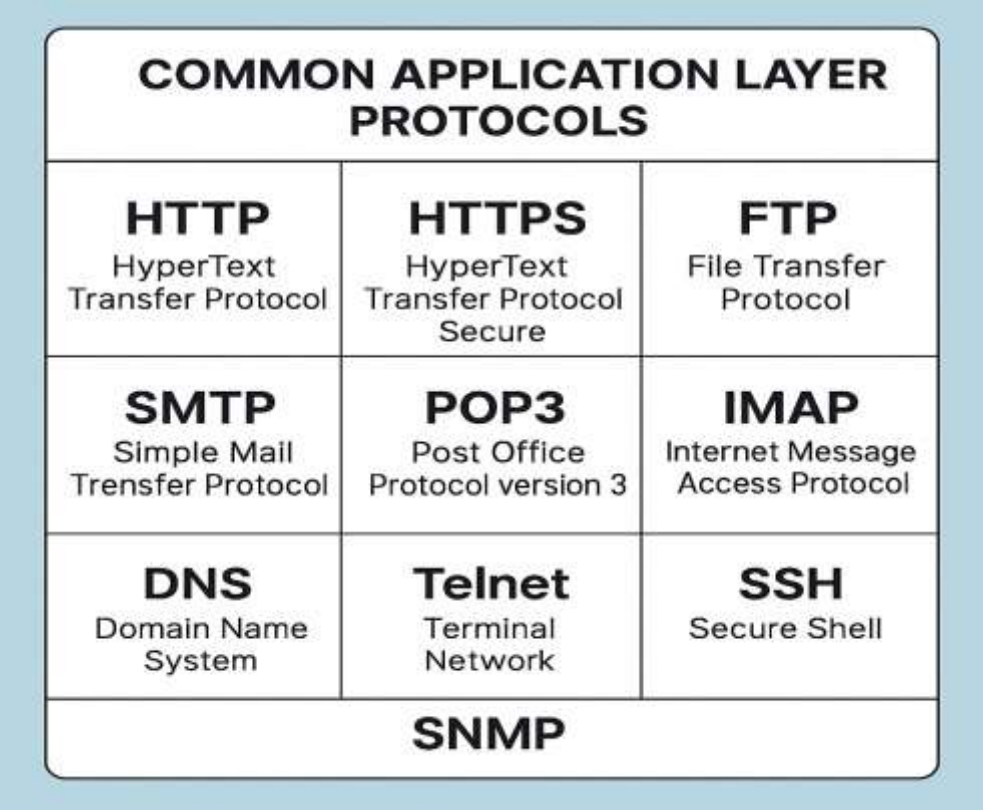
Key Points to Remember

- OSI is a conceptual model mainly used for understanding.
- TCP/IP is the real-world model used in internet communication.
- OSI has 7 layers; TCP/IP has 4 layers.
- Layers such as Presentation and Session in OSI are part of the Application layer in TCP/IP.

UNIT-II

1. **What are the common Application Layer protocols and explain them?**

   The Application Layer is the topmost layer (Layer 7) in the OSI model and directly interacts with the end-user. It provides services and network applications to users.



Explanation of Each Protocol:

1. **HTTP (HyperText Transfer Protocol)**
   - Used to load web pages through browsers.
   - Works on port **80**.
   - It is a **stateless** protocol (doesn't remember previous sessions).
2. **HTTPS (HTTP Secure)**
   - Same as HTTP, but with **encryption using SSL/TLS**.
   - Secures data like passwords and credit card details.
   - Works on port **443**.
3. **FTP (File Transfer Protocol)**
   - Transfers files between a client and a server.
   - Requires login (username & password).

- Uses ports **20 (data)** and **21 (control)**.

4. **SMTP (Simple Mail Transfer Protocol)**
    - Sends emails from client to server or server to server.
    - Only **sends**, not retrieves.
    - Works on port **25** (or 587 for secure).
5. **POP3 (Post Office Protocol v3)**
    - Downloads emails to the client and **removes them from the server**.
    - Good for offline access.
    - Works on port **110** (or 995 for secure).
6. **IMAP (Internet Message Access Protocol)**
    - Accesses and manages emails **directly on the server**.
    - Allows folder structure and multiple device sync.
    - Works on port **143** (or 993 for secure).
7. **DNS (Domain Name System)**
    - Resolves domain names (like google.com) to IP addresses (like 142.250.182.206).
    - Acts like the internet's phonebook.
    - Works on port **53**.
8. **Telnet (Terminal Network)**
    - Allows remote login to another computer using text-based commands.
    - Not secure; sends data in plain text.
    - Works on port **23**.
9. **SSH (Secure Shell)**
    - Secure version of Telnet using encryption.
    - Used by network administrators for remote system access.
    - Works on port **22**.
10. **SNMP (Simple Network Management Protocol)**
    - Used by network admins to **monitor and control network devices**.
    - Can collect performance data and detect network issues.
    - Works on port **161**.

---

2. Explain in detail about HTTP and show the structure of HTTP request and response?

- HTTP (Hypertext Transfer Protocol) is the protocol used for communication between web browsers and web servers.

- It is used to load web pages using hyperlinks, forms, images, and other resources.
- It is a stateless protocol, meaning it does not remember previous requests.
- It works over the client-server model: the client (browser) sends a request, and the server sends a response.

Structure of an HTTP Request: An HTTP request has three main parts:

1. Request Line:

- Contains the HTTP method (like GET or POST), the URL, and the HTTP version.
- Example: GET /index.html HTTP/1.1

2. Headers:

- Provide additional information about the request like browser type, accepted content, etc.
- Example: Host: www.example.com, User-Agent: Mozilla/5.0

3. Body (Optional): Used in methods like POST or PUT to send data to the server (such as form data).

Structure of an HTTP Response: An HTTP response also has three main parts:

1. Status Line:

- Contains the HTTP version, status code, and status message.
- Example: HTTP/1.1 200 OK

2. Headers

- Provide information about the server and the response.
- Example: Content-Type: text/html, Content-Length: 1024

3. Body: Contains the actual content (such as HTML, image, or JSON data).

Common HTTP Methods

- GET – Request data from the server.
- POST – Send data to the server.
- PUT – Update existing data.
- DELETE – Remove data from the server.

3. **How do you calculate the response time in persistent and non-persistent connections?**

Non-Persistent Connection

- A new TCP connection is opened for each HTTP request (like images, text, etc.).
- Each connection includes 2 RTTs (Round Trip Times) before receiving the object:
    - 1 RTT to establish the TCP connection.
    - 1 RTT for the HTTP request and first byte of response.

Total response time per object:

2 RTT + file transmission time

If a webpage has *n* objects, total time is roughly:

n × (2 RTT + transmission time)

Persistent Connection

- One TCP connection is reused to download multiple objects.
- Only one TCP setup is needed at the start.

For the first object:

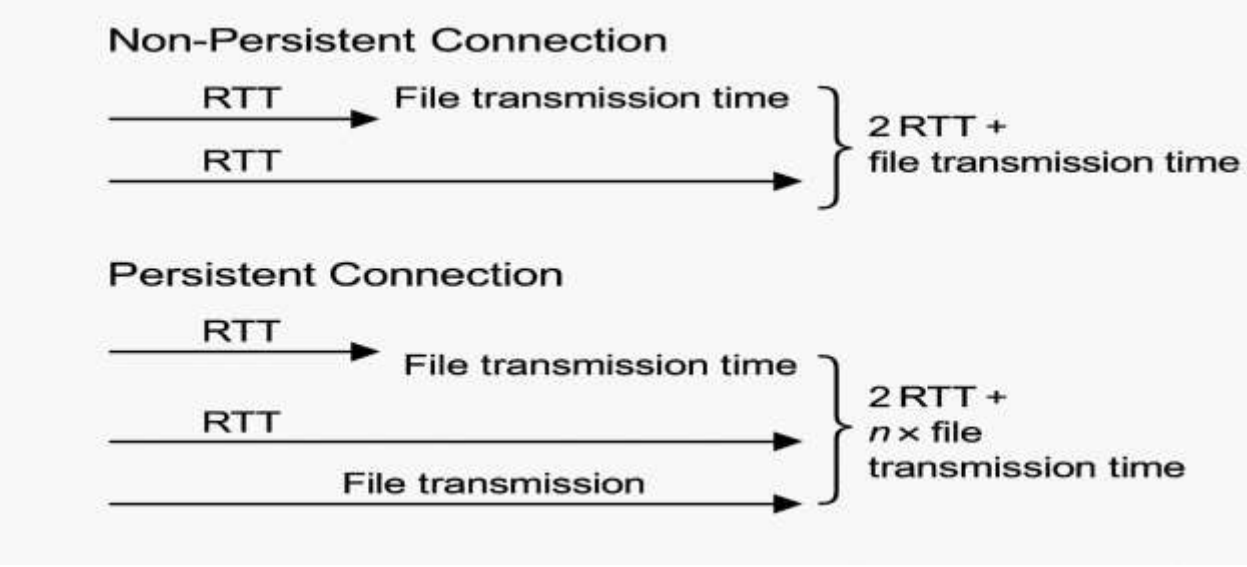
2 RTT + transmission time

For the remaining (n – 1) objects (if pipelining is used):

1 RTT + transmission time total

Total response time for *n* objects:

2 RTT + n × transmission time



4. **What is meant by FTP? State the features, architecture, working and modes of operations in FTP?**

FTP (File Transfer Protocol) is a standard network protocol used to transfer files between a client and a server over the Internet or a private network. It uses TCP for reliable data transfer and operates on port 21.

Features of FTP

- Transfers files between computers on a network
- Supports file upload and download
- Allows user authentication (username and password)
- Supports anonymous access if enabled
- Can resume interrupted downloads
- Can transfer files in binary or ASCII mode

Architecture of FTP

- FTP uses a client-server architecture:
- FTP Client: The user-side application that sends commands (e.g., FileZilla, WinSCP)
- FTP Server: Receives commands and responds by transferring files
- Two connections are used:
- Control Connection: For commands and responses (uses port 21)
- Data Connection: For actual file transfer (uses port 20 or another port)

Working of FTP

1. The client connects to the server using the control connection
2. The user logs in using a username and password
3. The client sends a request (such as upload or download)
4. A separate data connection is made to transfer the file
5. Once the transfer is complete, the connection is closed

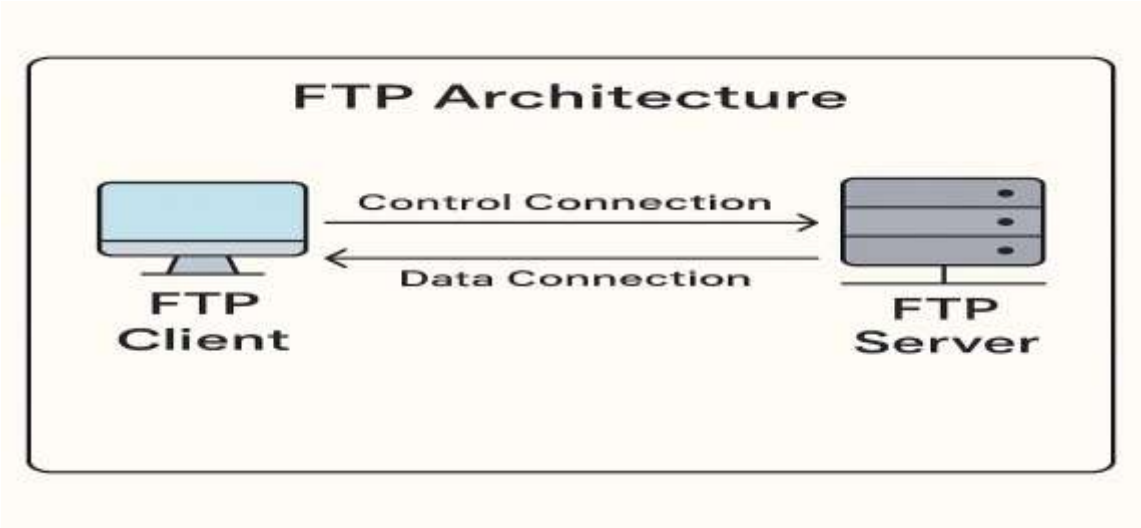
Modes of Operation in FTP

Active Mode

- The client opens a random port and sends it to the server
- The server connects back to the client's port for data transfer

Passive Mode

- The server opens a port and informs the client
- The client connects to that port for data transfer
- Used when clients are behind a firewall



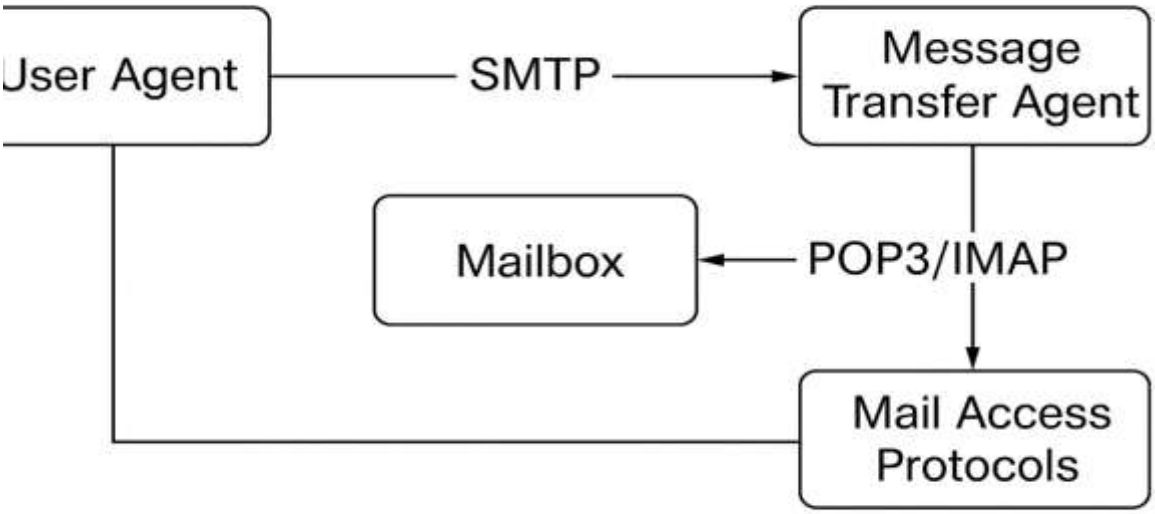
---

5. **What are the components in electronic mail, explain about each component?**

Electronic Mail (E-mail) is a method of exchanging messages and files over a computer network. It is one of the most common ways of communication on the Internet.

Components of E-mail System

There are five main components in an E-mail system:



1. User Agent (UA)

- Also known as an E-mail Client
- It is the software or application used by the user to read, write, send, and receive emails
- Examples: Gmail app, Microsoft Outlook, Thunderbird

2. Mailbox

- A storage space for incoming and outgoing messages
- Every user has a mailbox on the mail server
- The inbox holds received emails, while the outbox or sent folder stores sent messages

3. Message Transfer Agent (MTA)

- This is the mail server that helps in transferring emails between sender and receiver
- It works in the background to send, receive, and forward emails
- Examples: Sendmail, Postfix

4. Simple Mail Transfer Protocol (SMTP)

- A protocol used to send emails from the sender's device to the mail server and from one server to another
- Works on port 25
- It cannot be used to receive emails

5. Mail Access Protocols (POP3/IMAP)

- These are used to receive or read emails from the server:
- POP3 (Post Office Protocol 3): Downloads emails to the device and usually removes them from the server
- IMAP (Internet Message Access Protocol): Keeps emails on the server and syncs across multiple devices

---

6. Explain in detail about the role of each protocol in the email?

1. SMTP (Simple Mail Transfer Protocol)

- **What it does:** Sends emails from your device to the mail server and to another person's mail server
- **Used for:** Sending emails only
- **Direction:** From your email app to the mail server

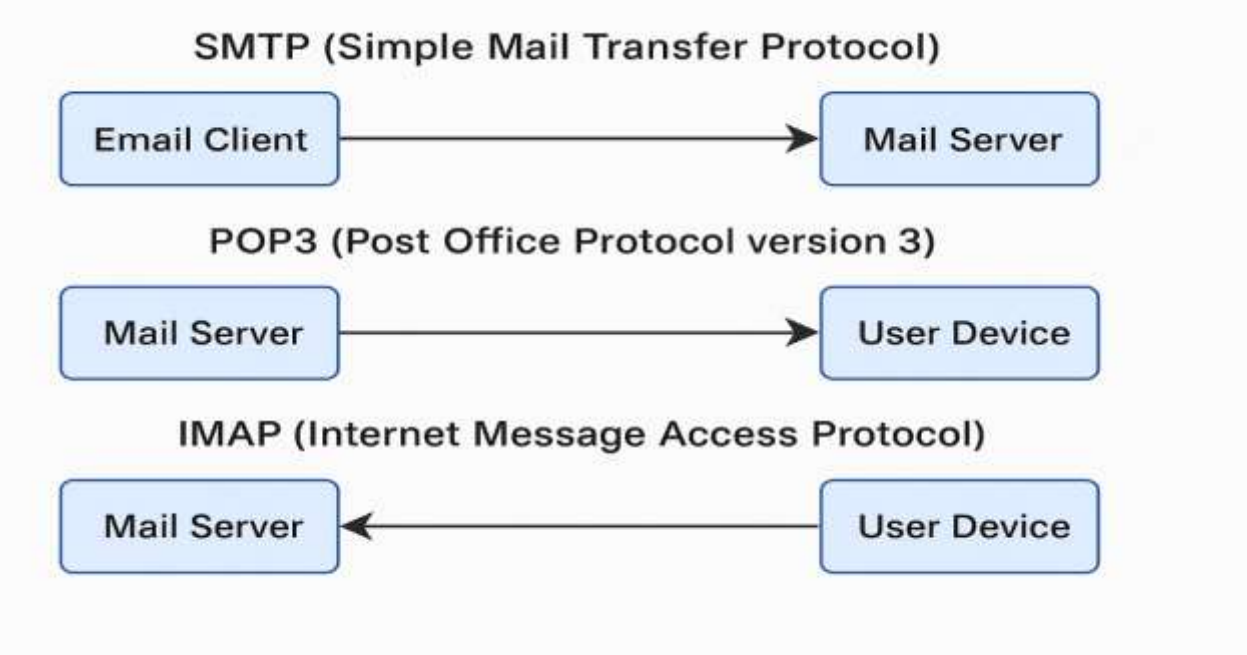
- **Example:** When you click "Send" in Gmail, SMTP takes your email to the other person's mail server

2. POP3 (Post Office Protocol 3)

- **What it does:** Downloads emails from the server to your device
- **Used for:** Receiving emails
- **Direction:** From mail server to your phone or computer
- **Note:** Emails are usually deleted from the server after downloading
- **Good for:** Reading emails offline

3. IMAP (Internet Message Access Protocol)

- **What it does:** Lets you read and manage emails directly on the mail server
- **Used for:** Receiving and syncing emails
- **Direction:** Between server and your device (both ways)
- **Good for:** Accessing the same email inbox from phone, tablet, and computer



---

7. **Explain the step by step process of email communication and show the email message format?**

   Email Composition:

   The user writes a message using an email client (like Gmail or Outlook).

   Sending the Email (SMTP)

   The email client sends the message to the mail server using SMTP protocol.

Mail Server to Mail Server Transfer

If the receiver uses a different mail server, SMTP forwards the email to that server.

Storing the Email

The receiver's mail server stores the email in their mailbox.

Receiving the Email (POP3/IMAP)

The receiver uses an email client to fetch or read the email using POP3 or IMAP protocol.

- POP3 downloads the email to the device.
- IMAP keeps the email on the server and shows it on the device.

Reading the Email

The receiver opens the email and reads it on their device.

Email Message Format

An email message is made up of two main parts:

Header – Contains control information

- From: Sender's email address
- To: Receiver’s email address
- Subject: Topic of the email
- Date: When the email was sent
- CC/BCC: Additional recipients

Body – Contains the actual message

Can include text, images, links, or attachments.

Example Email Format

```yaml
From: alice@example.com
To: bob@example.com
Subject: Meeting Reminder
Date: 30 June 2025

Hi Bob,

Just a reminder about our meeting tomorrow at 10 AM.

Best,
Alice
```

8. **What are the services provide by DNS? Explain how DNS will function?**
9. **How many types of resolution methods are there in DNS? With a neat sketch explain about each method?**
10. **What is meant by an attack? And what types of attacks are possible on DNS? And how do you overcome them? (Same As 8th)**

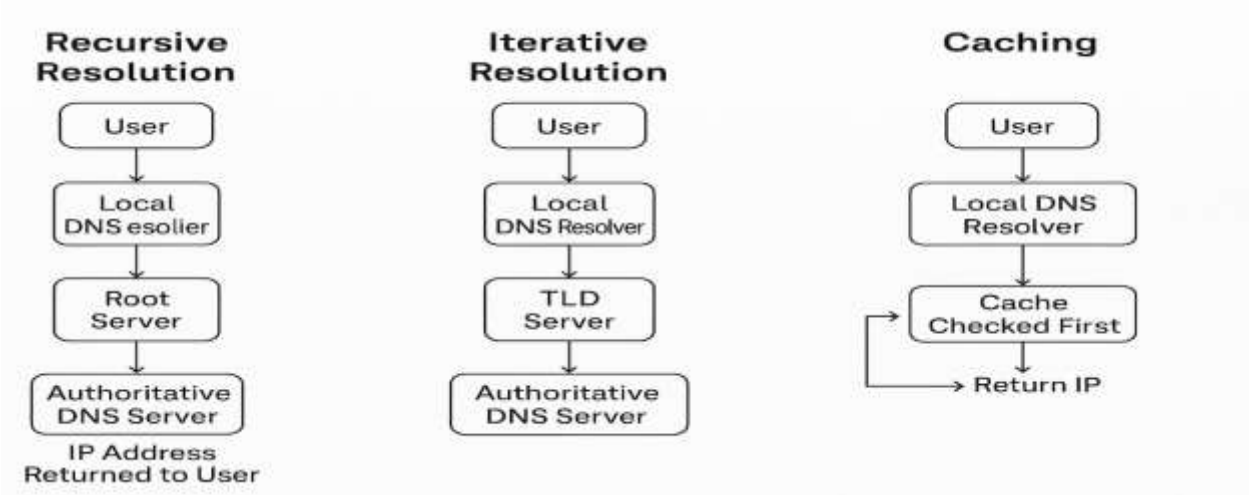
a) Services Provided by DNS:

- **Hostname to IP Address Mapping:** Converts human-readable domain names (e.g., google.com) into IP addresses.
- **Email Routing:** Maps domain names to mail server IPs (via MX records).
- **Load Distribution:** Uses multiple IPs for a single domain (round-robin DNS) to balance traffic.
- **Alias Creation:** Provides alternate names (CNAMEs) for the same host (e.g., www.example.com → example.com).
- **Reverse Mapping:** Resolves IP addresses back to domain names.

Functioning of DNS:

- User enters a domain name in the browser.
- Local DNS resolver checks its cache; if not found, it queries a DNS server.
- Query passes through:
  - Root DNS server → directs to TLD server (.com, .org)
  - TLD server → directs to authoritative server
  - Authoritative server → responds with the IP address
- Resolver sends IP back to the browser, which connects to the destination server.

Types of DNS Resolution Methods (Easy Explanation)



1. Recursive Resolution

- The DNS resolver does all the work.
- It asks each server (root → TLD → authoritative) one by one and gives the final IP address to the user.
- Simple for users, mostly used by internet service providers.

2. Iterative Resolution

- The resolver asks one server at a time.
- Each server replies with the next server to contact.
- Resolver repeats this until it gets the final answer.

3. Caching

- Resolver saves the result of previous DNS lookups.
- If the same domain is asked again, it quickly returns the saved IP.
- Saves time and reduces network load.

What is Meant by an Attack?

An attack is a deliberate attempt to breach or disrupt a system, steal information, or compromise service availability.

Types of DNS Attacks and Prevention

1. DNS Spoofing / Cache Poisoning

What happens?

Attacker sends fake IP to DNS so user goes to a wrong (malicious) website.

Prevention: Use DNSSEC, clear DNS cache regularly.

2. DDoS (Distributed Denial of Service)

What happens?

DNS server is flooded with too many requests and crashes.

Prevention: Use rate limiting, load balancers, and Anycast to manage traffic.

3. DNS Tunneling

What happens?

Hackers steal or send data secretly through DNS queries.

Prevention: Watch for suspicious DNS activity, block unknown domains.

4. Man-in-the-Middle (MitM)

What happens?

Attacker sits between user and DNS, changes the DNS reply.

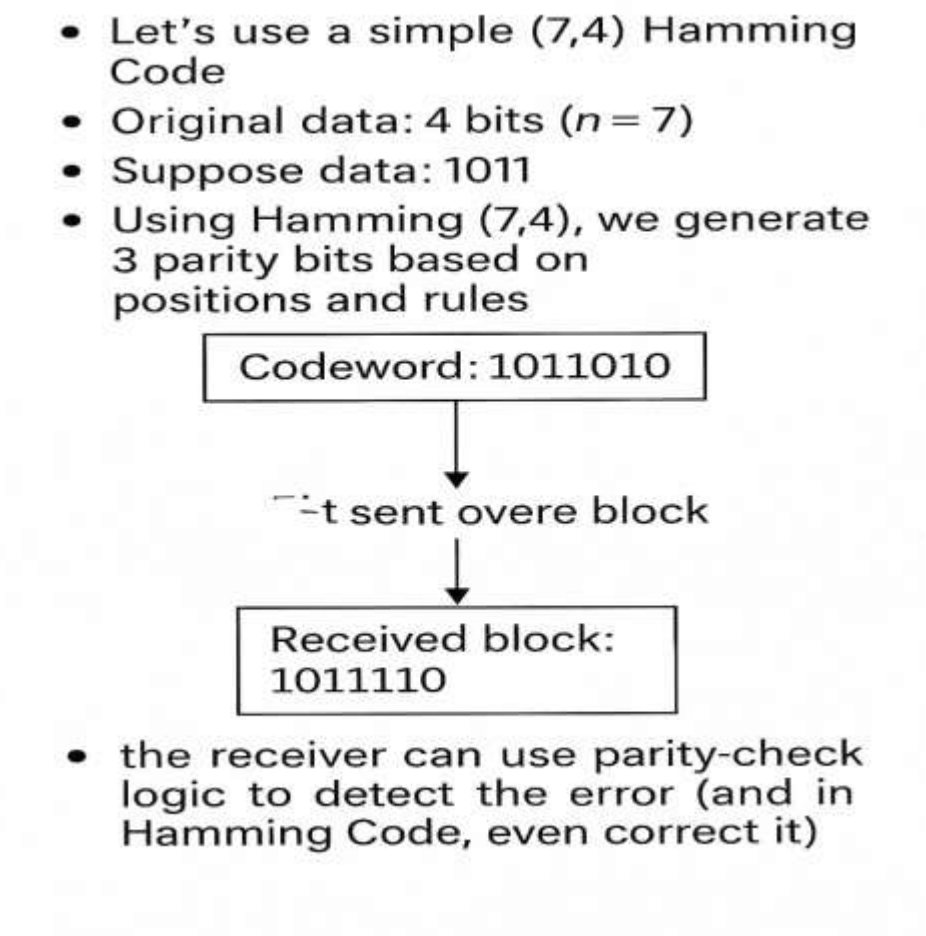
Prevention: Use DNS over HTTPS (DoH) or DNS over TLS (DoT) to encrypt DNS requests.

UNIT-III

1. **Explain how block coding and checksum methods are used for error detection in data communication. Illustrate with suitable examples?**

   Block Coding:

   - Block coding (also called Linear Block Coding) is an error detection and correction technique that adds redundant bits (called parity bits) to the original data bits to form a codeword.
   - Each block of data (k bits) is encoded into a larger block (n bits).
   - The difference (n - k) is the number of redundancy bits used to detect (and sometimes correct) errors.
   - **Example:**



Checksum:

- A checksum is a value derived from the sum of the data segments.
- The sender calculates this checksum and sends it along with the data.
- The receiver recalculates the checksum to verify data integrity.

Steps:

1. Divide data into fixed-size segments (e.g., 8 or 16 bits).
2. Add all segments (using 1's complement arithmetic).
3. Take the 1's complement of the sum → this is the checksum.
4. Send data + checksum.
5. Receiver adds all received segments (including checksum). If the result is all 1s, no error is detected.

- **Example:**

- Word1: 01010151
  Word2: 01100116
  Word3: 01111100
  = 00110111
  Step 2: Add them. (4ərrer [–with disciar-ding carry and using 1's complement addition)
  11001000
  Step 2: 1-s complement of sum =
  11001000 This is the chechsum
- Receiver Side:
  - Add all 4 words.
  - If the result is 111111111, then no error.
  - If, not, error is detected

2. **How do you classify flow control and error control protocols? Explain Go-Back-N ARQ? Describe the importance of flow control and error control in data communication?**

In data communication, **flow control** and **error control** are two important mechanisms that ensure reliable and efficient transmission of data between sender and receiver.

Flow Control:

- Flow control prevents the sender from overwhelming the receiver with too much data at once.
- It ensures that the sender sends data at a rate the receiver can handle.

Two common types:

- **Stop-and-Wait** – Sender sends one frame and waits for acknowledgment.
- **Sliding Window** – Allows multiple frames to be sent before needing acknowledgment.

Error Control:

- Error control ensures that data is delivered accurately and without corruption.
- It detects and corrects errors that occur during transmission.

Common techniques: Automatic Repeat reQuest (ARQ) protocols like:

- **Stop-and-Wait ARQ**
- **Go-Back-N ARQ**
- **Selective Repeat ARQ**

Go-Back-N ARQ (Error Control Protocol)

- A type of sliding window ARQ protocol.
- The sender can send **N frames** without waiting for individual acknowledgments.
- If an error is detected in one frame (say frame 3), the receiver **discards that frame and all after it**, even if some are correct.
- The sender then **goes back and retransmits from the errored frame** (frame 3 onward).
- **Example:**
    - Sender sends frames: 1, 2, 3, 4, 5
    - If frame 3 is lost or damaged, the receiver sends a NACK or ignores frame 3 and all later ones.
    - Sender then re-sends: 3, 4, 5 again.

Importance of Flow and Error Control

- **Flow control** avoids buffer overflows and keeps sender/receiver in sync.
- **Error control** ensures data accuracy and reliability, preventing communication breakdowns.
- Together, they provide **efficient and error-free communication** in networks, which is essential for applications like:
    - Video calls
    - Emails
    - File transfers
    - Online streaming, etc.

---

3. **Discuss about Stop-and –wait protocol for a noiseless channel?**
4. **Discuss about Stop-and-wait ARQ for a noisy channel?**

(3A) Stop-and-Wait Protocol (for a Noiseless Channel):

- This is the simplest flow control protocol.
- It works under the assumption that no errors occur during data transmission.

Working:

- Sender sends one frame at a time.
- Then it waits until it receives an acknowledgment (ACK) from the receiver.
- After receiving ACK, sender sends the next frame.
- Receiver processes the frame and sends ACK back.

Key Points:

- Very simple and reliable (in noiseless channel).
- Slow, because only one frame is sent at a time and sender waits idly.
- No need for frame numbering or retransmission logic.

(4A) Stop-and-Wait ARQ (for a Noisy Channel):

- This is an error control version of the stop-and-wait protocol.
- It assumes that frames or ACKs may get lost or damaged due to noise in the channel.

Working:

- Sender sends one frame and starts a timer.

- If ACK is received before timer expires → send next frame.
- If no ACK or negative ACK (NAK) is received → resend the same frame.
- Frames are numbered 0 and 1 alternately to avoid confusion of duplicates.

Key Features:

- Handles lost frames or corrupted ACKs.
- Uses timers and retransmission.
- Ensures reliable delivery even in the presence of noise.

---

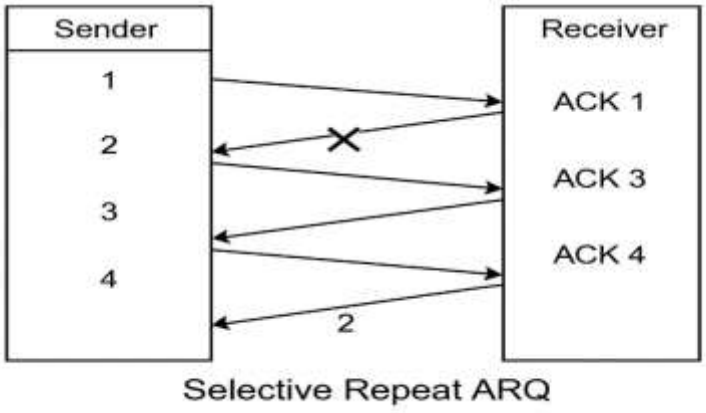
5. **Discuss about Selective repeat ARQ for a noisy channel?**

Selective Repeat ARQ (Automatic Repeat reQuest)

- Selective Repeat ARQ is an advanced error control protocol used in noisy communication channels.
- It improves efficiency by resending only the specific frames that are lost or damaged — not all subsequent frames.

How It Works:

- The sender sends multiple frames at a time (based on a sliding window).
- Each frame has a unique sequence number.
- The receiver checks each frame and sends an ACK for each correct frame.
- If a frame is missing or damaged, the receiver:
  - Sends a NAK (Negative ACK), or
  - Just doesn't acknowledge it.
- The sender selectively resends only those specific frames that were not acknowledged.



Selective Repeat ARQ

Stender sends |: 2, "0110116 "ACK 3;" and Chekchsum

6. **Describe the importance of flow control and error control in data communication?(Same as 2Q)**

---

7. **Illustrate three Random Access Protocols (e.g., ALOHA, CSMA, CSMA/CD) with suitable examples related to Multiple Access scenarios?**
    - These protocols are used when many devices share the same network, and they want to send data randomly.
    - But sometimes collisions happen — two devices send data at the same time. So these methods help manage that.

1. ALOHA:

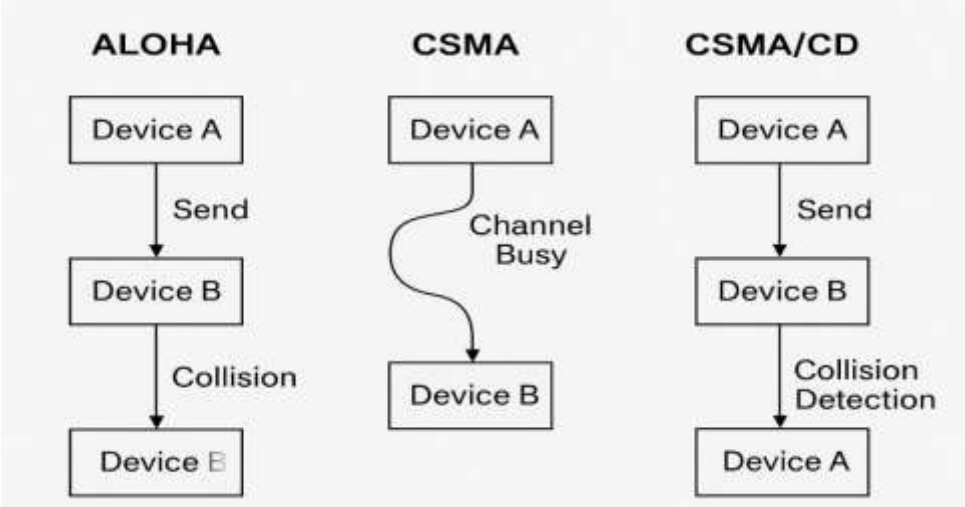
- Devices send data whenever they want.
- If two devices send at the same time → collision happens.
- They wait randomly and try again.
- **Example:** Mobile phones sending data to a tower. If two send at once → they retry after some time.
- Slotted ALOHA divides time into slots to reduce collisions**.**

2. CSMA (Carrier Sense Multiple Access):

- Before sending, the device checks if the channel is busy.
- If it's busy → it waits
- If it's free → it sends
- **Example:** In a Wi-Fi network, your laptop waits until the network is free to upload a file.

3. CSMA/CD (with Collision Detection):

- Same as CSMA, but it also listens while sending.
- If a collision happens → it stops immediately, sends a jam signal, and retries later.
- **Example:** In wired Ethernet, if two PCs send data at once, they both detect the collision and stop.



UNIT-IV

1. **Discuss about the overview of Network Layer?**

Network Layer (Layer 3 of OSI Model):

In the OSI model, there are 7 layers. The Network Layer is the third layer, and it plays a key role in delivering data across different networks.

Where It Sits:

- It is above the Data Link Layer (Layer 2)
- It is below the Transport Layer (Layer 4)

Main Functions of Network Layer:

- **Routing:** Finds the best path for data to travel from sender to receiver using routers.
- **Logical Addressing:** Assigns IP addresses to devices for identification on a network.
- **Packet Forwarding:** Sends data in packets from one device to another, across multiple routers.
- **Fragmentation & Reassembly:** Splits big packets if needed and reassembles them at the destination.

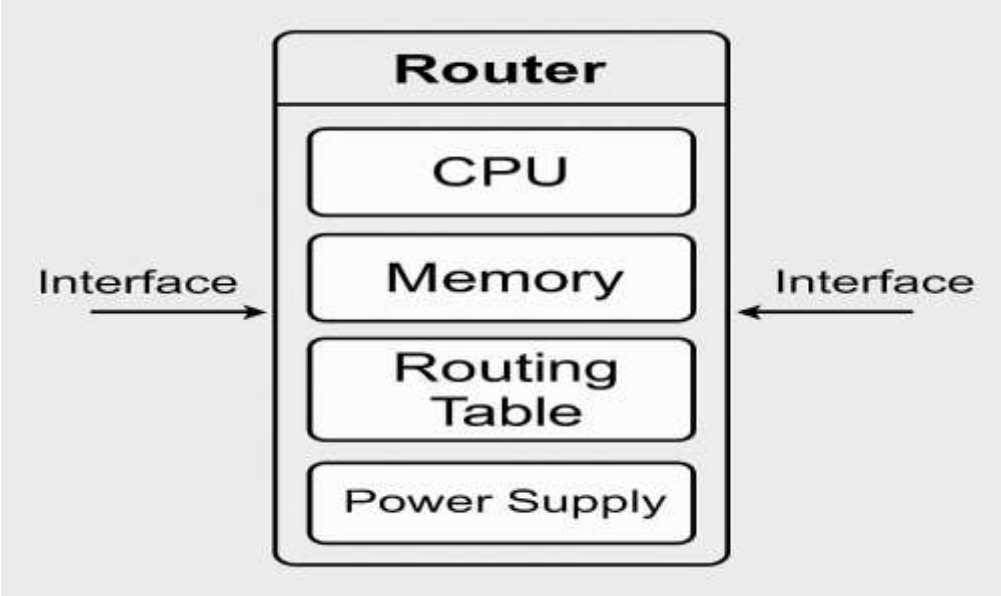
Key Devices: Routers work at this layer to forward packets between networks.

- **Common Protocols:**
- **IP** (IPv4/IPv6)
- **ICMP** (used in ping)
- **ARP/RARP** (for address resolution)

- **Conclusion:** The Network Layer makes sure data is delivered from one network to another using IP addresses and routing. It is essential for Internet communication.

---

2. **Explain about the major components of a Router?**
    - A router is a device that connects different networks and forwards data between them.
    - It uses IP addresses to decide where to send the data.
    - The major components of a router are:



1. Central Processing Unit (CPU)

- The brain of the router.
- It controls all activities and runs the router's operating system.
- Processes routing decisions and handles configuration commands.

2. Memory: Routers use different types of memory:

- **RAM (Random Access Memory):** Stores routing tables, running configuration, and temporary data.
- **ROM (Read-Only Memory):** Stores the boot-up instructions.
- **NVRAM (Non-Volatile RAM):** Stores the startup configuration.
- **Flash Memory:** Stores the router's operating system (like Cisco IOS).

3. Interfaces / Ports: These are the physical connections used to connect the router to other networks/devices.

Types include:

- **Ethernet Ports** (LAN/WAN)
- **Console Port** (for local setup and configuration)
- **Serial Ports** (for older network setups)

4. Routing Table

- A table stored in memory that shows the best path to reach different network destinations.
- Updated automatically using routing protocols (e.g., OSPF, RIP) or manually.

5. Power Supply: Provides the necessary power for the router to function.

---

3. **Describe the structure of an IPv4 packet. What are the purposes of key fields such as TTL, Header Checksum, and Fragmentation?**

IPv4 Packet Key Components

An IPv4 packet consists of two main parts:

1) TTL (Time to Live)

- **Purpose:** Prevents infinite looping of packets in the network.
- **How it works:** Each router that forwards the packet decrements the TTL by 1. If **TTL = 0**, the packet is discarded.
- **Example:** If a packet gets stuck in a routing loop, TTL ensures it doesn't circulate forever.

2) Header Checksum

- **Purpose:** Ensures the integrity of the IPv4 header only (not the payload).
- **How it works:**
  - Sender calculates a checksum over the header fields.
  - The receiver recalculates and verifies it.
  - If there is a mismatch → packet is dropped.
- **Helps in:** Detecting accidental errors during transmission.

3) Fragmentation Fields (Identification, Flags, Fragment Offset)

- **Purpose:** Allow large packets to be split into smaller pieces (fragments) to pass through networks with smaller **MTUs** (Maximum Transmission Units).
- **Fields:**
  - **Identification:** Same for all fragments of a packet.
  - **Flags:** Includes the **"More Fragments"** bit.

- **Fragment Offset:** Indicates where this fragment belongs in the original packet.
- **Reassembly:** Done at the destination using these fields.

Summary:

- **TTL** prevents packet looping.
- **Header Checksum** protects against corrupted headers.
- **Fragmentation Fields** enable transmission across networks with different size limits.

---

4. State the differences between IPv4 and IPv6?

Differences Between IPv4 and IPv6	IPv4	
Address Size	32-bit (like 192.16.1.1)	128-bit (liake 2001.08.1)
Number of Addresses	Around 4 billlion	Vory large (almost
Format	Uses numbers and dots	Has built-in IPSec (more secure)
Security	No built-in security	Has built-in IPSec (more secure)
Configuration	Supports broadcasting	Can auto-configure itself
Broadcasting	Supports broadcasting	Uses multicasting (no broadcast)
Speed & Performance	Needs NAT due to limited IPs	No need for NAT (more IPs)

---

5. Explain the working of the Link State Routing Algorithm. How does Dijkstra algorithm help in finding the shortest path?

Link State Routing Algorithm:

- The **Link State Routing Algorithm** is used by routers to dynamically determine the shortest path to all other routers in a network.
- It is the foundation of protocols like **OSPF (Open Shortest Path First)**.

Working of Link State Routing:

- Each router **discovers its neighbors** and learns their IP addresses.
- **Measures the cost** (delay, bandwidth, etc.) to each of its neighbors.

Builds a Link-State Packet (LSP) containing:

- **ID of the router**
- **List of directly connected neighbors**
- **Cost to each neighbor**
- **Floods LSPs** to all other routers in the network.
- Each router builds a **complete network topology map** from all received LSPs.

Dijkstra's Algorithm – Finding the Shortest Path

- Dijkstra's algorithm helps determine the most efficient path from one node (router) to all other nodes in the network.

Steps of Dijkstra's Algorithm:

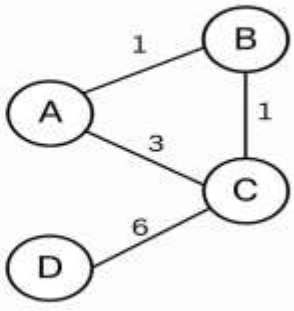
1. **Initialize:**
    - Set the distance to source node as **0** and all others as ∞.
    - Mark all nodes as **unvisited**.
2. **Choose the unvisited node** with the smallest distance.
3. **Update the distance** to its neighbors if a shorter path is found.
4. **Mark the node as visited.**
5. **Repeat steps 2–4** until all nodes are visited.

Advantages of Link State Routing:

- Fast convergence
- Accurate and complete network knowledge
- Less chance of routing loop

Example:

- Let's say Router A wants to find the shortest path to all other routers:
    - It knows the costs to its direct neighbors (B, C).
- It receives LSPs from all routers and constructs the full graph.
- Then, it applies Dijkstra's algorithm to calculate:
    - Cost to B = 2
    - Cost to C = 5
    - Cost to D = 3((via B), etc.



6. **Compare and contrast Link State and Distance-Vector Routing algorithms. Discuss their practical advantages and limitations in dynamic network environments?**

1. Working Principle:

- **Distance Vector Routing:** Each router shares its routing table with directly connected neighbors periodically.
  It calculates paths using the *Bellman-Ford algorithm* based on hop count or other metrics.
- **Link State Routing:** Each router discovers the full network topology by exchanging *Link State Advertisements (LSAs)* with all routers.
  It then computes the shortest path using *Dijkstra's algorithm*.

2. Information Sharing:

- **Distance Vector:** Shares minimal info – only with neighbors.
- **Link State:** Shares detailed topology – flooded to all routers in the area.

3. Convergence and Stability:

- **Distance Vector:** Slow convergence. Vulnerable to *routing loops* and *count-to-infinity* problems.
- **Link State:** Fast convergence. Network changes are quickly propagated and recomputed, reducing the risk of loops.

4. Resource Usage:

- **Distance Vector:** Low CPU, memory, and bandwidth usage.
- **Link State:** High resource consumption due to full topology storage and computation.

5. Practical Advantages in Dynamic Environments:

Feature	Distance Vector	Link State
Setup Complexity	Simple	Complex
Loop Avoidance	Needs extra techniques (e.g., Split Horizon)	Naturally loop-free
Convergence Speed	Slower	Faster
Scalability	Limited to small networks (e.g., RIP)	Scales well in large networks (e.g., OSPF)
Fault Recovery	Delayed	Quick adaptation to link failures

Practical Advantages: Distance Vector vs Link State

Feature	Distance Vector	Link State
Implementation	Easy to implement	Faster convergence
Resource Usage	Requires less CPU and memory	More reliable and loop-free
Best Suited For	Small or simple networks	Large or changing networks

Limitations in Dynamic Networks

Routing Type	Limitations
Distance Vector	Slow to converge in big networks Routing loops can occur (count-to-infinity problem) Poor handling of frequent changes
Link State	Needs more memory and CPU Complex to set up and manage More control traffic during major changes

---

UNIT-V

1. **Describe how Cyclic Redundancy Check (CRC) works for error detection. Provide a detailed example and verify the result. Workout some problems.**

1. Introduction to CRC

Cyclic Redundancy Check (CRC) is an error-detection technique used in data communication. It treats data as a binary number and appends extra bits (called the CRC checksum) to detect errors during transmission. The method is based on binary division using modulo-2 arithmetic.

2. CRC Working Steps

Sender Side:

- The sender chooses a generator polynomial (divisor) G(x).
- Appends n–1 zeros to the data (n is the degree of G(x)).
- Performs binary division (modulo-2) of the extended data by the generator.
- The remainder (CRC bits) is appended to the original data and transmitted.

Receiver Side:

- The receiver performs the same division on the received frame.
- If the remainder is zero, the data is assumed to be correct.
- If the remainder is non-zero, it indicates transmission errors.

Step 1: Append zeros to data:

```
  1 1 0 1
-------
  1 0 0 0
```

Step 2: Divide using modulo-2 (like XOR):

```
          1 0 1 1
       -----------
  X11  | 1 1 0 0
       |     ×
  ←X   |   X O R  ← XOR
       -----------
         1 1 0 1
           ×   ↓
       -----------
         0 1 1 0  ← Remainder (CRC bits)
```

Step 3: Append remainder to original data:

```
  1 1 0 1
  0 1 1 0
-------
Final Transmitted Datata
= 1101 0110
```

4. Receiver Side Verification

- **Received data:** 11010110
- **Perform division** by **1011**. If remainder is **0** → No error.
- (If you divide **11010110** by **1011**, you will get a remainder of **0**, confirming no error.)

5. Conclusion: CRC is a powerful and efficient method for detecting errors in digital communication. It can detect common types of errors like:

- Single-bit errors
- Burst errors
- Double-bit errors

---

2. **What is CSMA/CA, and how does it improve network performance in wireless communication?**

1. Definition: CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is a network access method used in wireless communication (e.g., Wi-Fi/IEEE 802.11) to avoid data collisions before they occur, improving overall network reliability.

2. Why Collision Avoidance (CA) is Needed in Wireless: Unlike wired networks, wireless devices cannot detect collisions effectively due to:

- Signal interference
- Hidden node problem (one device is out of range of another)

3. Working of CSMA/CA:

- Carrier Sensing: The device listens to the channel to check if it is idle.
- Backoff Timer: If the channel is busy, the device waits for a random backoff time to reduce the chance of collision when multiple devices want to send data.
- RTS/CTS (optional): Request to Send (RTS) and Clear to Send (CTS) packets are exchanged to reserve the channel before transmission, reducing collisions further.
- Data Transmission: If the channel is still clear, the device transmits data.
- Acknowledgement (ACK): The receiver sends an ACK to confirm successful reception.

4. How CSMA/CA Improves Wireless Network Performance:

- **Reduces Collisions:** Prevents simultaneous transmissions using carrier sensing and backoff timers.
- **Handles Hidden Terminal Problem:** RTS/CTS mechanism helps inform other devices not to transmit.
- **Improves Throughput:** Fewer retransmissions mean more efficient use of bandwidth.
- **Enhances Fairness:** All devices get a chance to access the channel.
- **Conclusion:** CSMA/CA is a key protocol for wireless communication that improves performance by avoiding rather than detecting collisions.

---

3. **Discuss about various controlled access protocols related to multiple access?**

In multiple access communication, when many devices share a single transmission medium (like a bus or wireless channel), controlled access protocols ensure that only one device sends data at a time, avoiding collisions.

Types of Controlled Access Protocols:

1. Reservation Protocol:

- Devices reserve the channel before transmitting.
- A control frame is used to request time slots.

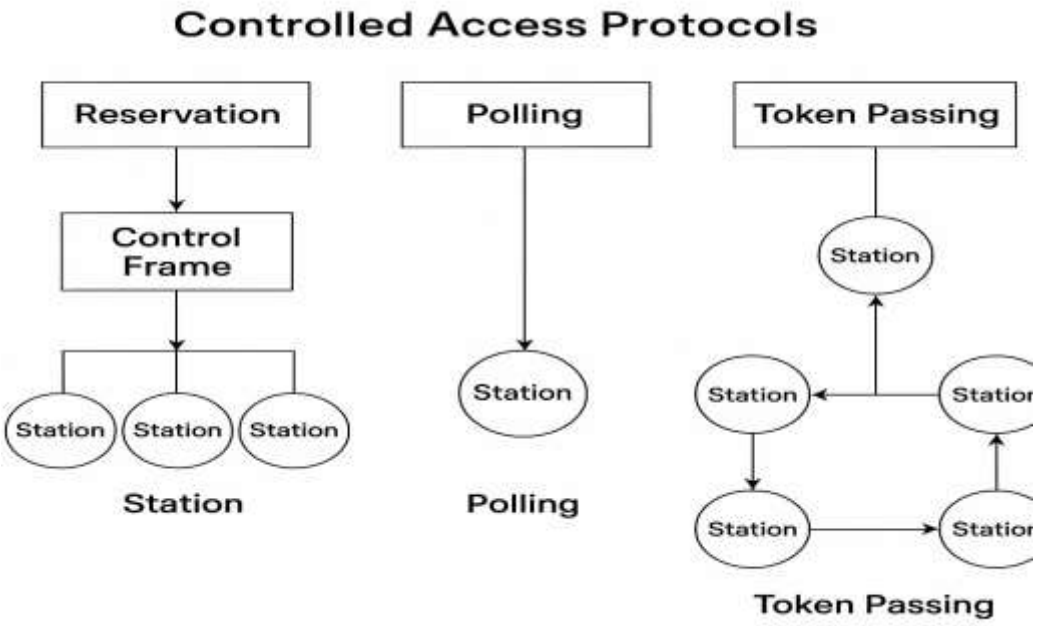
- The channel is divided into slots, and devices get permission to transmit in their reserved slots.
- **Example:** Used in satellite networks and wireless communication systems.

2. Polling Protocol:

- A central controller (master) asks each device (slave) one by one if it wants to send data.
- Only the device being polled can transmit.
- Reduces collisions but adds delay.
- **Example:** Used in mainframe-terminal networks or token-based LANs.

3. Token Passing Protocol:

- A special message called a token is passed from one device to another in a logical ring.
- Only the device holding the token can transmit.
- Prevents collisions and is efficient in high-load networks.
- **Example:** Used in Token Ring and FDDI networks.



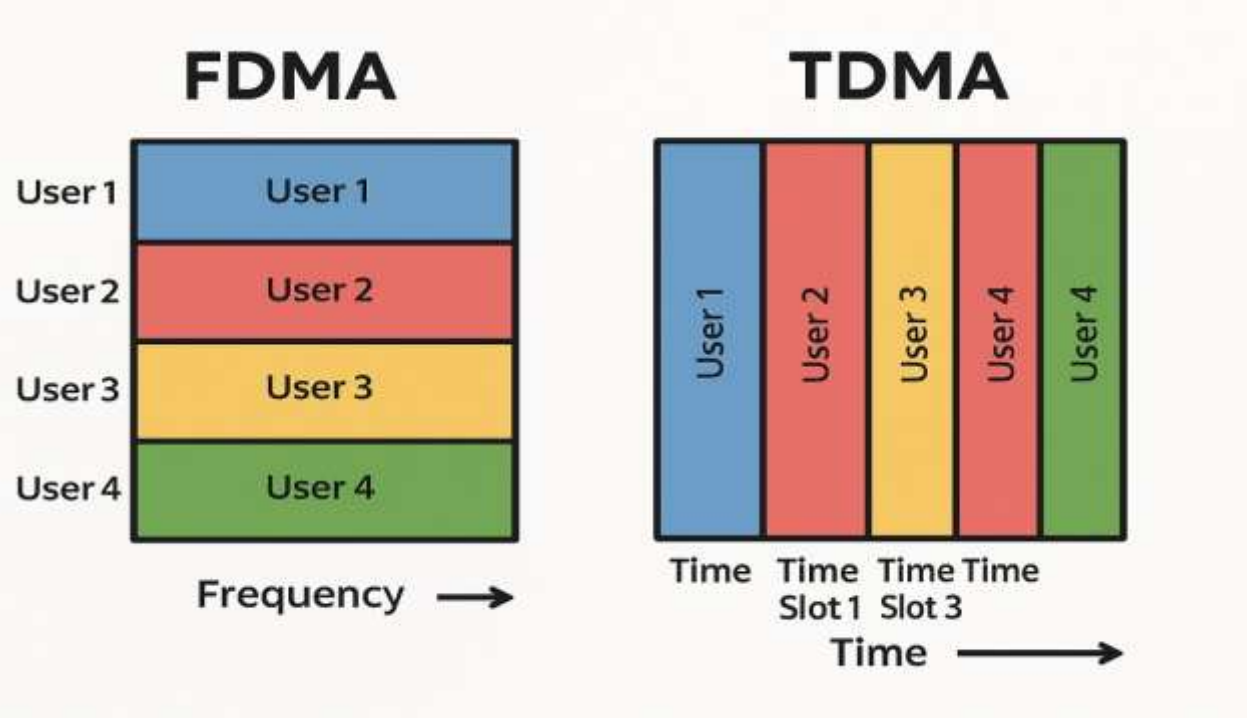
4. **Discuss about FDMA and TDMA?**

FDMA (Frequency Division Multiple Access)

- In FDMA, the available bandwidth is divided into separate frequency bands.
- Each user gets a dedicated frequency for the entire call or session.
- All users can send data at the same time, but on different frequencies.
- **Example:** Used in analog cellular systems, radio broadcasting.
- **Key Points:**
    - No interference as each user has a separate frequency.
    - Less efficient if the user is silent — the channel stays reserved.

TDMA (Time Division Multiple Access)

- In TDMA, users share the same frequency, but use it in different time slots.
- Each user is assigned a specific time slot in a repeating schedule.
- Users take turns to send their data.
- **Example:** Used in 2G GSM mobile networks, satellite systems.
- **Key Points:**
    - Efficient use of bandwidth.
    - Requires synchronization to avoid overlapping time slots.



5. **Explain the different types of errors that can occur in data transmission. How do error-detection and error-correction techniques help in reliable communication?**

1. Types of Errors in Data Transmission

In data communication, various types of errors can corrupt the data during transmission due to **noise**, **interference**, or **signal distortion**. The main types are:

a) Single-bit Error

- Only *one bit* of the data unit is altered.
- **Example:** 1011 → 111

b) Burst Error

- Two or more consecutive bits are altered.
- **Example:** 10110101 → 10000101 (a burst of 4 bits)

c) Random Errors

- Occur **sporadically**, often due to **electromagnetic interference** or **weak signals**.

d) Cross-talk Errors

- Caused by **signals from nearby cables** interfering.

e) Impulse Noise Errors

- **Short-duration noise pulses** that cause sudden bit flips.

2. Role of Error Detection Techniques

Error detection methods help **identify if an error** has occurred during transmission. Common techniques include:

- **Parity Bit:** Adds a bit to make the number of 1s even or odd.
- **Checksum:** Adds all data segments and sends the sum to detect errors.
- **Cyclic Redundancy Check (CRC):** Performs polynomial division to detect errors.
- **Hamming Code (with detection + correction):** Uses redundancy bits to detect and locate errors.

3. Role of Error Correction Techniques

Error correction methods not only **detect** but also **correct** errors without needing retransmission. These are especially useful in **satellite** or **noisy wireless** networks.

- **Forward Error Correction (FEC):** Adds enough redundancy to data so the receiver can fix errors
  (e.g., Hamming Code, Reed-Solomon Code)
- **Automatic Repeat Request (ARQ):** Uses **acknowledgments (ACK/NACK)** to request retransmission when errors are detected.

4. Importance in Reliable Communication

- **Data Integrity:** Ensures received data matches sent data.
- **Reduced Data Loss:** Prevents corruption of important messages.
- **Efficiency:** Minimizes retransmissions and improves throughput.
- **Applicability:** Essential in real-time systems like **VoIP**, **video streaming**, and **mission-critical systems**.

---

6. **Explain the Taking-Turns Protocols used in multiple-access communication. How do polling and token-passing mechanisms ensure coordinated access?**

1. Introduction

- In multiple-access communication, multiple devices share a common communication medium.
- To avoid collisions and ensure fairness, Taking-Turns Protocols allow devices to take turns accessing the medium in an organized manner.
- Two common methods under this category are Polling and Token Passing.

2. Taking-Turns Protocols

- These protocols divide channel access sequentially among the nodes, ensuring only one device transmits at a time, thus preventing collisions and improving efficiency.

3. Polling Mechanism

Centralized control: A master station or controller manages communication.

- It sends a poll to each node in a fixed order asking if it wants to transmit.
- If the polled node has data, it sends it; otherwise, the master moves to the next node.

Advantages:

- Efficient in low-load networks.
- No collisions due to central control.

Limitations:

- Delay increases with the number of nodes.
- Master failure halts communication.

4. Token Passing Mechanism

- A logical token (a small data frame) circulates among all devices in a predefined order.

- Only the device holding the token can transmit data.
- After transmission, the token is passed to the next device.

Advantages:

- Prevents collisions completely.
- Fair access to the medium.

Limitations:

- Token loss or duplication requires recovery mechanisms.
- Token-passing overhead can cause delays in low-load scenarios.

5. Coordinated Access: Both methods ensure orderly access to the shared channel

- Polling relies on a controller to decide whose turn it is.
- Token passing lets devices decide based on who holds the token.

*****