Computer Networks

INDEX

 **Important topics**

Introduction

1. **What is a computer network?**

   A computer network is a group of interconnected nodes (devices) that share resources and information via communication links.

2. **What is a node?**

   - A node is a device capable of sending and receiving data.

   End nodes: PC, printer, server, smartphone.
   Intermediary nodes: Router, Bridge, Modem.

3. **What is a communication link?**

   A communication link is a medium to transfer data.

   Types
   - Wired (coaxial cables)
   - Wireless (Wi-Fi, Bluetooth, etc.)

4. **What are the uses of a computer network?**

   - Communication
   - Resource sharing
   - Remote access
   - Collaboration
   - EEE stands for Entertainment, Education, E-commerce.

5. **What are characteristics of computer network?**

   - **Fault Tolerance** (Continue to work despite failures).
   - **Scalability** (Add devices without degrading performance).
   - **Quality of Service** (Network works smoothly without delays/interruptions).
   - **Security** (protection of data from unauthorized access, alteration, and disruption).

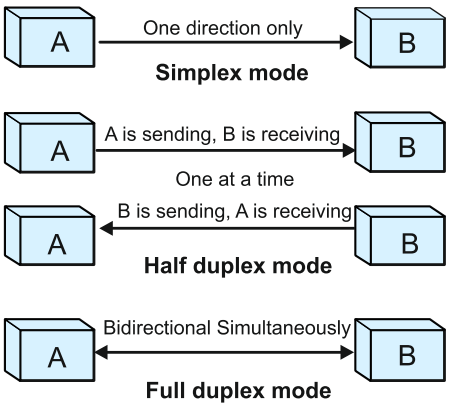
6. **What are the services offered by computer networks?**

   Email, instant messaging, video telephony, storage service, online games, web browsing, file sharing, VOIP (Voice over Internet Protocol), etc.

7. **What is a protocol?**

   - Set of **rules** for effective communication.
   - The sender and receiver follow a common set of rules to communicate.

8. Types of data flow



i. Simplex: One-way communication (e.g., keyboard to CPU)
ii. Half Duplex: Two-way communication, but only one direction at a time (e.g., walkie-talkie)
iii. Full Duplex: Two-way communication simultaneously (e.g., telephone)

9. What does a protocol generally contain?

Information about

- Sender
- Receiver
- Communication media

10. What are the elements of a protocol?

- Message Encoding
- Message Formatting and Encapsulation
- Message Timing
- Message Size
- Message Delivery Options

11. Types of Networks

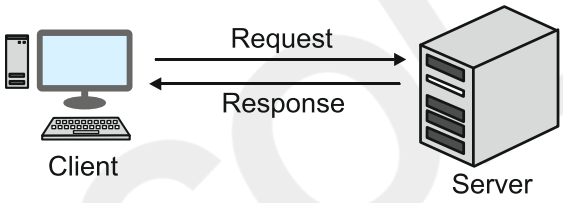
Networks	PAN	LAN	CAN	MAN	WAN
Full Form	Personal Area Network	Local Area Network	Campus Area Network	Metropolitan Area Network	Wide Area Network
Technology	Bluetooth, IrDa, Zigbee	Ethernet, wi-fi	Ethernet, Wi-Fi, Fiber optics	Metro Ethernet, MPLS	Leased line, MPLS, VPN, Satellite
Range	1 - 100 m	upto 2km	1 - 5 km	5 - 50 km	Above 50 km to global range
Transmission speed	Very high	Very high	High	Average	Low
Area	Within a room	Within office building	Within university, office, etc.	Within cities like Mumbai	Within countries
Ownership	Private	Private	Private	Private/ Public	Private/ Public
Maintenance	Very easy	Easy	Moderate	Difficult	Very difficult
Error Rate and Cost	Very low	Low	Moderate	High	Very high

12. Client-Server vs Peer-to-Peer (P2P) Network Models

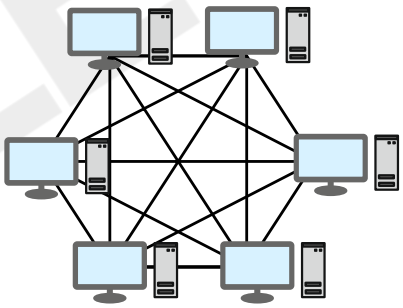
Client Server Network	Peer to Peer (P2P) Network
Centralized with one or more dedicated **servers**	**Decentralized** network where each system functions as both client and server
Server manages access and resources	Each user manages his or her own system" OR "their own systems
More secure with **authentication** and **access control**	**Less secure** due to lack of central control
Easily scalable for growing networks	**Limited scalability**, Becomes unstable as the number of peers increases
Higher cost due to server setup and maintenance	**Lower cost**, no dedicated server required
Used in **web apps, email systems, databases, enterprises**	Used in File Sharing, P2P Games, Blockchain, and Home Networks

Architecture

- **Client- Server**



- **Peer-To-Peer Network**



13. What is a topology?

Topology refers to the **layout and arrangement** of nodes in a computer network.

Categories of topology

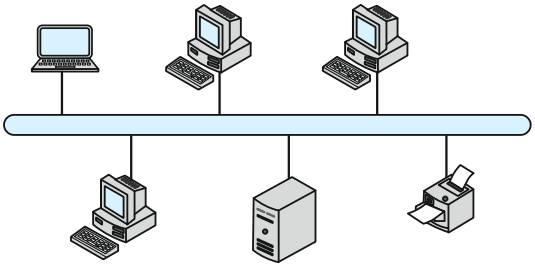
i. **Physical topology:** placement of various nodes.
ii. **Logical topology:** deals with the data flow in the network.

> **Key Points to Remember:**
> 1. Shape
> 2. Working
> 3. Advantages and Disadvantages

Types: Bus, Ring, Star, Mesh, Tree, Hybrid.

Bus Topology

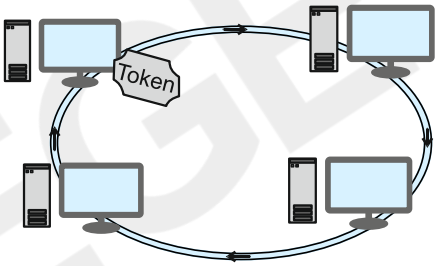
- **Definition:** All devices share a single central cable.
- **Data Flow:** Bidirectional along the bus.
- **Pros:**
  - Easy to set up.
  - Minimal cabling.
- **Cons:**
  - Cable failure breaks network.
  - Limited devices and slow with traffic.



- **Use:** Suitable for small networks or temporary network setups.

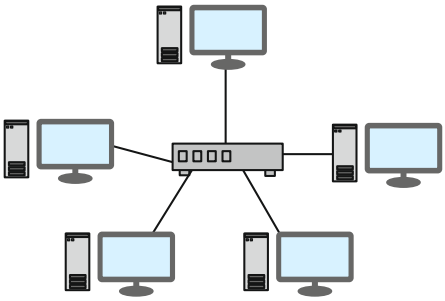
Ring Topology

- **Definition:** Devices form a closed loop, each connected to two others.
- **Data Flow:** Typically, one-directional using a circulating token (some versions support bidirectional flow).
- **Pros:**
  - Simple setup.
  - Fast data transfer.
- **Cons:**
  - One failure affects the whole network.
  - Harder to troubleshoot.
- **Use:** Token-based, orderly data networks.



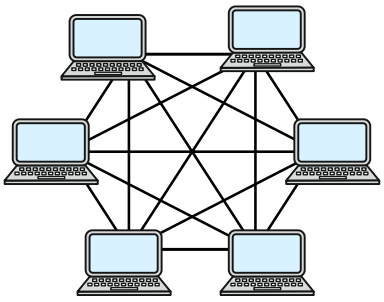
Star Topology

- **Definition:** All devices connect to a central hub/switch.
- **Data Flow:** Via central hub to destination.
- **Pros:**
  - Easy setup & fault isolation
  - One device failure doesn't affect others
- **Cons:**
  - Hub failure breaks network
  - More cabling needed
- **Use:** Common in homes/offices.



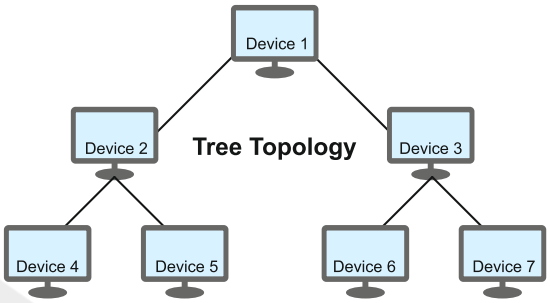
Mesh Topology

- **Definition:** Every device is connected to every other device directly.
- **Data Flow:** Data can take multiple paths between devices.
- **Advantages:**
  - Highly reliable and fault-tolerant.
  - If one link fails, data routes through other paths.
  - Provides high privacy and security.
- **Disadvantages:**
  - Expensive and complex to install.
  - Requires lots of cables and ports.
- **Use:** Critical networks requiring high availability and redundancy.



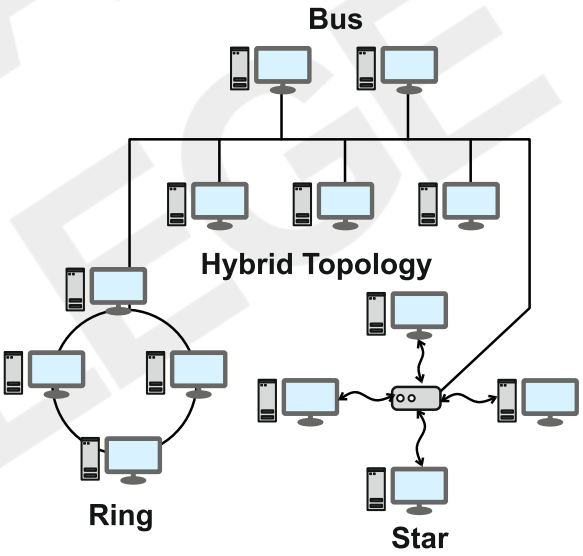
Tree Topology

- **Definition:** Hierarchical topology combining multiple star networks connected to a central bus backbone.
- **Data Flow:** Data flows from leaves to root and vice versa through the backbone.
- **Pros:**
  - Scalable and easy to manage.
  - Fault isolation is easier than bus topology.
  - Supports future expansion.
- **Cons:**
  - If the backbone fails, entire network segments go down.
  - More cabling required than bus or star.
- **Use:** Large networks with hierarchical structure, like university campuses.



Hybrid Topology

- **Definition:** Mix of two or more topologies (e.g., bus + ring + star).
- **Data Flow:** Depends on combined types.
- **Pros:**
  - Flexible & scalable.
  - Customizable design.
  - Easier fault isolation.
- **Cons:**
  - Complex setup.
  - Higher cost.
- **Use:** Large, tailored networks.

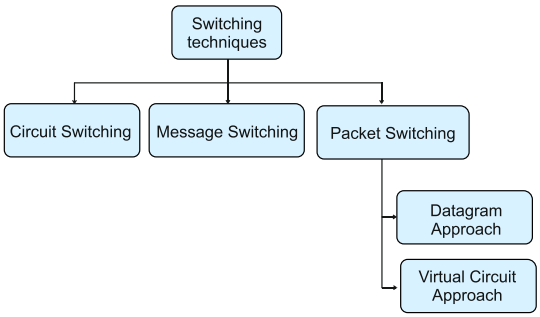


Test your understanding

i. Which network topology is the most expensive to implement and maintain?
ii. Which topology allow data to flow in only one direction?
iii. Which topology ensures communication continues even if one link fails?
iv. Which topology offers centralized control at minimal cost for connecting branch offices to a central hub?

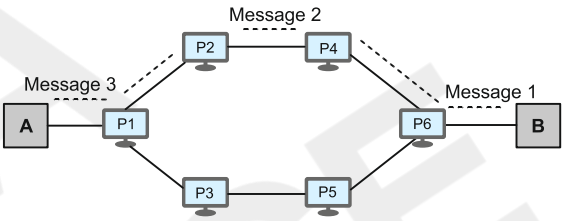
14. What is Switching?

Switching is a method of transmitting data between devices by selecting the best available path.



i. Circuit Switching

- **Circuit switching** is a switching technique in which a **dedicated communication path** is established between two endpoints for the duration of a transmission.
- **Phases:**
  1. Connection Establishment.
  2. Data Transfer.
  3. Connection Termination
- **Used in:** Traditional telephone networks (a dedicated line is reserved during a call).
- **Pros:** Reliable, fixed path.
- **Cons:** Wastes resources if idle.

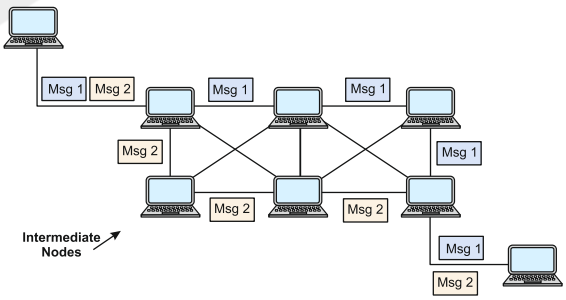


ii. Message Switching

- Uses **store-and-forward** mechanism.
- Message is transferred as a **complete unit** → stored → and then forwarded.
- Suitable for **email** and non-real-time data.
- Not suitable for streaming applications due to intermediate message storage delays.

Pros:
- No need for a dedicated path.
- Handles variable message lengths easily.

Cons:
- Slower due to message storage at each node.
- Not ideal for real-time communication.



iii. Packet Switching

- Data is broken into **small packets**, each sent independently.
- Uses **store-and-forward** at each node.
- Efficient for internet, emails, and most data networks.
- Supports multiple paths and dynamic routing.

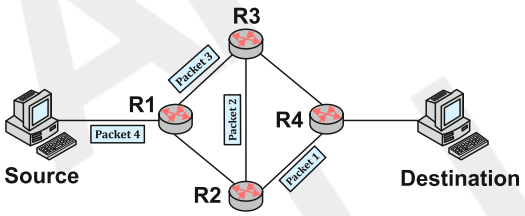
Pros:
- Efficient bandwidth use.
- Robust and scalable.
- Efficiently handles bursty traffic, where data is sent in irregular bursts.

Cons:
- Possible delays and packet loss.
- Requires complex routing.

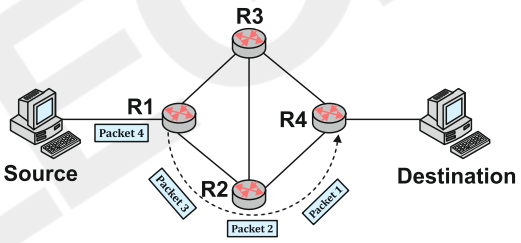
Types of packet switching

Feature	Datagram Approach	Virtual Circuit Approach
Connection	Connectionless: No path is established before data transfer	Connection-oriented: Path established before data transfer
Routing	Each packet may take a different route	All packets follow the same established route
Addressing	Packets carry full destination address	Packets carry a virtual circuit identifier
Setup Time	No setup required	Setup phase before data transfer
Reliability	Packets may arrive out of order or be lost, making it less reliable.	Packets arrive in order (more reliable)
Overhead	Higher, since each packet carries the complete destination address.	Lower, as packets use short virtual circuit identifiers instead of full addresses.
Example	Internet Protocol (IP)	ATM, Frame Relay

Datagram Approach



Virtual Circuit Approach



Test your understanding

i. Students all get on the same bus together. Which switching method does this remind you of?
ii. College friends come from different places and meet up at the canteen. Which switching technique does this represent?
iii. How does a virtual circuit reduce overhead compared to the datagram approach?

Devices

1. Devices

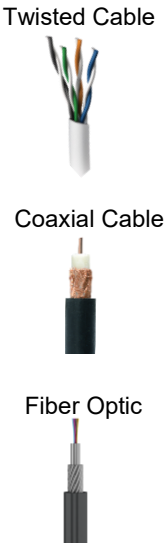
Hardware: Cable, Repeater, Hub, Modem

Hardware + Software: Bridge, Switch, Router

Software: Gateway, IDS, Firewall.

2. Cables

Cables are a medium used to send data from one device to another.



Types of Cables:

i. Unshielded Twisted Pair (UTP) Cables

- Commonly used in homes and offices.
- Pairs of wires are twisted to reduce electromagnetic interference (EMI).

ii. Coaxial Cable

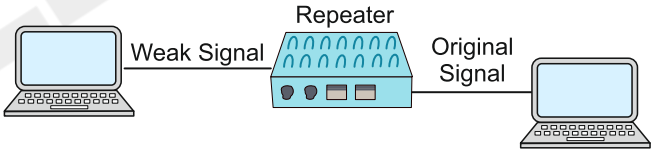
- Has a single copper wire in the center.
- Used for cable TV and internet.

iii. Fiber Optic Cable

- Sends data using light signals.
- Very fast and supports long distances.

3. Repeaters

Work at the Physical Layer
A **2-port device** (one input, one output)



Why Repeaters Are Needed:

- When signals travel long distances, they lose strength **(attenuation).**
- A **repeater boosts** the signal back to its **original strength**, so it doesn't get lost while traveling.

Key Points:

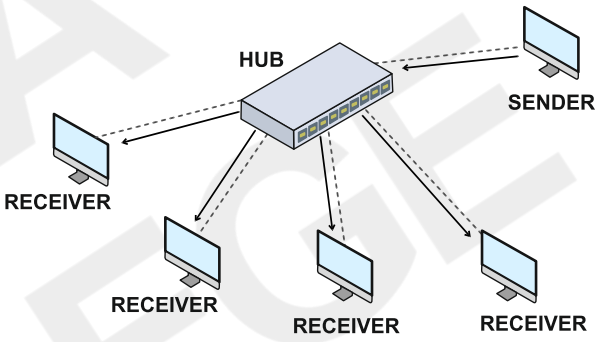
- Placed at regular intervals along the cable.
- **Does forwarding** of the signal.
- **No filtering** — only boosts and sends.
- **Collision domain = n** (where n is the number of devices) because repeater transmits without memory or buffering
  - No buffer or memory.
  - Does not use store-and-forward; simply amplifies and transmits signals in real-time.

4. **Difference Between Repeater and Amplifier:**

Feature	Repeater	Amplifier
Purpose	Restores signal to its **original strength**	Just **boosts** the signal power (e.g., 1x → 2x or 3 x)
Noise Removal	**Removes noise** from the signal	**Amplifies both** signal and noise

5. **Hub**

- **Works at the Physical Layer.**
- It is **pure hardware** (no smart processing).
- Acts as a **multiport repeater** (has multiple input/output ports).

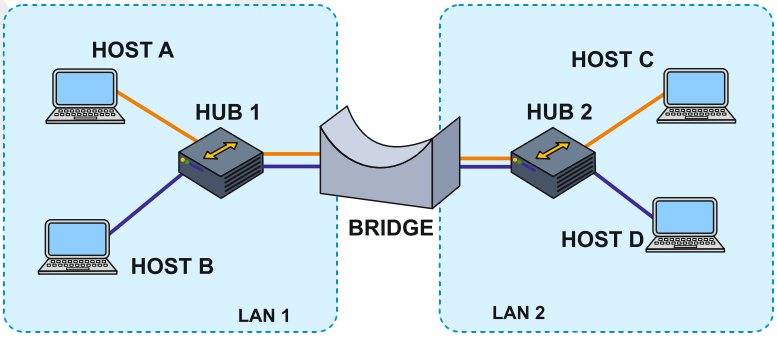


Key Features:

- **Forwards** data to **all connected devices** (broadcasts).
- **No filtering –** sends everything to everyone
  - This causes **high traffic** on the network.
- Hubs are prone to **collisions** since all data is broadcast.

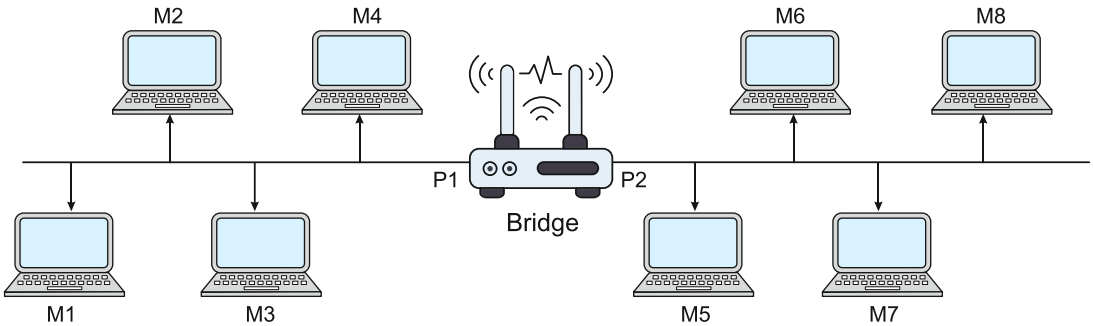
6. **Bridges**

- Work at the Data Link Layer (Layer 2)
- Used to connect **two or more** LAN segments.
- More intelligent than hubs or repeaters.



Key Features:

- **Does forwarding** of data
- Checks **MAC addresses** to decide where to send data
- Can **filter traffic** and **reduce collisions ⇒ because of store-and-forward** mechanism
- Maintains a table to **learn** which MAC addresses are on which side



Example:

- Has **two interfaces/ports:**
  - **P1** (connected to M1, M2, M3, M4)
  - **P2** (connected to M5, M6, M7, M8)

- If **M1 sends a message to M3**,
  - The message goes to the **Bridge first**
  - Bridge checks its MAC table
  - It learns **M3** is on the same port segment P1
  - So, it **does not forward** the message to P2.
  - This reduces unnecessary traffic on the other segment.

- If **M1 sends a message to M7**,
  - The message goes to the **Bridge first**
  - Bridge checks its MAC table
  - It learns **M7** is on the **other side** of **P1**
  - So, it **forwards** the message to P2

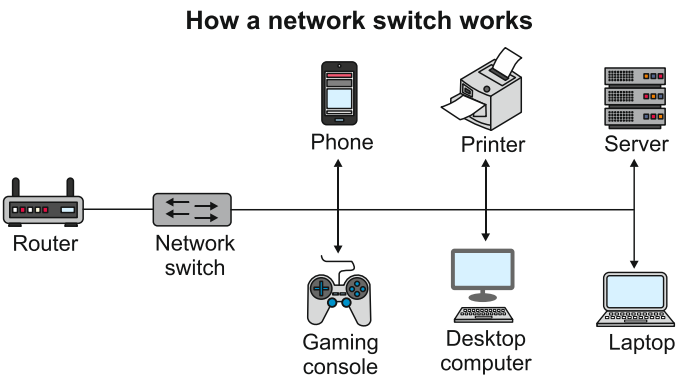
This is something **repeaters and hubs cannot do.**

7. Types of Bridges: Static vs Dynamic

Feature	Static Bridge	Dynamic (Transparent) Bridge
Configuration	Manual (by administrator)	Automatic (self-learning)
Learning Capability	No	Yes
MAC Address Table	Manually maintained	Built and updated automatically
Handling New Devices	Requires manual update	Learns automatically through traffic
Port Changes	Manual reconfiguration needed	Automatically adapts
Initial Operation	Minimal or no broadcasting (depending on manual table setup)	Broadcasts initially to learn devices
Flexibility	Low	High

8. Switch

- Operates at **Layer 2 (Data Link Layer)**
- Functionally similar to a multiport bridge, but more advanced in terms of speed and switching logic
- Supports **full-duplex links** (communication both ways simultaneously)
- Creates **separate communication paths** for different device pairs (e.g., A-B, C-D)
- Results in **minimal network traffic**
- Collision domain per port = 1; **no collision across ports**

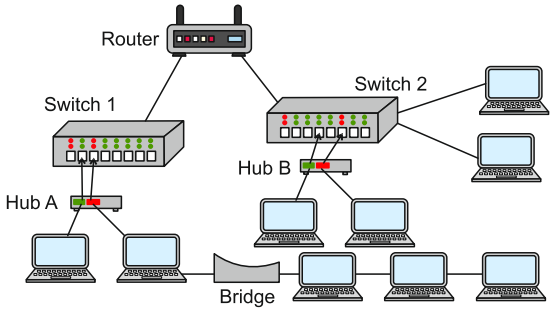


- **No collisions** occur because each link is a dedicated circuit

Example:

- Commonly used to connect devices to the network, e.g., **Laptop** → **Switch** → **Router**

9. Roputer

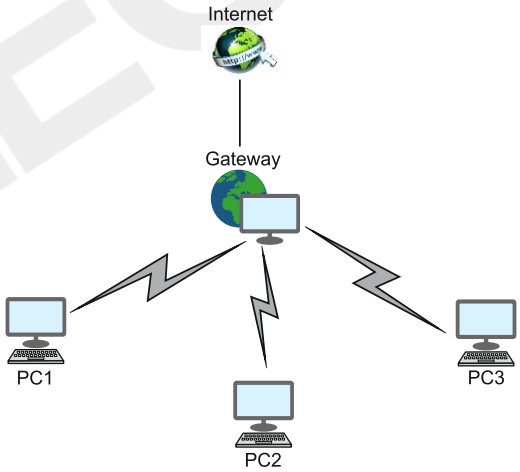


- Works at the Network Layer (Layer 3) of the OSI model.
- Connected to **multiple networks.**
- Forwards packets using a **routing table**.
- Routers **drop or ICMP unreachable**; flooding is not default behavior.
- Routes packets based on **IP addresses.**
- Routers separate collision domains, but collisions may still occur on half-duplex Ethernet interfaces.

Protocols Used

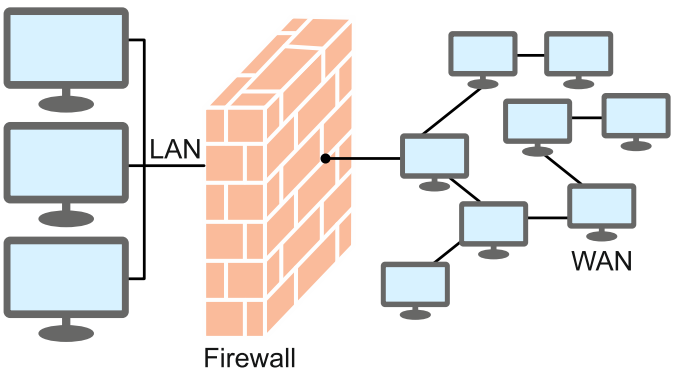
- **RIP (Routing Information Protocol)**
- **OSPF (Open Shortest Path First)**
- **BGP (Border Gateway Protocol)**

10. Gateway



- Connects two different networks (e.g., **LAN to WAN**).
- Acts as a **translator** between different network protocols.
- **Placed at the edge of LAN**, connecting to the internet.
- Gateways work across **multiple OSI layers** depending on protocol translation but is commonly associated with layer 7 (Application Layer)
- Handles **protocol conversion** and sometimes **NAT/firewall.**
  Example:
  A VoIP gateway connecting IP-based networks with traditional telephone networks.

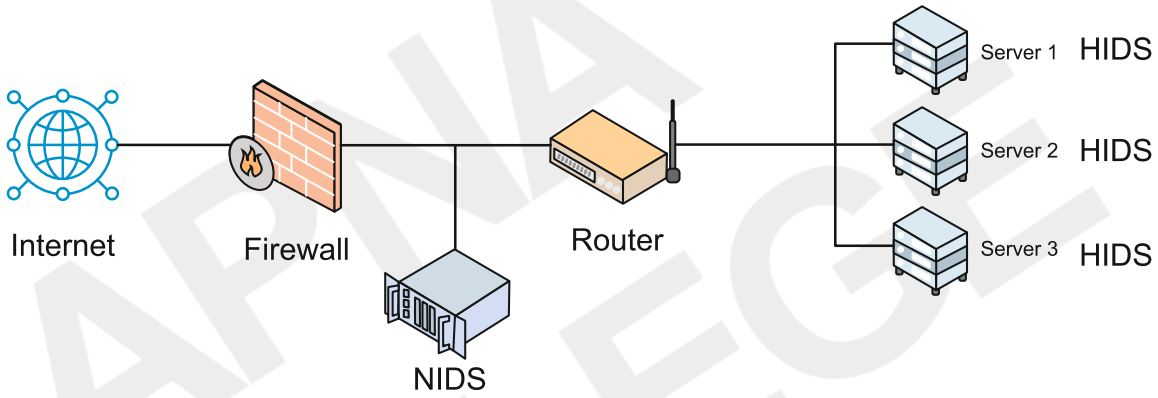
11. Firewall



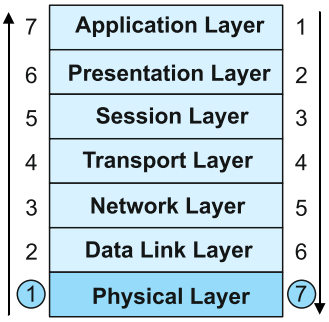
- A **security device** that monitors and controls network traffic.
- Acts as a **barrier** between trusted (LAN) and untrusted (Internet) networks.
- **Placed at the network boundary**, usually before/after the router.
- Filters traffic based on **IP, port, or protocol** using predefined rules.
- Can be **hardware, software, or both.**
  Example: pfSense, Cisco ASA, Windows Defender Firewall.

12. IDS (Intrusion Detection System)

- **Monitors network or system activity** to detect suspicious behavior or attacks.
- **Does not block traffic**, only **detects and alerts**.
- Can detect threats like **unauthorized access, malware, or policy violations**.
- Works by Analyzing **traffic patterns, logs, or behavior**.
- Two main types:
  - **NIDS** (Network IDS) - monitors entire network.
  - **HIDS** (Host IDS) - monitors individual devices.
- Example:
  Snort (for NIDS), OSSEC (for HIDS)



Physical Layer



1. **Where is it located in the OSI Model?**

   It is the 1st layer from the bottom (7th from the top) in the OSI Model.

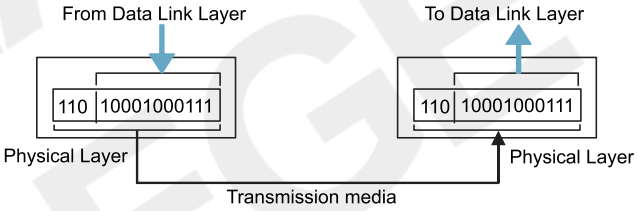
2. **What is the form of data?** → Bits

3. **Which devices operate at the Physical Layer?**

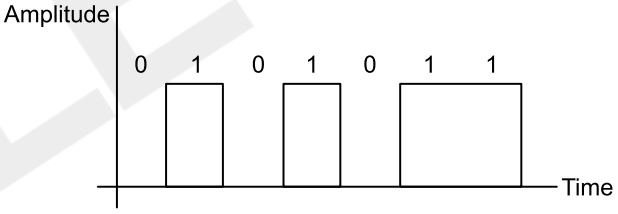
   - Cables
   - Hub
   - Repeater

4. **What are the services offered by this layer?**

   i. **Bit-by-bit Transmission**
   - Data is sent as a continuous stream of **individual bits**.

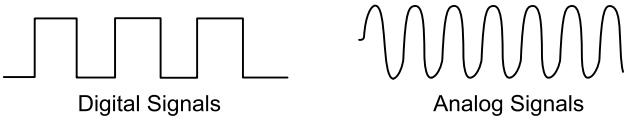


   ii. **Encoding and Decoding**
   - **Encoding** converts **data into signals** suitable for transmission (e.g., electrical, optical).
   - **Decoding** converts received signals into a form that can be interpreted as data by higher layers.



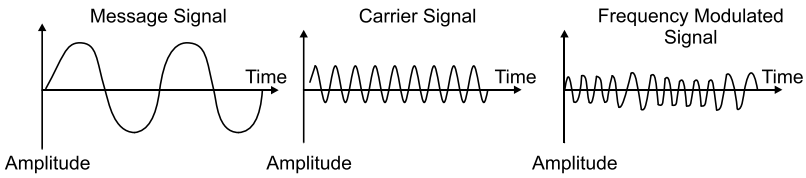
Example of digital signal representation

   iii. **Signal Transmission**
   - It refers to transmitting encoded data over a physical medium such as wires, fiber optics, or air.



   - The transmission can use **analog** (continuous) or **digital** (discrete) signals, depending on the communication method.

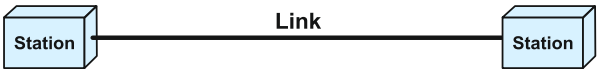
   iv. **Modulation and Demodulation**
   - Modulation alters a carrier signal to encode and transmit data.
   - Demodulation retrieves the original data from the modulated carrier signal at the receiver.
   - A modem is the device used for modulation and demodulation.



5. Types of connection

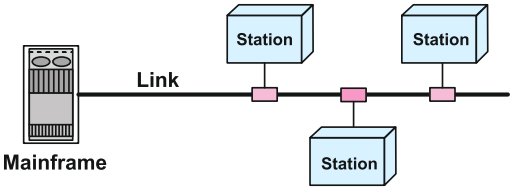
i. Point-to-point

- Connects exactly **two devices** directly.
- Used for direct communication between two devices.



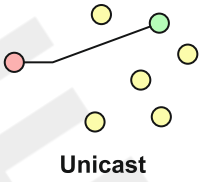
ii. Multipoint

- **Multiple devices** share a single link.
- Used for shared communication over a single physical link.

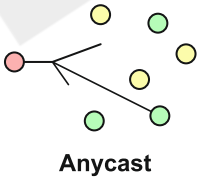


6. Types of communication modes?

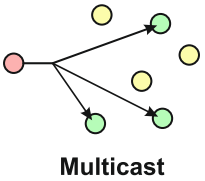
i. Unicasting: If the message is sent from the source to a **single node**, it is known as **unicasting**. This is the most common communication method in networks, where data is sent from one sender to one specific receiver.



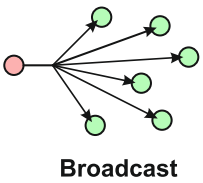
ii. Anycasting: If the message is sent from the source to **any one of the nodes**, it is known as **anycasting**. It is commonly used in Content Delivery Networks (CDNs), where the request is routed to the nearest or best-performing server.



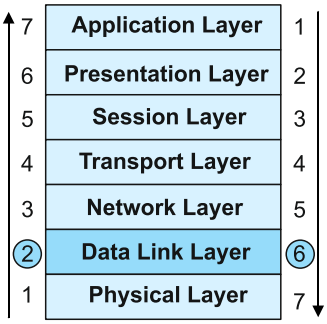
iii. Multicasting: If the message is sent to a **subset of nodes** from the source then it is known as **multicasting**. Used to efficiently deliver the same data to a selected group of receivers simultaneously.



iv. Broadcasting: If the message is sent from a source to **all the nodes** in a network, it is known as **broadcasting**. Protocols such as DHCP and ARP use broadcasting within local area networks (LANs).



Data link Layer



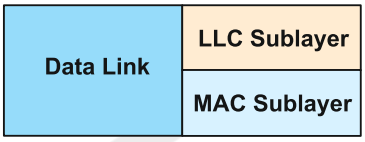
1. **Where is it located in the OSI Model?**

   It is the 2nd layer from the bottom (6th from the top) in the OSI Model.

2. **What is the form of data?** → Frames

3. **Sublayers**

   i. **Logical Link Control (LLC)**
   - Handles **flow control** and **multiplexing** between upper layers and the MAC sublayer.
   - Provides:
     - Error detection notifications to higher layers
     - **Acknowledgments**



   ii. **Media Access Control (MAC)**
   - Manages device interaction with the physical medium.
   - Responsibilities:
     - **Frame addressing**
     - Provides access control to the physical transmission medium
     - Encapsulates Network Layer packets into frames and forwards them to the Physical Layer.

4. **What are the services offered by this layer?**

   i. **Framing**
   Encapsulates packets into frames for transmission.

   ii. **Physical / MAC Addressing**
   Adds source and destination MAC addresses to the frame header.

   iii. **Flow Control**
   - Prevents the sender from overwhelming the receiver.
   - Protocols:
     - **Stop-and-Wait**
     - **Go-Back-N (GBN)**
     - **Selective Repeat (SR)**

   iv. **Error Control**
   Provides error detection and optional correction.

   Error Detection Techniques:
   - Single-bit parity check
   - Two-dimensional parity check
   - Checksum
   - Cyclic Redundancy Check (CRC)

   Error Correction Technique:
   - Hamming Code

v. **Access Control**
Manages access to the shared medium to avoid collisions.

Random Access Protocols:
- ALOHA
- CSMA (Carrier Sense Multiple Access)
  - 0-persistent
  - 1-persistent
  - p-persistent
- CSMA/CD (Collision Detection)

Controlled Access Protocols:
- Polling
- Token Passing

Channelization Protocols:
- FDMA (Frequency Division Multiple Access)
- TDMA (Time Division Multiple Access)

5. How does data link layer work?

- **Receives raw bits from the Physical Layer** and organizes them into meaningful **frames**.
- **Detects and corrects errors** in the received frames before passing data to the Network Layer.
- **Uses MAC addressing** to ensure the frame reaches the correct device on the local network.
- **Controls access to the shared medium**, ensuring devices transmit data without collisions.
- **Delivers clean, framed data** to the Network Layer, which adds IP logic and handles routing.

6. Which devices operate at the Data Link Layer?

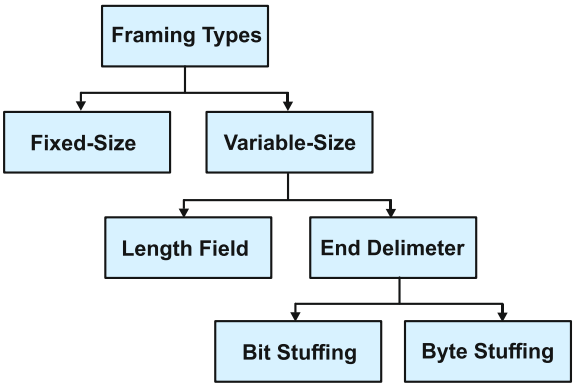
- **Switch**
- **Bridge**
- **Network Interface Card (NIC)**
- **Wireless Access Point (WAP)**

7. Framing

- Groups raw bits into **frames** for clear separation.
- Enables reliable node-to-node (hop-to-hop) delivery across the physical link.
- Structure of a frame:

  - **Flag:** Like an envelope in a postal system. It defines the start and end of data (delimiter).
  - **Header:** Contains source and destination MAC address.
  - **Payload Field:** It is the actual data.
  - **Trailer:** Error detection and correction bits.

Types of Framing



i. **Fixed-Size Framing**
   - Frame size is constant.
   - No delimiters needed; length defines frame boundaries.

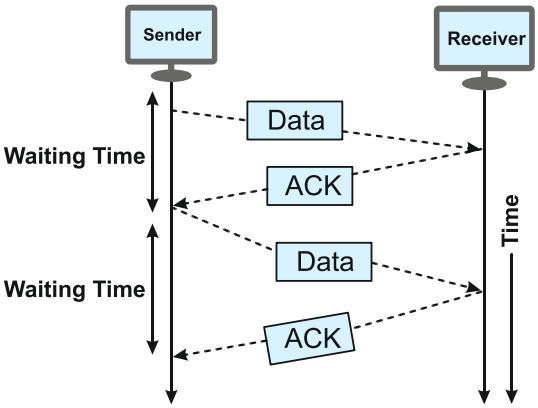
ii. **Variable-Size Framing**
   - Frame size varies; requires delimiters.
   - Two methods:
     a. **Length Field -** Specifies frame size in the header.
     b. **End Delimiter -** Special pattern marks end of frame.
        - **Byte Stuffing -** Adds extra byte to avoid confusion (character oriented).
        - **Bit Stuffing -** Adds extra bits to avoid confusion (bit-oriented).

8. Physical/ MAC addressing

- **NIC (Network Interface Card):** A hardware component that connects a device to a network.
- **NIC** can be wired (Ethernet) or wireless.
- **WAP (Wireless Access Point):** A device that lets wireless devices connect to a wired network using Wi-Fi.
- **WAP** connects wireless devices to a wired network and extends wireless coverage.
- **MAC Address:** A unique hardware identifier assigned to the NIC for identification on a local network.

9. Flow Control

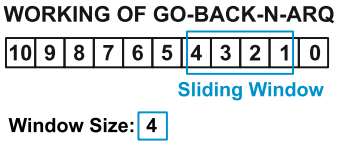
i. **Stop-And-Wait**
   - **Window size = 1** (only one frame in transit).
   - Sender sends **1 frame**, waits for **ACK** before sending next.
   - Sender is **idle** during wait.
   - Prevents data loss if receiver is slower.
   - Simple but **inefficient** due to idle time.

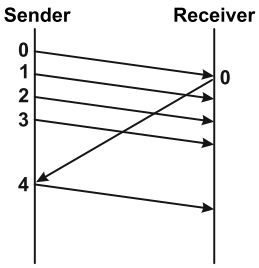


Problems in Stop-and-Wait Flow Control
   - Lost Data
   - Lost Acknowledgement
   - Delayed ACK/Data

ii. **Go-Back-N Protocol**
   - It is a **sliding window** protocol.
   - It is a **flow control and error control** protocol.
   - The sender can send **N (size of sender window) frames in a row** without waiting for an acknowledgment.
   - Receiver sends cumulative acknowledgment for the last correctly received in order frame.

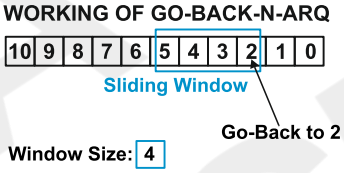
Transmission Process

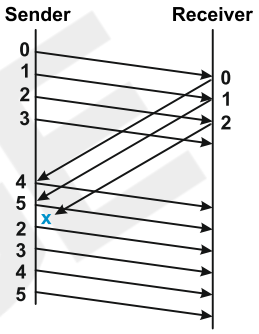




- Suppose the sender has to send **11 frames** (0 to 10).
- Window size is **4**.
- Sender sends frames: **0, 1, 2, 3**.
- Receiver sends ACK for **frame 0**.
- Sender slides window and sends **frame 4** (now window is 1,2,3,4).
- Receiver ACKs frame 1, sender sends frame 6 (window is 3,4,5,6).
- This process continues.

Error Handling (Go-Back Mechanism)

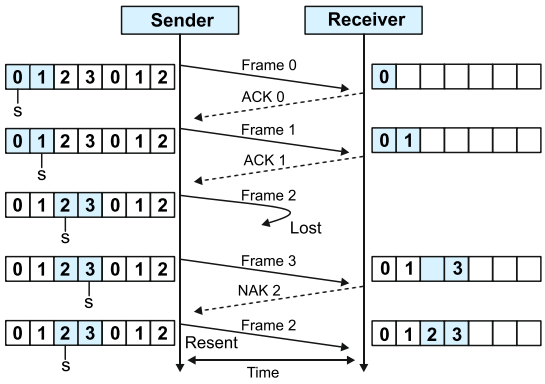




- If a **frame or ACK is lost**, the sender waits.
- If ACK for **frame 2** is missing:
  - Sender cannot send **frame 6**.
  - After a timeout, sender **goes back** to frame 2.
  - Retransmits frames 2 onward (e.g., 2, 3, 4...), even if some were received correctly.
- This ensures **reliable delivery** but may cause **extra retransmissions**.

iii. Automatic Repeat reQuest / Selective Repeat ARQ

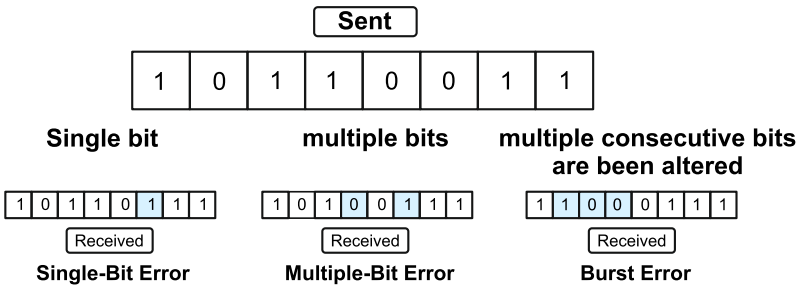
- Multiple frames are sent; multiple frames are acknowledged.
- **Only lost/damaged frames** are retransmitted.
- **Receiver buffers** out-of-order frames temporarily.
- **ACK** sent for each correctly received frame.
- **NACK** sent for missing or corrupted frames.
- **More efficient** than Go-Back-N (no unnecessary retransmissions).



10. Error detection and correction

If the receiver does not receive the data exactly as sent by the sender, data is said to be **corrupted** and **error** has occurred.

Types of errors



i. Error detection method

a. Single Bit Parity Check

- Helps detect **single-bit errors** in transmission.
- A basic **error detection** technique.
- **Adds 1 extra bit (parity bit)** to the data.
- Ensures total number of 1s is **even**.
  - If data has **odd** number of 1s ⇒ **add 1**
  - If data has **even** number of 1s ⇒ **add 0**
- In advanced parity methods, like 2D matrix-based parity, errors involving 2 or 3 bits can be detected but not corrected.

b. Checksum

- A computed value (often a sum or similar operation) of data bits before transmission.
- Checksum is transmitted along with the data.
- Receiver recomputes the checksum and **compares it**.
- If values **don't match**, an **error** is detected.

c. Cyclic Redundancy Check

- Uses **polynomial division** to generate a check value.
- More **robust** than parity and checksum.
- Commonly used in **networks** and **file systems** for error detection.

ii. Error Correction method

a. Two types of error correction:

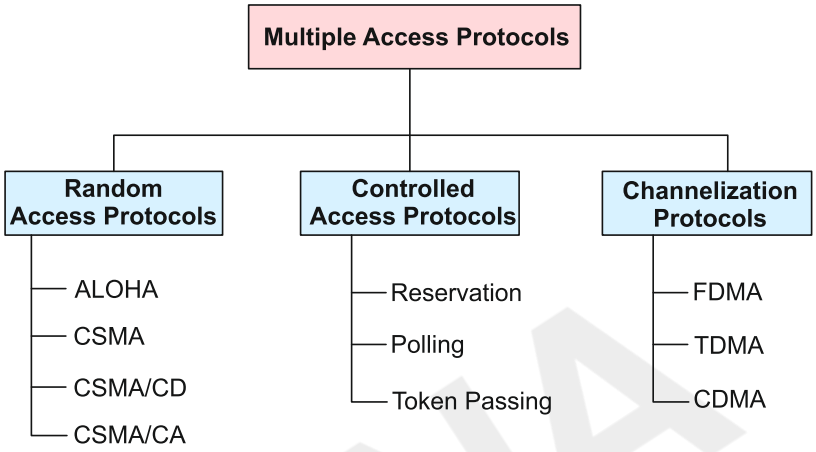
- **Backward Error Correction:**
  Receiver **detects error**, asks sender to **retransmit**.
- **Forward Error Correction:**
  Receiver uses **extra bits** to **detect and correct** errors without **retransmission**.

b. Hamming code for error correction

- **Parity bits** are added at positions that are powers of 2 (1, 2, 4, 8…).
- Each parity bit checks a fixed set of positions based on binary rules.
- **Even parity** is used — parity bit is 0 if total 1s (including itself) is even, else 1.
- Sender sends the full message (data + parity bits).
- Receiver recalculates parity bits to detect errors.
- The binary value formed by the positions of incorrect parity bits indicates the position of the error.
- Receiver flips the bit at that position to correct the error.

11. MAC (Multiple Access Control)

Determines how multiple devices share the communication channel.



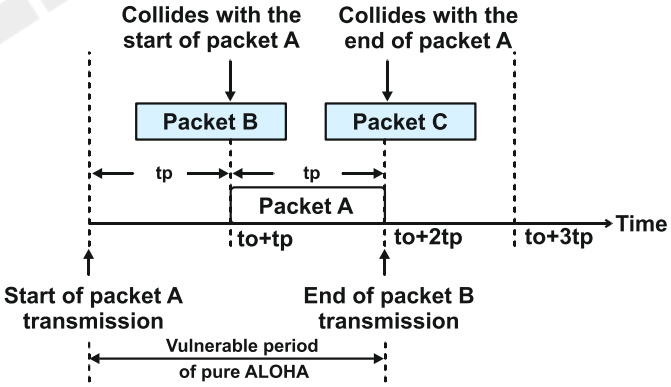
1) Random Access Protocol

- **No station has priority,** all are equal.
- **Any station can transmit** when the medium is **idle**.
- **No fixed timing,** stations transmit at **random times**.
- **Collisions may occur** if two stations send at the same time.

i. ALOHA

a. PURE ALOHA

- When a station sends data, it waits for an **ACK (Acknowledgement)**.
- If **no ACK is received**, it assumes a **collision has occurred**.
- The station waits for a **random backoff time (Tb)** before retrying.
- The **back-off time** in Pure ALOHA is random
- Pure ALOHA does not perform carrier sensing.
- Used in **early LAN protocols**, now mostly **obsolete**.
- Only **transmission time** is considered; **propagation delay is ignored**.

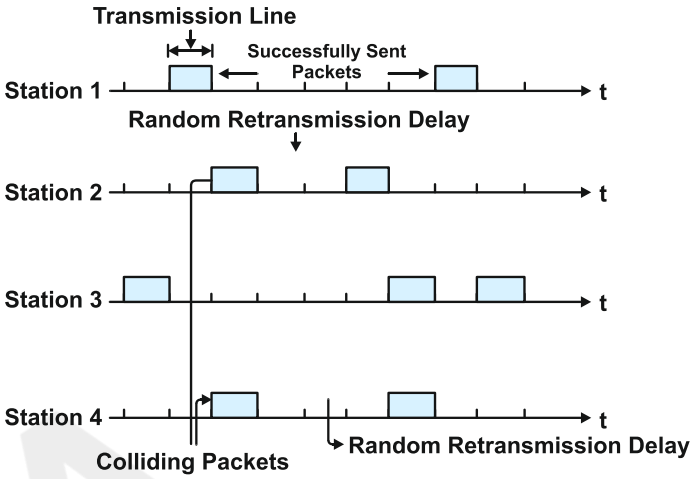


Vulnerable Time = 2 x tp

- tp = amount of time required to send a frame between the two most widely separated stations

b. SLOTTED ALOHA

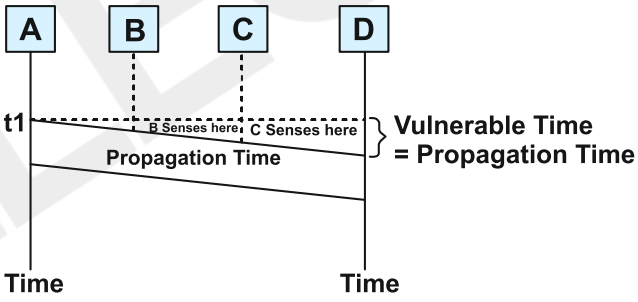
- Any station **cannot start transmission at any time**.
- **Time is divided into equal slots.**
- A station can **only start transmission at the beginning of a time slot**.
- If a station misses the start of the slot, it **must wait for the next one**.
- This reduces the **chance of collisions** compared to Pure ALOHA.



Vulnerable Time = tp

ii. CSMA (Carrier Sense Multiple Access)

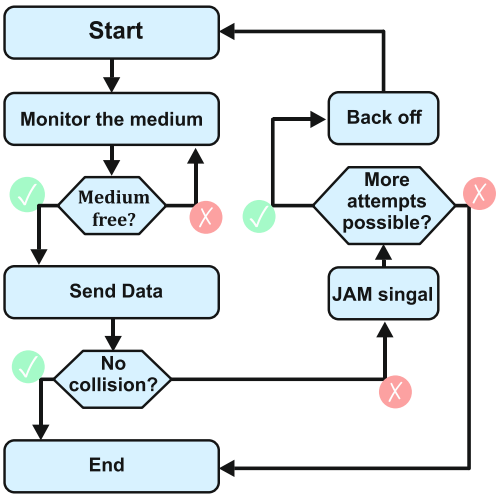
- A station **senses the medium** before transmitting.
- If the medium is **idle**, it starts **transmitting data**.
- If the medium is **busy**, it **waits** until it becomes idle.
- **Fewer collisions** than ALOHA due to sensing.
- However, **collisions can still occur** due to **propagation delay**.
- Two stations may sense the medium as idle at the **same time** and transmit, causing a **collision**.



CSMA ACCESS MODES

1-Persistent CSMA	P-Persistent CSMA	0-Persistent CSMA
• Sense channel → if **idle, transmit immediately**. • If **busy, keep checking** continuously. • Transmit **as soon as** medium becomes idle (with 100% chance). • **High chance of collision** if multiple nodes do the same.	• Sense channel → if **idle, transmit with probability p**. • With **(1-p)** probability, **wait a time slot**, check again. • Repeat until the frame is sent. • **Balance between efficiency and collision control**. • Used in Wi Fi, radio	• **Priority-based**; nodes have predefined **transmission order**. • If channel is **idle**, each node waits for **its slot**. • Used when nodes follow a fixed backoff scheme.

iii. CSMA/CD (Collision Detection)



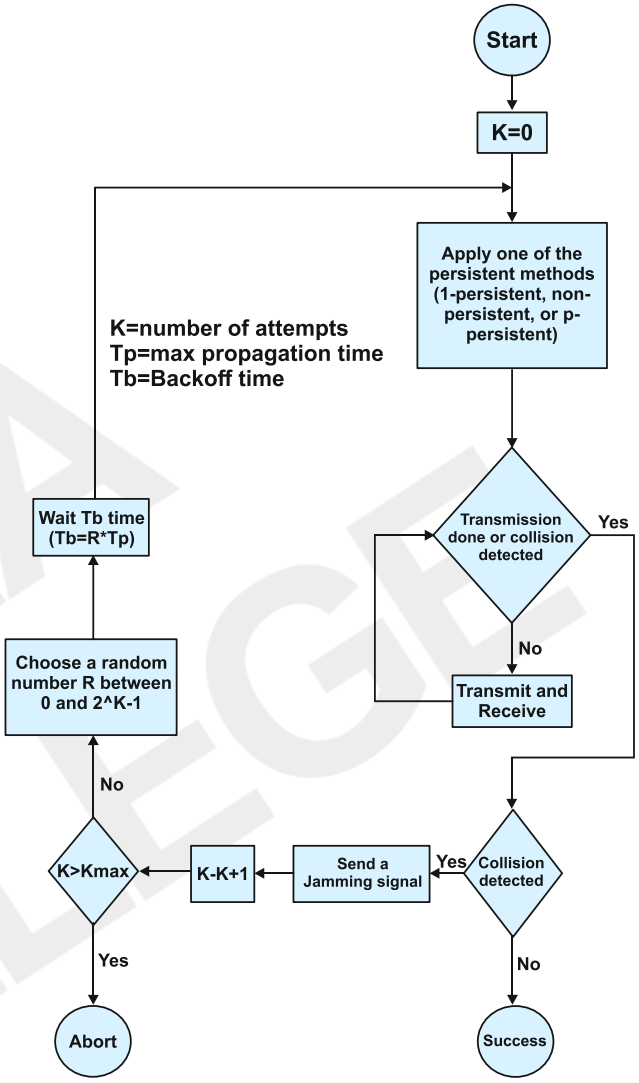
- It is a slight modification of CSMA.
- If a **collision occurs** during transmission:
  - Sends a **jam signal** to notify others.
  - Waits a **random backoff time**, then retries.
- CSMA/CD doesn't use acknowledgments; it relies on collision detection.
- Requires **transmission time** $\geq 2 \times$ **propagation delay** for proper collision detection. So that collisions can be detected before transmission ends."
- **Not used in modern networks** (e.g., switched Ethernet).

iv. CSMA/CA (Collision Avoidance)

- **Carrier Sensing:**
  - Station senses the medium to check if it's **idle or busy**.
- **Interframe Space (IFS):**
  - If medium is idle, station waits for a short time (IFS) before proceeding.
  - This helps avoid overlap due to **propagation delay**.
- **Contention Window:**
  - After IFS, the station selects a **random backoff time** (in slots).
  - If medium becomes busy during backoff, the **timer pauses**, and resumes once it's idle again.
- **Transmission:**
  - Once backoff ends and the medium is idle, data is transmitted.
- **Acknowledgement:**
  - Receiver sends an **ACK**.
  - If **ACK not received**, sender assumes collision and **retries**.

CSMA/CA Flow:

- **Station senses the channel:**
  Check if the medium is idle.
- **If medium is idle:**
  Wait for **Interframe Space (IFS)** —a short delay based on priority.
- **After IFS:**
  Check medium again.
  - If still idle, proceed to send data.
  - If busy, wait.
- **If medium was busy initially or after IFS:**
  Enter **Contention Window:**
  - Randomly select a backoff slot before transmission attempt.
  - If transmission fails, double the contention window size and select a new random backoff time (exponential backoff).
  - Pause timer during busy periods and resume when medium becomes idle.
- **Send data.**
- **Wait for acknowledgement (ACK):**
  - If ACK received, transmission is successful.
  - If no ACK is received within timeout, retransmit using the same backoff mechanism.

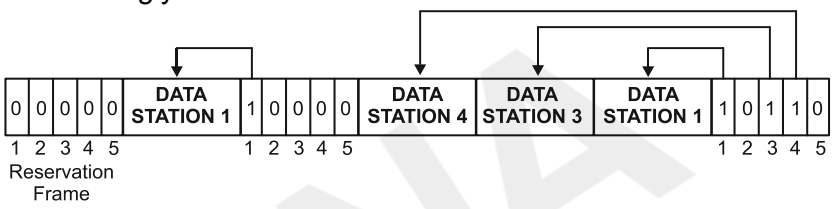


2) Controlled Access Protocol

A protocol used in shared channels to avoid collisions by regulating device transmission access.

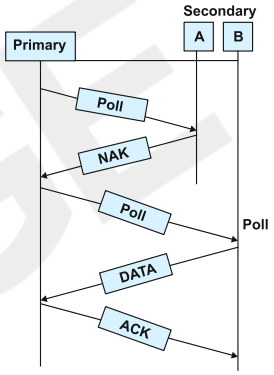
i. Reservation

- Devices reserve future time slots by marking their requests in a reservation frame (containing mini-slots for requests). This is done in a centralized or distributed manner depending on the system.
- Only the devices that reserved slots are allowed to transmit, **eliminating collisions**.
- **Image shows:** Stations mark their slots in the reservation frame and send data accordingly in turn.



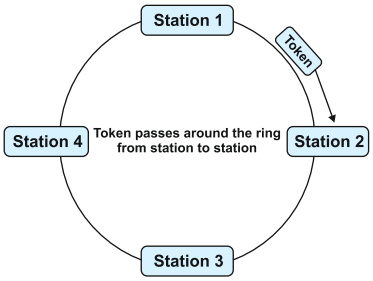
ii. Polling

- A **central controller (primary)** polls devices one at a time to check if they have data to send.
- Only the **polled device** can respond, preventing data collisions.
- **Image shows:** Primary polls Station A and B; A responds with no data (NULL or silence), B sends DATA and receives ACK.



iii. Token Passing

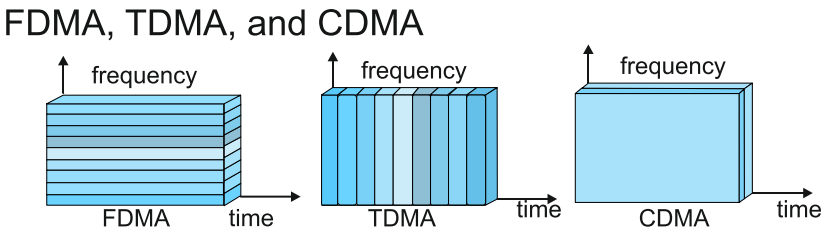
- A token circulates in a logical ring; only the holder transmits. If lost, the protocol regenerates the token.
- Ensures **orderly and collision-free** data transmission.
- **Image shows:** The token being passed from Station 1 to 2, 3, and then 4 in a circular manner.



3) Channelization protocols

Channelization protocols divide the **shared communication medium** so that multiple users can **transmit data simultaneously without interference**.

FDMA	TDMA	CDMA
Frequency Division Multiple Access	Time Division Multiple Access	Code Division Multiple Access
Users get separate frequency bands	Users get separate time slots	Users share frequency & time with unique codes
Transmit simultaneously on different frequencies	Transmit one after another in sequence	Transmit simultaneously using different codes
Divides total bandwidth into fixed frequency bands, which can lead to underutilization.	TDMA is more bandwidth-efficient than FDMA in digital systems.	Highly efficient due to shared frequency/time; supports encryption for security.
Used in analog systems (e.g., radio)	Used in GSM (2G) networks	Used in 3G networks (e.g., Qualcomm)



Ethernet Frame Format

- **LAN technology Protocol**
- Used by CSMA/CD

PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH	DATA	CRC
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46-1500 Bytes	4 Bytes

i. Preamble (7 Bytes)

- Used for **bit synchronization.**
- Receiver gets alert before the actual data starts flowing.

ii. Start Frame Delimiter - SFD (1 Byte)

- Marks **start** of frame (before destination address).
- Sometimes considered part of Preamble (making 8 Bytes total).

iii. Destination Address (6 Bytes)

- MAC address of the **receiver.**
- Can be unicast, broadcast, or multicast.

iv. Source Address (6 Bytes)

- MAC address of the **sender.**
- Always unicast (LSB of first byte is 0).
- Multicast addresses have LSB = 1.

v. Length (2 Bytes)

- Indicates **length of the data field** (up to 1500 Bytes).
- Value range can go up to 65535 in IEEE 802.3 but Ethernet payload limit is 1500 Bytes.

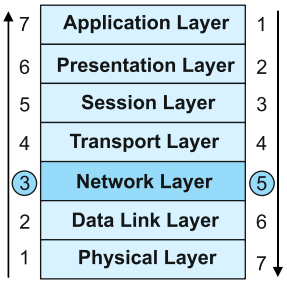
vi. Data / Payload (46-1500 Bytes)

- Contains actual **IP header + user data.**
- Padding is added if total frame (excluding preamble / SFD) is less than 64 Bytes (14-byte header + 4-byte CRC + 46 – 1500 Bytes data).

vii. Cyclic Redundancy Check - CRC (4 Bytes)

- 32-bit **error-detecting hash.**
- Calculated on: Destination Address + Source Address + Length/Type + Data + Padding (if present).
- Used to detect **corruption** at receiver side.

Network Layer



1. **Where is it located in the OSI Model?**

   It is 3rd layer from bottom; 5th from top.

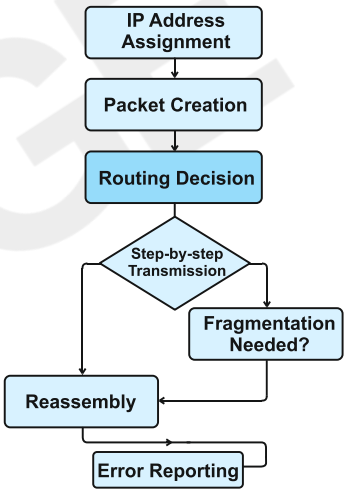
2. **What is the data format?** → Packets

3. **What are the services offered by this layer?**

   - Routing
   - Logical Addressing
   - Packet Switching
   - Forwarding
   - Fragmentation & Reassembly
   - Subnetting
   - NAT (Network Address Translation)

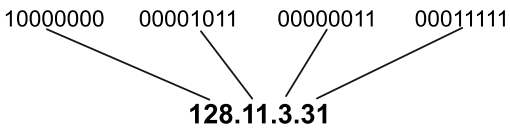
4. **How does the Network Layer work?**

   i. Each device gets a **unique IP** address for identification.
   ii. Data is **split** into packets with **source & destination IPs.**
   iii. Routers choose the **best path** to reach the destination.
   iv. Packets move **hop-by-hop** through routers.
   v. Large Packets are broken into **smaller packets.**
   vi. At the destination, packets are **combined** into original data.
   vii. If **delivery fails**, error messages (like **ICMP**) are sent back.



5. **What is an IP address?**

   - An IP address can be 32-bit (IPv4) or 128-bit (IPv6), used to identify devices on a network.
   - Helps locate and communicate with devices, especially **outside** your local network (LAN).
   - 32 bits → each 8 bits are separated by **dots(.)**
   - **Types of IP address**
     i. Public
     ii. Private
   - We have two **different addressing types**
     - **Classful addressing**
     - **Classless addressing (CIDR)**



6. **What is Classful addressing**

   - IPv4 addresses divided into **5 classes:** A, B, C, D, E.
   - Each class has a **fixed range** and a **class-specific subnet mask**.
   - Determines how the IP is split into **network** and **host** parts.

Class	First Octet Range	Leading Bits	Network Bits	Host Bits	Default Subnet Mask	Purpose	Number of Networks Calculation
A	0-127	0	8	24	255.0.0.0	Large networks	2^7=128
B	128-191	10	16	16	255.255.0.0	Medium/large networks	2^{14}=16,384
C	192-223	110	24	8	255.255.255.0	Small networks	2^{21}=2,097,152
D	224-239	1110	N/A	N/A	Not defined	Multicasting	N/A
E	240-255	1111	N/A	N/A	Not defined	Research / Experimental	N/A

First octet range

The **first octet range** shows the first number in the IP address.
Ex. 0-127 = 0.0.0.0 to 127.255.255.255 (similarly for B-E)

Subnet Mask

32-bit number used to determine the network portion of an IP address via **bitwise AND**.

IP & Subnet Mask = Network ID

Example

- 10.20.30.40 & 255.0.0.0 → 10.0.0.0
- 172.16.5.4 & 255.255.0.0 → 172.16.0.0
- 192.168.1.20 & 255.255.255.0 → 192.168.1.0

Number of Networks Calculation

Total network bits – fixed leading bits = bits for networks

Number of networks = $2^{\text{Total network bits}}$

Example (Class A):

- Network bits = 8
- Leading bits fixed = 1
- Remaining bits = 8 - 1 = 7
- Networks = 2^7= 128

Number of Hosts Calculation

Total host bits = bits reserved for hosts in the IP class

Number of hosts per network = $2^{\text{Host bits}} - 2$

Example (Class A)

- Host bits = 24
- Actual hosts = 2^{24} - 2 = 16,777,214

Why do we subtract 2?

Type	Description	Host Bits	Purpose
Network ID	First IP address in the network	All host bits = 0	Identifies the network
Broadcast Address	Last IP address in the network	All host bits = 1	Sends data to all hosts

Example for Class C

Address Type	Host Portion (last 8 bits)	IP example
Network ID	00000000	192.168.1.0
Usable hosts	00000001 to 11111110	192.168.1.1 to 192.168.1.254
Broadcast Address	11111111	192.168.1.255

7. Disadvantages of Classful addressing?

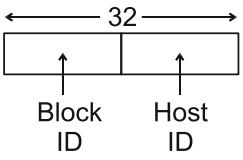
- **Class A, B:** Blocks are too large for most organizations → many addresses wasted.
- **Class C:** Blocks can be too small for some organizations → insufficient addresses.
- **Class D:** Used for multicast groups → addresses not assigned to individual hosts, causing wastage.
- **Class E:** Reserved for future use → currently unused, causing wastage.

8. Classless addressing

- **Improved IP system** replacing classful addressing.
- **Efficient IP allocation** based on user need.
- **CIDR = Classless Inter-Domain Routing.**
- IPs are given in format:

CIDR Block

- A **CIDR block** is a range of IP addresses assigned dynamically.
- It contains the **exact number of IPs** needed.



Rules to Form CIDR Block

i. All IPs in the block must be **contiguous.**
ii. **Block size must be a power of 2** (e.g., 2^1, 2^2, 2^3, ...).
iii. **First IP must be divisible by the block size.**

NOTE: An address is divisible by 2^n if its last n bits in binary are all equal to 0.

CIDR Notation (/n)

- n = number of **network bits**
- 32 - n = number of **host bits**

CIDR notation → x.y.z.w/n

Example:
192.168.1.0/28
→ 28 bits for network, 4 bits for host
→ 2^4 = 16 - 2 = 14 usable hosts.

9. Subnet

- A **subnet** is a **smaller group** inside a larger network.
- Allows **efficient data routing** within smaller, isolated sections.
- Each subnet can operate and be managed **independently**.
- Subnetting means dividing a large network into smaller networks called subnets.

Benefits of subnetting

- Efficient IP allocation.
- Reduces network congestion.
- Enhances security, scalability and speed.

10. How Subnetting works

- Each subnet has its **own IP range**; inside which devices can talk directly.
- **Routers** connect different subnets.
- **Borrow bits** from **host** part to create subnets:
- **Subnet ID =** First IP in subnet (network address).
- **Broadcast ID =** Last IP in subnet (for sending to all hosts).
- **Formula:**

$$\text{Subnets} = 2^{\text{borrowed\_bits}}$$

$$\text{Hosts} = 2^{\text{host\_bits\_left}} - 2$$

Example (for Class C):

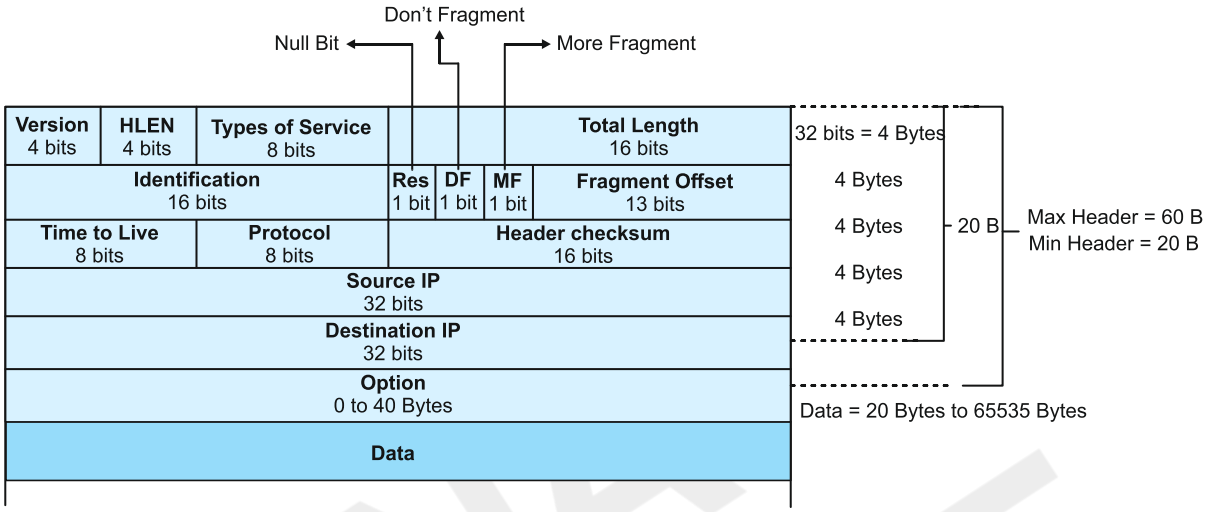
- Borrow 2 bits → $2^2 = 4$ subnets; 8 - 2 = 6 host bits left,
- $2^6 - 2 = 62$ (64 - 2) hosts each

NOTE: More subnets mean fewer hosts per subnet.

11. Subnetting types

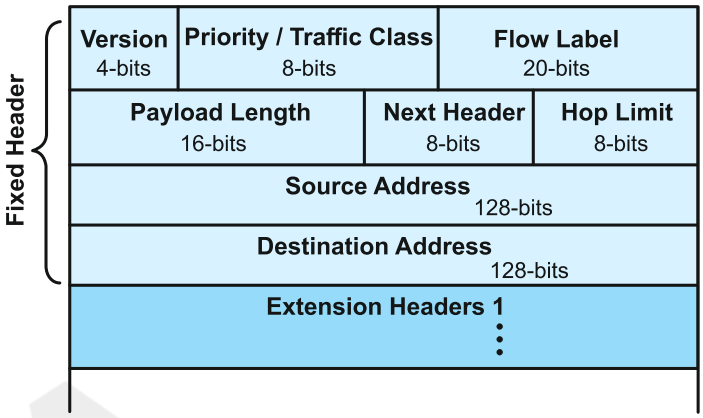
Feature	FLSM (Fixed Length Subnet Mask)	VLSM (Variable Length Subnet Mask)
Subnet Mask	Same mask for all subnets	Different masks per subnet based on need
IP Allocation	Fixed-size blocks, which may not always match specific host requirements.	Variable-size blocks, matching the host needs
Efficiency	Less efficient; wastes IP addresses	More efficient; reduces IP wastage
Addressing Type	Typically used with classful addressing	Typically used with classless addressing (CIDR)
Example	192.168.1.0/24 split into four /26 subnets (64 Ips each)	192.168.1.0/24 split into /25, /26, /27 subnets matching host needs

12. IPV4 Header



- **Version (4 bits):** IP version number (4 for Ipv4).
- **Header Length (HLEN, 4 bits):** Length of the header in 32-bit words (min 5, max 15).
- **Type of Service (8 bits):** Priority flags like Low Delay, High Throughput, Reliability.
- **Total Length (16 bits):** Size of header + data (20 to 65,535 bytes).
- **Identification (16 bits):** Unique ID for packet fragments.
- **Flags (3 bits):**
  - Reserved: Always set to 0; not currently used.
  - DF (Don't Fragment): Prevents the packet from being fragmented.
  - MF (More Fragments): Indicates that more fragments are coming after this one.
- **Fragment Offset (13 bits):** Position of fragment in original data (in 8-byte units).
- **Time to Live (TTL, 8 bits):** Limits packet lifetime (prevents infinite looping).
- **Protocol (8 bits):** Indicates protocol (TCP, UDP, ICMP) for payload.
- **Header Checksum (16 bits):** Error-check for header integrity.
- **Source IP Address (32 bits):** Sender's IP.
- **Destination IP Address (32 bits):** Receiver's IP.
- **Options (variable):** Optional data (e.g., routing info, diagnostics).

13. IPV6 Header



- **Version (4 bits):** IP version number, always 6 (binary 0110).
- **Traffic Class (8 bits):** Indicates packet priority/class; helps with QoS. Similar to Ipv4's Type of Service.
- **Flow Label (20 bits):** Identifies packet flow for special handling like QoS or real-time service.
- **Payload Length (16 bits):** Length of data following the header (includes extension headers + actual payload). Max = 65,535 bytes.
- **Next Header (8 bits):** Type of next header (can be an extension header or upper-layer protocol like TCP/UDP).
- **Hop Limit (8 bits):** Like TTL in Ipv4; max number of hops before the packet is discarded.
- **Source Address (128 bits):** Ipv6 address of the packet sender.
- **Destination Address (128 bits):** Ipv6 address of the final receiver.
- **Extension Header**
  - To replace IPV4 options field.
  - Each has a **Next Header (8 bits)** → points to next header or upper-layer protocol (TCP/UDP).
  - We can have it for Hop-by-Hop options, Destination options, Routing, Fragment, Authentication, Encapsulating Security Payload.

Important difference

Feature	Ipv4 Header	Ipv6 Header
Header Size	Variable (20-60 bytes)	Fixed (40 bytes)
Version	4	6
Address Length	32-bit	128-bit
Options/Extensions	Options in header (variable length)	Uses separate **Extension Headers**
Checksum	Present (Header Checksum)	**Not present** (removed to speed up processing)
Fragmentation	Done by routers and hosts	Done **only by source** using extension header
QoS Support	ToS (Type of Service) field	Traffic Class field
Flow Label	Not available	Present (used for flow identification)
Other Fields	Identification, Flags, Fragment Offset, etc.	Simplified; adds Flow Label, removes some fields
TTL/Hop Limit	Time to Live (TTL)	Renamed as Hop Limit

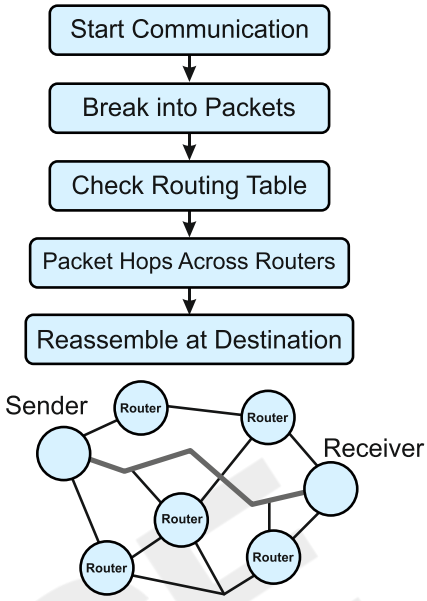
14. Routing

Routing is the process of selecting the **best path** for data to travel from a source to a destination across networks.

How does it work?

i. A device sends a **request** to another device (e.g., a web server).
ii. Data is split into **smaller packets**. Each packet gets the **destination IP** address.
iii. The router checks its **routing table** to find the best path to the destination.
iv. Packets travel through **multiple routers (hops)** to reach the destination.
v. Packets are **collected** and **reassembled** at the destination. Errors are **checked** and **corrected**.

Working of Routing

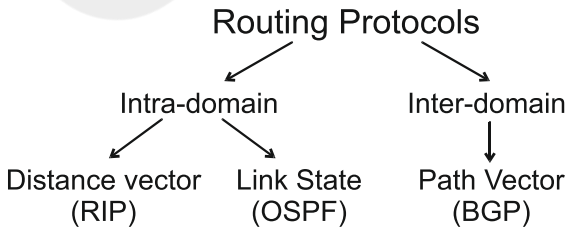


Routing table

A routing table shows the **best paths** to reach different networks.

Feature	Static Routing	Dynamic Routing
Configuration	Manually set by the network admin	Automatically updated by routing protocols
Updates	Changes only when manually reconfigured	Adjusts automatically to network changes
Use Case	Best for small and stable networks	Ideal for large, complex, or changing networks
Resource Use	Low resource usage, but less flexible	Higher resource usage, but more flexible
Example	Small office router with fixed paths	Protocols like RIP, OSPF, EIGRP

15. Types Routing Protocols [used for unicast communication]



- Intra-domain: Routing **within** a single organization or autonomous system (AS).
- Inter-domain: Routing **between** different organizations or ASes.

For the solved example have a look on the video.

i. Distance Vector Routing (RIP)

- Each router maintains a table listing the best-known distance and next hop to each destination.
- Routers **periodically share only their distance vectors** (not the entire routing table) with immediate neighbours.
- Upon receiving a neighbour's distance vector, a router updates its own vector by selecting routes with the lowest distance metric.
- Shares routing info only with neighbour routers.

RIP: Routing Information Protocol
RIP is a Distance Vector protocol using **hop count** as its metric.

Advantages:
- Simple setup and operation.
- Low resource usage.
- Auto-updates with network changes.
- Suitable for small networks.

Disadvantages:
- Slow convergence.
- Poor scalability for large networks.
- High bandwidth usage due to periodic updates.
- May choose suboptimal routes.

Count-to-Infinity Problem
- Occurs when a network route goes down or becomes unreachable.
- Routers keep increasing the hop count for that route in their updates (e.g., 14 → 15 → 16), slowly "counting to infinity".
- This leads to **routing loops** and slow convergence, where routers mistakenly believe the route is still reachable.

ii. Link state Protocol

- Also called **Open Shortest Path First (OSPF)** protocols.
- Use **Hello messages** (keep-alive) for discovering and maintaining neighbour routers.
- Use **triggered updates** for routers to send updates only when network topology changes.
- Sends updates when topology changes occur.
- **Tables used?** Neighbour Table, Topology Table, Routing Table

Advantages
- Fast **updates**, quickly **adapts** to network changes.
- Accurate **routing** with full network visibility.
- Works well in **large** networks.
- Prevents routing loops.
- More reliable and stable routing.

Disadvantages

- Requires more memory and CPU power.
- More **complex** to configure and maintain.
- Uses more **bandwidth** for routing updates.
- Not suitable for small networks due to overhead.

iii. Path Vector Protocol

- Dynamic **exterior gateway** protocol for routing between **autonomous systems (AS)**.
- Uses **path-vector mechanism** to maintain and update route information dynamically.
- Selects the **most efficient and reliable path** for data transmission across networks. Chooses the **best path** for reliable data transfer.

Advantages:

- **Scalable** for large, complex networks like the Internet.
- Supports **multi-homing** for better redundancy and reliability.
- Provides **fine control** over routing policies and traffic management.
- Quickly adapts to **network failures** and topology changes.
- Enables **inter-domain routing** across diverse networks.

Disadvantages

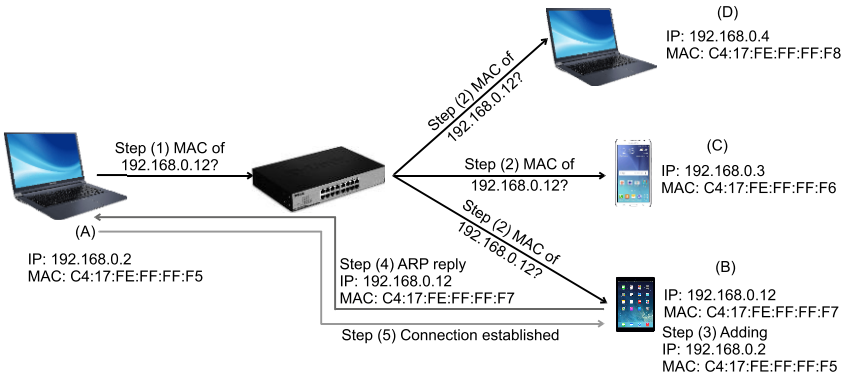
- Complex to configure and manage compared to interior protocols.
- Requires more **CPU and memory resources** on routers.
- Can be slower to converge during topology changes.
- Vulnerable to **misconfigurations** which can affect large parts of the Internet.

16. Address Resolution Protocol (ARP)

- **Purpose:** Maps an **IP address** to its corresponding **MAC (hardware) address** in a Local Area Network (LAN).
- **Role:** Enables devices to communicate within the same network by resolving IP addresses to MAC addresses.
- Operates at **Network Layer** (part of TCP/IP suite).

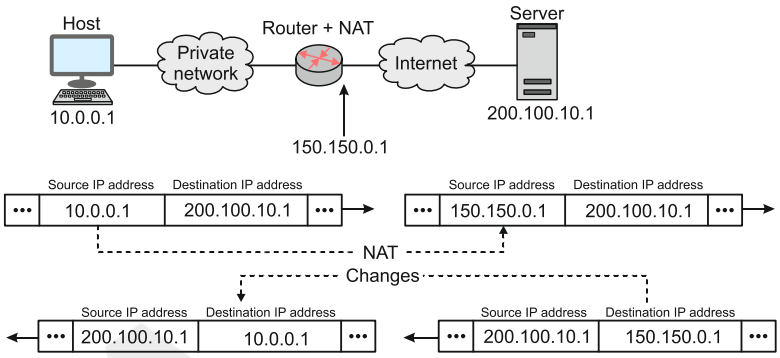
How it works?

i. Device wants to send data but only knows the IP.
ii. Checks ARP Cache for MAC address.
iii. If not found, broadcasts ARP Request asking who has the IP.
iv. Target device replies with its MAC address.
v. Sender saves MAC in ARP Cache.
vi. Data is sent using the MAC address.



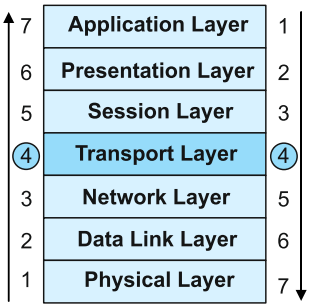
17. NAT

- **Purpose:** Allows multiple private IPs to **share a single public IP** for internet access.
- **Reason:** IPv4 addresses are limited. NAT allows devices to share one public IP by using private IPs, helping avoid address shortage.



- **How it works:**
  - NAT runs on a **router/firewall** at the network border.
  - Does both types of translation:
    - Private IP → Public IP
    - Public IP → Private IP
  - Translates IP and port numbers using a NAT table.
- **If no free address:** Packets dropped, ICMP "host **unreachable**" message is sent.

Transport Layer



1. **Where it is located in OSI Model?**

   It is 4th layer from bottom; 3rd from top.

2. **What is the data format?** → Segments

3. **What are the services offered by this layer?**

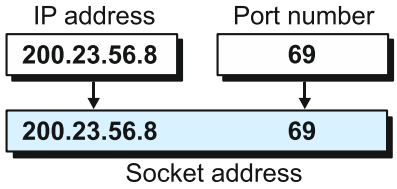
   - End-to-End control (port-to-port/ process-to-process)
   - Reliable Data Delivery
   - Error Control
   - Flow Control
   - Congestion Control
   - Multiplexing and Demultiplexing

4. **How does a transport layer work?**

   - Each application is assigned a **port number** for identification.
   - Data from applications is split into **segments** with source & destination ports.
   - **Multiplexing** allows many apps to send data simultaneously.
   - The layer ensures **reliable delivery** using acknowledgments (TCP) or sends data quickly without (UDP).
   - **Flow control** manages the data rate to avoid overload.
   - **Error detection** checks for damaged or lost segments and requests retransmission (TCP).
   - At the receiver, **demultiplexing** delivers data to the correct application.

5. **What is a Socket Address?**

   A **socket address** is a **unique identifier** for a network connection.



   Format:

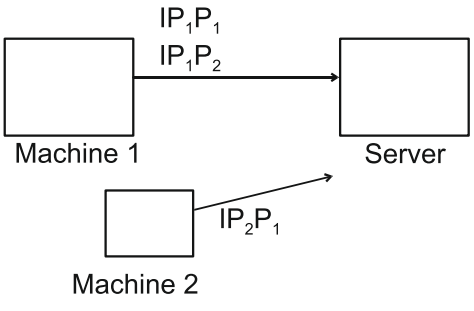
   Socket Address = IP Address + Port Number

   This combination ensures that data is delivered to the correct application on the correct device.

   i. **Why is it required?**

   - The **IP address** identifies the device (e.g., a computer or smartphone).
   - The **port number** identifies the specific application or service running on that device.

   Together, they allow accurate and complete routing of data from one application to another across a network.



ii. **Why IP Address or Port alone is not enough?**

- **If Only IP is used:** Cannot identify which application should receive the data.
- **If Only Port is used:** Cannot identify which device should receive the data.

Example:

- 192.168.1.10:80 ⇒ Web server
- 192.168.1.10:21 ⇒ FTP server
  Same IP; different ports — different applications.

6. TCP

Transmission Control Protocol (TCP) is a **connection-oriented protocol** for communications that helps in the **exchange of messages** between different devices over a network.

Features

i. **Reliable Data Transfer**
Ensures error-free, in-order delivery using acknowledgments and retransmissions.

ii. **Connection-Oriented**
Establishes a connection using a **three-way handshake** (SYN, SYN-ACK, ACK) and closes it with a **three-step process** (FIN, FIN-ACK, ACK).

iii. **Byte Streaming**

- Data from the application layer is treated as a **continuous stream of bytes**.
- The transport layer divides this byte stream into **segments**, where **one segment can contain multiple bytes**.

iv. **Full Duplex Communication**
TCP Supports **simultaneous two-way data transmission** between sender and receiver.

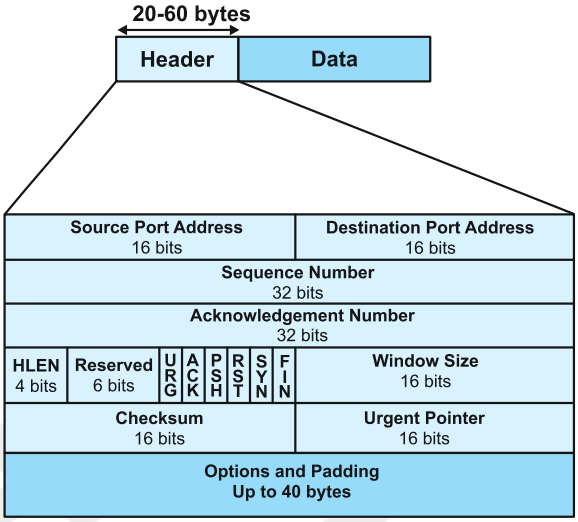
v. **Piggybacking**
When sending an acknowledgment (ACK), if there is also outgoing data, both are sent together in the same segment to **optimize efficiency**. This is known as **piggybacking**.

vi. **Flow Control**
Manages the data rate using the receiver's buffer size to prevent overflow.

vii. **Congestion Control**
Uses algorithms like **Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery** to manage network traffic.

viii. **Error Detection**
Uses a **checksum** in the TCP header to detect data corruption and request retransmission if needed.

7. TCP Header



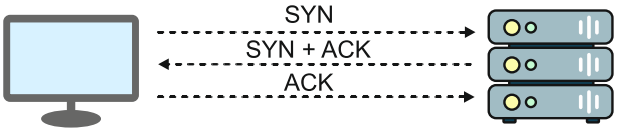
- **Source Port (16 bits):** Port number of the sending application.
- **Destination Port (16 bits):** Port number of the receiving application.
- **Sequence Number (32 bits):** Byte number of the first byte in this segment; used for reassembling out-of-order segments.
- **Acknowledgement Number (32 bits):** Byte number which the receiver expects next; acknowledges successful receipt of previous bytes.
- **Header Length (HLEN) (4 bits):** TCP header size in 4 byte words; ranges from 5 (20 bytes) to 15 (60 bytes).
  To get the actual length, multiply by 4.
- **Control Flags (6 bits):** Manage connection and data flow:
  - **URG:** Urgent pointer valid.
  - **ACK:** Acknowledgement number valid.
  - **PSH:** Push data immediately.
  - **RST:** Reset connection.
  - **SYN:** Synchronize sequence numbers (connection setup).
  - **FIN:** Finish/Terminate connection.
- **Window Size:** Size of the sender's receive window (in bytes) for flow control.
- **Checksum:** Error-checking field (mandatory in TCP).
- **Urgent Pointer:** Points to urgent data byte (valid if URG flag set).
- **Options & Padding:** Optional extra data fields; used to send additional info like Maximum Segment Size (MSS).
  - MSS for Ethernet is typically 1460 bytes (1500 – 20 IP header – 20 TCP header).

8. Flags used in Communication

- **SYN (Synchronize):** Used to **initiate** a new TCP connection between two hosts.
- **ACK (Acknowledge): Confirms** that data or a connection request has been received.
- **FIN (Finish):** Gracefully **closes** a connection when one side has no more data to send.
- **RST (Reset):** Immediately **terminates** a connection due to errors or unexpected packets.

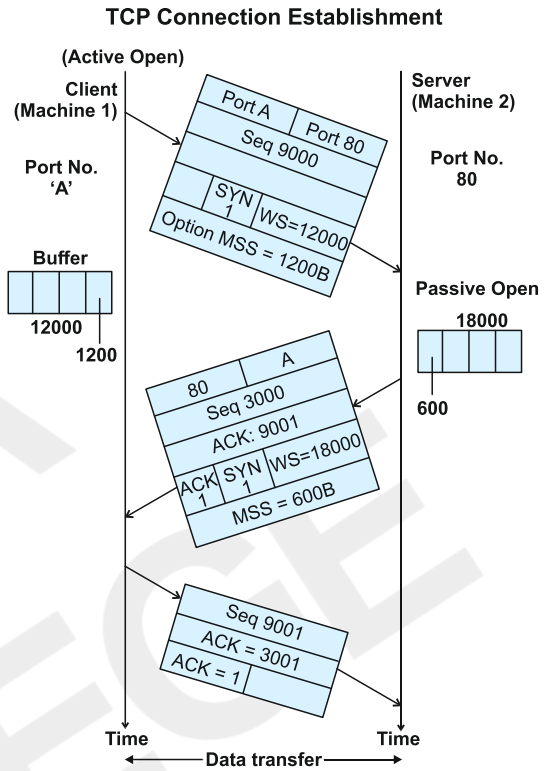
9. TCP Connection Establishment

i. **Client** sends **SYN** to request connection.
ii. **Server** responds with **SYN** and **ACK** to acknowledge and agree.
iii. **Client** sends **ACK** to confirm connection established.



i. Client sends SYN

- Picks a random sequence number (e.g., 9000).
- Sets SYN flag = 1 to request synchronization.
- Uses any source port (A) and destination port 80 (HTTP).
- Announces window size (e.g., 12000 bytes).
- MSS (e.g., 1200 bytes) in options.
- Sends SYN segment to server.



ii. Server responds with SYN + ACK

- Picks its own random sequence number (e.g., 3000).
- Acknowledges Client's SYN by setting. ACK = Client's Seq + 1 (9001)
- Sets SYN = 1 to synchronize back.
- Sets ACK = 1 to acknowledge.
- Uses source port 80 and destination port A.
- Announces its window size (e.g., 18000 bytes) and MSS (e.g., 600 bytes).
- Sends SYN + ACK segment to client.

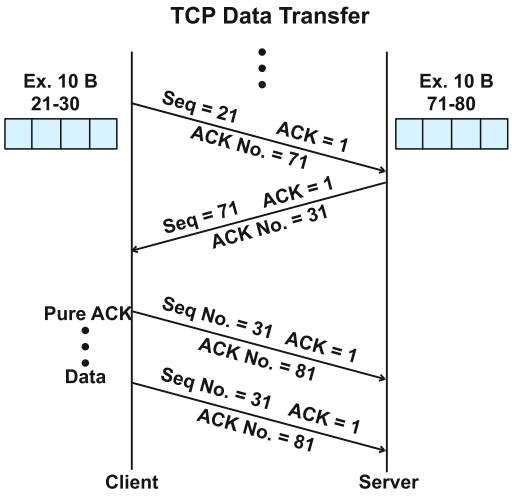
iii. Client sends final ACK

- Sends ACK with sequence number = Client's Initial Seq + 1 (9001).
- Sets ACK number = server's Seq + 1 (3001) to acknowledge server.
- Sets ACK flag = 1.
- Connection established and ready for data transfer.

10. TCP Data Transfer

i. Client to Server (Data Segment):

- Sending data bytes: 21-30 (10 bytes).
- Sequence Number: 21.
- ACK flag: 1 (acknowledging earlier segment).
- Acknowledgment Number: 71 (expecting byte 71 from server).



ii. Server to Client (Data Segment):

- Sending data bytes: 71-80 (10 bytes).
- Sequence Number: 71.
- ACK flag: 1.
- Acknowledgment Number: 31 (expecting byte 31 from client).

iii. Rule:

Acknowledgment Number always indicates the **next byte** the receiver expects.

iv. **TCP Acknowledgment**

a. Pure Acknowledgment (No Data):

If no data is to be sent, a segment can carry only the acknowledgment:

- Only ACK flag is set.
- Prevents receiver from waiting.
- Does not consume a new sequence number.
- No payload.
- Indicates successful receipt of data.

b. Data + ACK Combined:

If sending data and acknowledging at the same time:

- Sequence Number: starts with the next byte.
- ACK flag: 1
- Acknowledgment Number: updated based on last data received (e.g., 81 if last received byte was 80).
- This is called as **piggybacking**.

11. Number of Segments

- **Window Size:** 18000 Bytes
  → Total data that can be sent without waiting for ACK.
  → Reflects receiver's buffer capacity.
- **MSS (Max Segment Size):** 1800 Bytes
  → Max data per TCP segment (excluding header).
- **Segments Needed:**

Segments Needed = Window size / MSS

- In ideal conditions, Segments Needed = 18000 / 1800 = 10 segments (excluding header overhead).

12. TCP Connection Termination

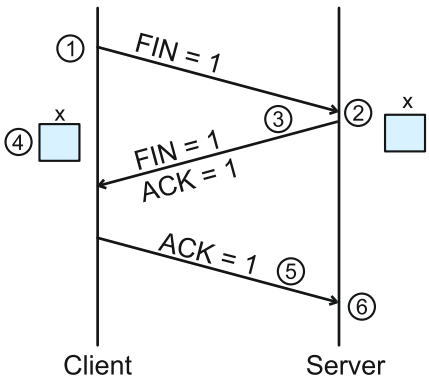
3 WAY

i. **Client sends FIN = 1**

- Indicates client wants to close its side of the connection.
- Data may or may not be sent along with FIN.

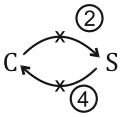
ii. **Server releases client-related resources**

- Client → Server communication is closed (half-close).
- Client cannot send data anymore, but can still send ACKs without resources.



iii. **Server responds with FIN = 1 and ACK = 1**

- Server signals it also wants to close connection.
- Acknowledges client's FIN.



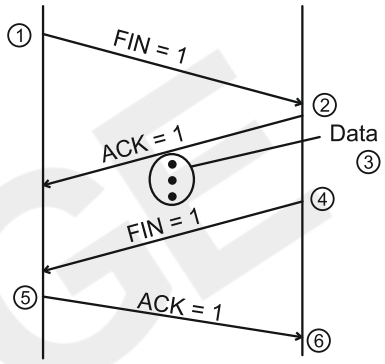
iv. **Client releases server-related resources**
- Server → Client data sending stops.
- Server can still send ACKs without resources.

v. **Client sends final ACK (ACK = 1)**
- Confirms server's FIN

vi. **Server receives final ACK and closes the connection fully**
- Both sides have released resources.
- Connection termination complete.

4 WAY

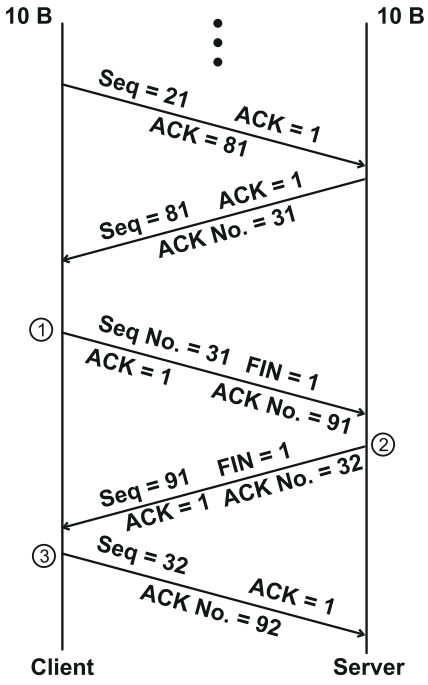
i. **Client sends FIN** = 1 to close its side.
ii. **Server sends ACK** = 1 to acknowledge FIN (connection half closed).
iii. **Server waits** (may send/receive data) before closing.
iv. **Server sends FIN =** 1 when ready to close.
v. **Client sends final ACK =** 1 to acknowledge server's FIN.
vi. Connection fully closed after final ACK.



TCP Connection termination example
Initial State: Ongoing Data Transfer

- **Client → Server:**
  SEQ = 21, ACK = 1, ACK No = 81
  (Client sending data starting from byte 21, acknowledging server's data up to byte 80)

- **Server → Client (ACK Segment):**
  SEQ = 81, ACK = 1, ACK No = 31
  (Server acknowledges receipt of data up to byte 30)



Step 1: Client initiates Connection Termination

- **Client → Server (FIN Segment):**
  SEQ = 31, FIN = 1, ACK = 1, ACK No = 91
  (Client wants to close the connection; acknowledges Server data up to byte 90)

Step 2: Server responds with FIN

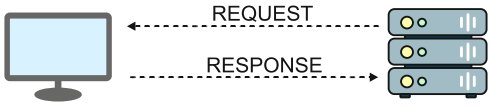
- **Server → Client (FIN + ACK Segment):**
  SEQ = 91, FIN = 1, ACK = 1, ACK No = 32
  (Server acknowledges Client's FIN, and also sends its own FIN)

Step 3: Client sends Final ACK

- **Client → Server (Final ACK):**
  SEQ = 32, ACK = 1, ACK No = 92
  (Client acknowledges server's FIN)

13. User Datagram Protocol

- **Connectionless & Unreliable.**
- **No handshake**, no delivery/order/error guarantee.
- **Lightweight & fast**, low overhead.
- **No flow or congestion control.**
- **Best for:** Time-sensitive, loss-tolerant applications.
- Since UDP relies on IP, segments may follow different paths based on IP routing.



14. UDP Header

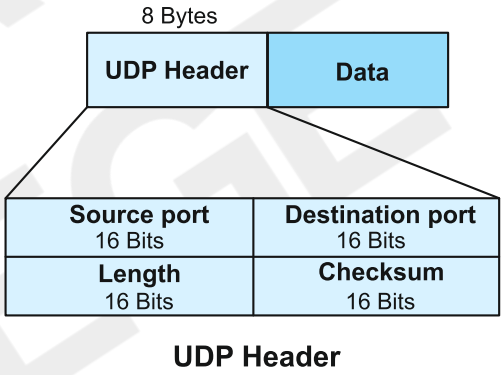
- **Fixed size:** 8 bytes (simple & efficient).
- **No options field** (unlike TCP).

UDP Header Fields (Each field is 2 bytes (16 bits)):

i. **Source Port:**
- Identifies the sender's port.
- Optional (can be 0 if not used).

ii. **Destination Port:**
- Identifies the receiver's port.
- Required for delivery.

iii. **Length:**
- Total length of UDP header + Data.
- Minimum value: 8 bytes.



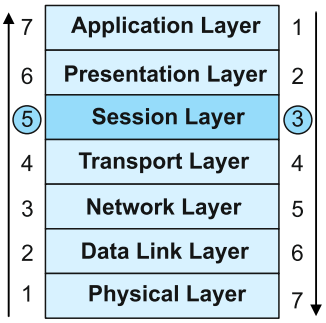
UDP Header

iv. **Checksum:**
- Used for error detection.
- Includes UDP header, data, and parts of the IP header (pseudo header).
- UDP checksum is optional in IPv4 but mandatory in IPv6.

15. Difference between TCP and UDP

TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Connection-oriented protocol	Connectionless protocol
Reliable	Not reliable
Supports robust error-checking mechanisms	Basic error-checking using checksums only
Acknowledgment mechanism is present	No acknowledgment segment
Slower compared to UDP	Faster, simpler, and more efficient
Retransmission of lost packets is supported	No retransmission of lost packets
Variable header length (20-60 bytes)	Fixed header length (8 bytes)
Used in: Web browsing, email, file transfer	Used in: Video calls, online games, streaming
Ex. HTTP, HTTPS, FTP, SMTP	Ex. YouTube, Zoom, WhatsApp Calls, DNS

Session Layer



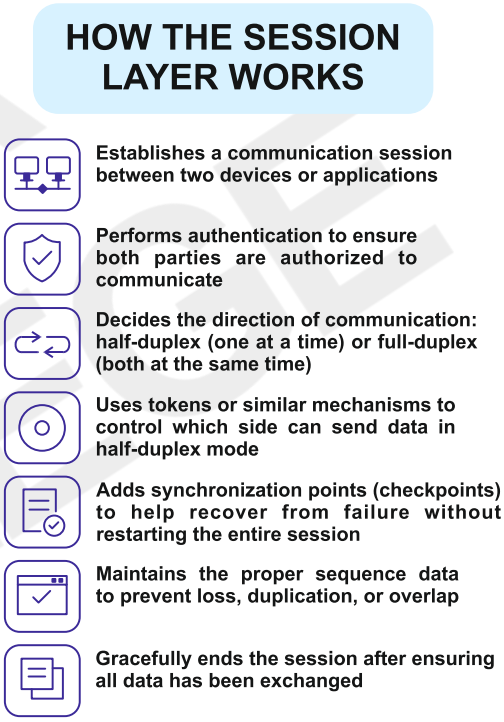
1. **Where it is located in OSI Model?**

   It is the 5th layer from bottom; 3rd from top.

2. **What is the main purpose of this layer?**

   It is responsible for **establishing, managing, and terminating** communication sessions.

3. **What are the services offered by this layer?**



   i. **Session Establishment**

   - Sets up and manages **sessions** between communicating devices.
   - Sessions are generally connection-oriented to manage reliable dialog.
   - Maps each session to the appropriate **transport layer connection**.

   ii. **Communication Synchronization**

   - Ensures reliable communication using **checkpoints** or **synchronization bits** in the data stream.
   - Facilitates **recovery** after errors or failures by rolling back to the last known good state.

   iii. **Dialog Management**

   - Controls the **direction of communication** (who sends and who receives).
   - Supports:
     - **Half-duplex mode:** Uses a **token mechanism—**only the token holder can transmit.
     - **Full-duplex mode:** Allows **simultaneous** two-way communication.

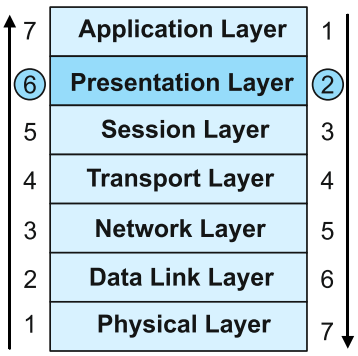
   iv. **Data Transfer**

   - Manages and monitors the **exchange of data** between systems within the session.
   - Ensures orderly flow of information.

   v. **Resynchronization**

   - Restores session state after an interruption by **resetting tokens and session parameters**.
   - Three resynchronization modes:
     - **Set:** Sets new synchronization points.
     - **Abandon:** Discards current session progress.
     - **Restart:** Restarts session from a previously saved state.

Presentation Layer



1. **Where it is located in OSI Model?**

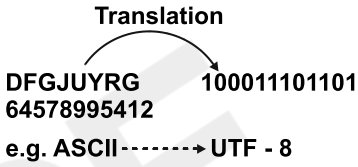
   It is 6th layer from bottom; 2nd from top.

2. **What is the main purpose of this layer?**

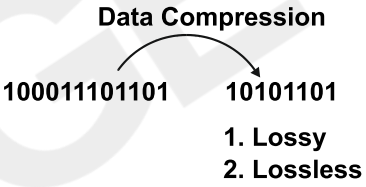
   Presents the data in a **format** that can be properly **interpreted** by the sender and receiver.

3. **What are the services offered by this layer?**

   i. **Data Translation:** Translates data between different formats to ensure compatibility between systems.
   It is also known as **Translation Layer**.



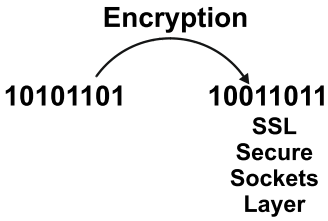
   ii. **Data Compression:** Reduces data size to minimize the number of bytes sent over the network.



   Types

   - **Lossy Compression**: Reduces file size by **permanently** removing some data. Original file **cannot be restored**.
   - **Lossless Compression**: Reduces file size **without any loss** of data. Original file can be **restored**.

   iii. **Encryption & Decryption:** Converts data into an unreadable format (encryption) and back to readable format (decryption)

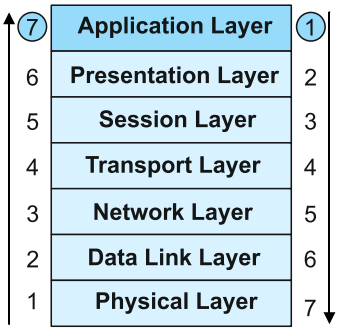


   iv. **Syntax Semantics Management:** Agrees on a common data syntax and semantics for smooth communication between devices.

Application Layer

1. **Where it is located in OSI Model?**

   It is **7th** layer from bottom; **1st** from top.



2. **What is the main purpose of this layer?**

   The application layer provides the functionality to **send and receive data** from users. It acts as the **interface** between the user and the application.

3. **Most Imp Protocols**

Protocol (Port)	Full Form	Purpose
HTTP (80)	Hyper Text Transfer Protocol	Core protocol of the Web; transmits HTML; client-server and stateless.
DNS (53)	Domain Name System	Translates domain names to IP addresses (e.g., www.abcd.com → 192.36.20.8).
TELNET (23)	Telecommunications Network	Allows remote command line access to servers; allows clients to access Telnet server resources.
DHCP (67, 68)	Dynamic Host Configuration Protocol	Provides IP addresses and configuration information to hosts.
FTP (20 data, 21 control)	File Transfer Protocol	Transfers files between devices; supports reliable, efficient data sharing.
SMTP (25, 587)	Simple Mail Transfer Protocol (Push)	Pushes emails from sender to recipient mail server.
POP (110)	Post Office Protocol (Pull)	Pulls emails from mail server to client for offline access.
NFS (2049)	Network File System	Allows remote hosts to mount and interact with file systems as if local.

Real-life examples

Protocol	Real-World Working Example
HTTP	When you open a browser and type www.example.com, your browser sends an HTTP request to the web server, which responds with the webpage content (HTML, images).
DNS	When you type www.google.com, your computer asks a DNS server "What's the IP for google.com?" The DNS server replies with the IP, so your computer can connect.
TELNET	A network admin remotely logs into a server using Telnet to configure settings or check system status over a text-based session.
DHCP	When you connect your laptop to Wi-Fi at a cafe, the DHCP server automatically assigns your device an IP address so it can communicate on the network.
FTP	You upload your website files to a web hosting server using an FTP client, transferring your HTML, images, and scripts. For secure transfer, FTPS or SFTP should be used.
SMTP	When you send an email from Outlook or Gmail, your email client uses SMTP to push the message to your mail server, which forwards it to the recipient's server.
POP	You use an email client (like Thunderbird) to download emails from your mail server via POP, allowing you to read emails offline.
NFS	A developer accesses shared project files on a remote Linux server as if they were on their own computer, using NFS to mount the remote directory locally.

4. HTTP

- **Port Number:** 80
- Uses **TCP** because HTTP itself is **not reliable.**
- **In-band protocol** (commands and data sent over the same connection).
- Each request is independent; the server does not retain client session information. Hence, it is **stateless**.

HTTP Method	Description
HEAD	Retrieves only the headers of a resource.
GET	Requests data from the server.
POST	Submits data to be processed to the server.
PUT	Updates or replaces a resource on the server.
DELETE	Deletes a resource from the server.
CONNECT	Establishes a tunnel, often for SSL connections.

5. HTTPS (Hyper Text Transfer Protocol Secure)

- **HTTPS uses TLS** (Transport Layer Security) to encrypt communication between client and server.
- It typically runs over **port 443** and provides data **confidentiality, integrity, and authentication**.

6. DNS

- The **Domain Name System (DNS)** acts like the **internet's phone book**.
- It maps **domain names** to their **IP addresses**.
- Some IP addresses are dynamic; DNS maps domain names to IPs regardless of type.
- **DNS lets you change servers without changing the domain**, keeping everything smooth for users.

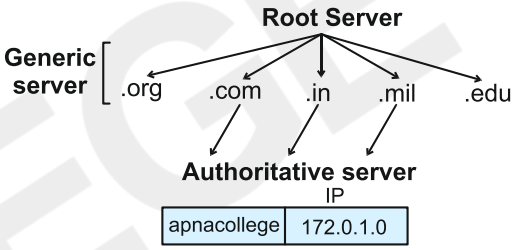
  Example: You host apnacollege.com on AWS. DNS maps the domain to its IP address. Users type the domain, not the IP. Later, you switch to a cheaper host. The IP changes, but you just update the DNS. Users still access apnacollege.com, no change on their end.

7. How it works?

We have a hierarchical structure for query-lookup.

DNS Hierarchy:

- **Root Server:** Top-level directory that points to the correct **TLD server** based on the domain extension (e.g., .com, .in).
- **TLD Server:** Manages all domains under that extension and directs to the **authoritative server** for the specific domain.
- **Authoritative Server:** Holds the actual DNS records and returns the IP address for the domain. Example: apnacollege.com → 172.0.1.0

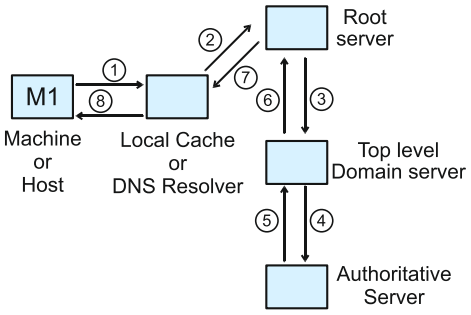


8. Types of DNS Query Lookups

A) Recursive

Your computer requests to resolve apnacollege.com in the following steps:

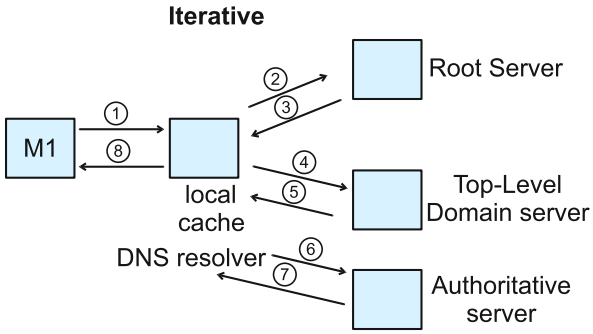
1. It first checks its **local cache** for the IP address.
   - If found, it uses it immediately.
   - If not, it sends the query to the **DNS resolver**.
2. The resolver contacts the **Root Server**.
3. The Root Server responds with the address of the .com **TLD Server** (since apnacollege.com ends with .com).



4. The resolver queries the .com **TLD Server**, which returns the address of the **authoritative DNS server** for **apnacollege.com**.
5. The resolver contacts the **authoritative server** for **apnacollege.com,** which returns the IP address, for example, 172.23.36.9.
6. The authoritative server responds to the resolver directly.
7. It travels back to **DNS Resolver**.
8. The resolver caches it inside the local cache for the future requests and then sends the IP address back to the computer.

B) Iterative

Your computer requests to resolve apnacollege.com in the following steps:



1. It checks the **local cache** for the IP address.
2. If not found, the **DNS resolver** contacts the **Root Server**.
3. The Root Server replies with a pointer to the .com **Top-Level Domain (TLD) server**.
4. The resolver queries the TLD server.
5. TLD responds with authoritative server.
6. Resolver queries the authoritative server.
7. Authoritative server replies with IP.
8. The resolver caches this IP address.

Finally, the resolver sends the IP address back to your computer.

9. FTP

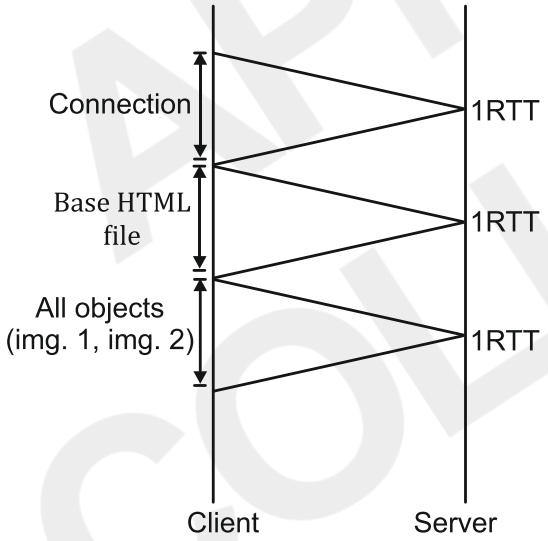
- **File Transfer Protocol**
- Used to **transfer files**.
- It is **out-of-band** protocol. It operates on **separate ports** for data and control channels.
- Data connection = non-persistent (20)
- Control connection (port 21) remains open throughout the session; data connection (port 20) is opened as needed.
- Reliable and stateful protocol.

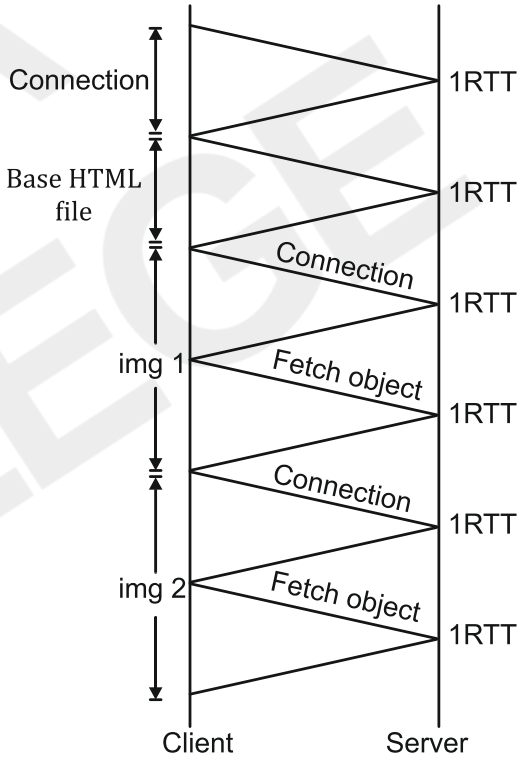
10. Email services

SMTP	POP	IMAP
Sends or **pushes** emails	Downloads or **pulls** emails	**Synchronizes** emails between server and client
Handles **outgoing** mail	Handles **incoming** mail	Handles **incoming** mail
SMTP typically works synchronously to ensure delivery acknowledgment.	Usually **synchronous**	Usually **synchronous**
Used to **send** mail from client to server or between servers	**Fetches** mail from server to client	**Accesses** mail directly on the server
Keeps emails on server (usually)	**Downloads** and often **deletes** emails from server	Emails **remain** on server, **accessible** from multiple devices

11. Difference between Persistent HTTP and Non-persistent HTTP

Persistent HTTP (HTTP/1.1)	Non-Persistent HTTP (HTTP/1.0)
Server keeps the connection open after sending response	Server closes the connection after sending response
Requires **1 RTT** to request referenced objects	Requires **2 RTTs** for each object
Less overhead (reuses same connection)	More overhead (opens new connection for each object)
Uses **TCP connection**	Uses **TCP connection**





Delays

Delays in computer networks, also called Nodal Delays, refer to the **time taken during different stages of processing and transferring** a packet. There are four main types of delays:

1. **Transmission Delay (Tt)**

Definition: Time required to **push all bits** of the packet onto the transmission medium.
Formula:

$$Tt = L/B$$

Where:

- L = Packet size (in bits).
- B = Bandwidth (in bits per second (bps)).

Example:
If bandwidth B = 1B = 1 bps and Packet size L = 20L = 20 bits, then:
$Tt = \frac{L}{B} = \frac{20 \text{ bits}}{1 \text{ bps}} = 20$ seconds

Factors affecting transmission delay:

- Multiple active sessions increase delay.
- Higher bandwidth reduces delay.
- MAC protocols affect delay in shared mediums.
- OS context switching adds to delay.

2. **Propagation Delay (Tp)**

Definition: Time taken by the **last bit to travel from sender to receiver** through the medium.
Formula:

$$Tp = \frac{\text{Distance}}{\text{Velocity}}$$

Factors affecting propagation delay:

- **Distance:** More distance → More delay.
- **Velocity:** Higher velocity → Less delay.

3. **Queueing Delay (Tq)**

Definition: Time a packet **waits in a buffer queue** before being processed.

NOTE: No exact formula exists, as it varies based on **real-time traffic** and **system conditions**.

Factors affecting queueing delay:

- Longer queues → Higher delay.
- Bursty or high traffic → Increased delay.
- Fewer servers/links → More queueing.

4. **Processing Delay (Tpro)**

   Definition: Time taken by routers or destination systems to **process the packet** (e.g., check headers, update TTL, etc.).

   NOTE: No formula, as it depends on **processor speed** and **system load.**

   Factors affecting processing delay:
   - More CPU speed→ less processing delay
   - Low System performance → high processing delay

5. **Total Delay**

   Definition: Sum of all delays.

   NOTE: It is also called as **with negligible Delay.**

$$T_{total} = T_t + T_p + T_{pro} + T_q$$

In ideal cases (no processing and queuing delay)

$$T_{total} = T_t + T_p$$

Interview Questions:

1. Can you name and briefly describe any five common network protocols and their purposes?
2. Which protocol ensures reliable communication between two endpoints on a network?
3. What is the difference between LAN, MAN, and WAN? Give examples.
4. Where would you typically use a PAN or CAN?
5. How does a client-server model differ from a peer-to-peer model?
6. Which architecture is more scalable and why?
7. What are the different types of network topologies and their advantages/disadvantages?
8. Why is star topology more commonly used than bus topology in modern networks?
9. What are the functions of the following devices: Hub, Switch, Router, Modem, Repeater?
10. How is a switch different from a hub in handling network traffic?
11. How does a router determine the best path to route a packet?
12. What's the difference between a router and a switch in terms of OSI layer and functionality?
13. What is a firewall and how does it protect a network?
14. Can a firewall block internal threats? Why or why not?
15. Explain the TCP 3-way handshake step by step.
16. What happens if the final ACK is lost during the handshake?
17. What are the four types of delays in a network? Explain each briefly.
18. Which delay is affected by congestion in the network?
19. What are the main differences between IPv4 and IPv6?
20. What is the purpose of private IP addresses?
21. Can a device with a private IP access the internet directly? Why or why not?
22. Describe step-by-step What happens when you hit enter after typing www.google.com in your browser.
23. What is the difference between unicast, multicast, broadcast, and anycast? Give real-life examples.
24. Why is a switch preferred over a hub in modern LANs?
25. In what form is data represented at each layer of the OSI model (bits, frames, packets, etc.)?
26. How does subnetting help in efficient IP address management?
27. What is DNS and how does it resolve domain names to IP addresses?
28. What is the difference between recursive and iterative DNS queries?

29. Why does a device need both MAC and IP addresses?
30. What is flow control in networking? Name two protocols that provide flow control.
31. What are two methods used for error detection and correction?
32. How does checksum differ from CRC in detecting errors?
33. Compare TCP and UDP in terms of reliability, speed, and use cases.
34. Which protocol would you use for live video streaming and why?
35. What are the responsibilities of the session layer in the OSI model?
36. What functions does the presentation layer perform in data communication?
37. Why do we need data encryption?
38. Explain how propagation delay and transmission delay differ.
39. Which delay is dependent on bandwidth and which on distance?
40. What is a VPN and how does it work?
41. What are some advantages and limitations of using a VPN?

MCQs:

1. **Which protocol ensures reliable communication?**
   A. UDP
   B. IP
   C. TCP
   D. FTP
2. **Which protocol is used to transfer files over a network?**
   A. HTTP
   B. FTP
   C. DHCP
   D. SNMP
3. **Which network is used within a single room or device range?**
   A. PAN
   B. LAN
   C. MAN
   D. WAN
4. **Campus-wide network is an example of:**
   A. WAN
   B. MAN
   C. CAN
   D. PAN
5. **Which architecture offers centralized control and better security?**
   A. Peer-to-peer
   B. Client-server
6. **In peer-to-peer networks, devices are both:**
   A. Clients only
   B. Servers only
   C. Clients and Servers
   D. None of the above
7. **Which topology connects all nodes to a central device?**
   A. Bus
   B. Ring
   C. Star
   D. Mesh
8. **Which topology provides full redundancy?**
   A. Mesh
   B. Star
   C. Bus
   D. Ring

9. **Which device works at the Data Link Layer (Layer 2)?**
   A. Router
   B. Switch
   C. Modem
   D. Repeater

10. **What does a router do?**
    A. Broadcasts data
    B. Forwards packets between networks
    C. Encrypts data
    D. Stores web pages

11. **Switches are better than hubs because they:**
    A. Use more power
    B. Flood all ports
    C. Forward data intelligently
    D. Are cheaper

12. **Which device connects two different networks?**
    A. Switch
    B. Repeater
    C. Router
    D. Hub

13. **Routers operate on which OSI layer?**
    A. Transport
    B. Network
    C. Session
    D. Data Link

14. **What is the primary function of a firewall?**
    A. Speed up connection
    B. Filter network traffic
    C. Provide IP addresses
    D. Store files

15. **Firewalls can block:**
    A. Outgoing only
    B. Incoming only
    C. Both incoming and outgoing
    D. None

16. **The first step in a TCP 3-way handshake is:**
    A. SYN
    B. ACK
    C. SYN-ACK
    D. FIN

17. **If the last ACK in handshake is lost, the sender:**
    A. Closes connection
    B. Retransmits SYN
    C. Retransmits ACK
    D. Waits or times out

18. **Which delay is caused by the time to push data into the link?**
    A. Propagation
    B. Queuing
    C. Transmission
    D. Processing

19. **Congestion mainly affects which type of delay?**
    A. Processing
    B. Propagation
    C. Queuing
    D. Transmission

20. **IPv6 address size is:**
    A. 32-bit
    B. 64-bit
    C. 128-bit
    D. 256-bit

21. **What is a benefit of using private IPs?**
    A. Global accessibility
    B. Cost saving
    C. Easy routing
    D. No need of NAT

22. **Can a private IP access the Internet directly?**
    A. Yes
    B. No

23. **What is the first thing that happens when we visit www.google.com?**
    A. TCP connection
    B. DNS resolution
    C. HTTP request
    D. IP assignment

24. **Which is an example of one-to-many communication?**
    A. Unicast
    B. Multicast
    C. Anycast
    D. Broadcast

25. **Which communication sends data to all nodes in the network?**
    A. Unicast
    B. Broadcast
    C. Multicast
    D. Anycast

26. **A VPN provides:**
    A. Physical security
    B. Encrypted remote access
    C. Faster internet
    D. Static IP

27. **Data at the Network layer is called:**
    A. Frame
    B. Segment
    C. Packet
    D. Bit

28. **Subnetting is used to:**
    A. Combine Ips
    B. Allocate bandwidth
    C. Divide IP space efficiently
    D. Reduce security

29. **DNS converts:**
    A. IP to MAC
    B. Domain to IP
    C. Port to MAC
    D. IP to Domain

30. **Which DNS query type returns the full IP resolution to the client?**
    A. Recursive
    B. Iterative
    C. Redundant
    D. Cache-only

31. **MAC address is unique to:**
    A. IP
    B. User
    C. Device
    D. Port

32. **IP address is used for:**
    A. Local communication only
    B. Internet routing
    C. Encryption
    D. Hardware control

33. **Flow control prevents:**
    A. Packet loss from congestion
    B. Address resolution
    C. Network discovery
    D. IP conflict

34. **Which protocol provide no flow control or error correction?**
    A. TCP
    B. UDP
    C. FTP
    D. ICMP

35. **Session layer is responsible for:**
    A. Logical addressing
    B. End-to-end delivery
    C. Dialog control
    D. Data encryption

ANSWER KEY									
1.	(C)	2.	(B)	3.	(A)	4.	(C)	5.	(B)
6.	(C)	7.	(C)	8.	(A)	9.	(B)	10.	(B)
11.	(C)	12.	(C)	13.	(B)	14.	(B)	15.	(C)
16.	(A)	17.	(D)	18.	(C)	19.	(C)	20.	(C)
21.	(B)	22.	(B)	23.	(B)	24.	(B)	25.	(B)
26.	(B)	27.	(C)	28.	(C)	29.	(B)	30.	(A)
31.	(C)	32.	(B)	33.	(A)	34.	(B)	35.	(C)