🌐 Protocol in Computer Networks – Easy Explanation

- **Definition:** A *network protocol* is a **standardized set of rules** that decide *what*, *how*, and *when* data is communicated between devices in a network.
- **Purpose:**
  - Devices ke beech ek **common language** provide karta hai.
  - Ensure karta hai ki communication **efficient, secure, aur error-free** ho.
  - Different hardware/software hone ke bawajood, protocols ke through sab ek dusre ko samajh pate hain.
- **Analogy:** Socho tum Jaipur se ho aur koi US se hai. Agar dono apni local language bolenge to samajhna mushkil hoga. Lekin agar dono English use karein, communication easy ho jayega. Isi tarah, protocols ek **shared language** hote hain computers ke liye.

🔑 Key Features of Protocols

- **Formatting rules:** Data packets ka structure fix karta hai.
- **Error handling:** Agar data corrupt ho jaye to usko detect aur correct karne ke rules deta hai.
- **Flow control:** Sender aur receiver ke speed ko balance karta hai.
- **Security:** Encryption aur authentication ke liye bhi protocols hote hain.

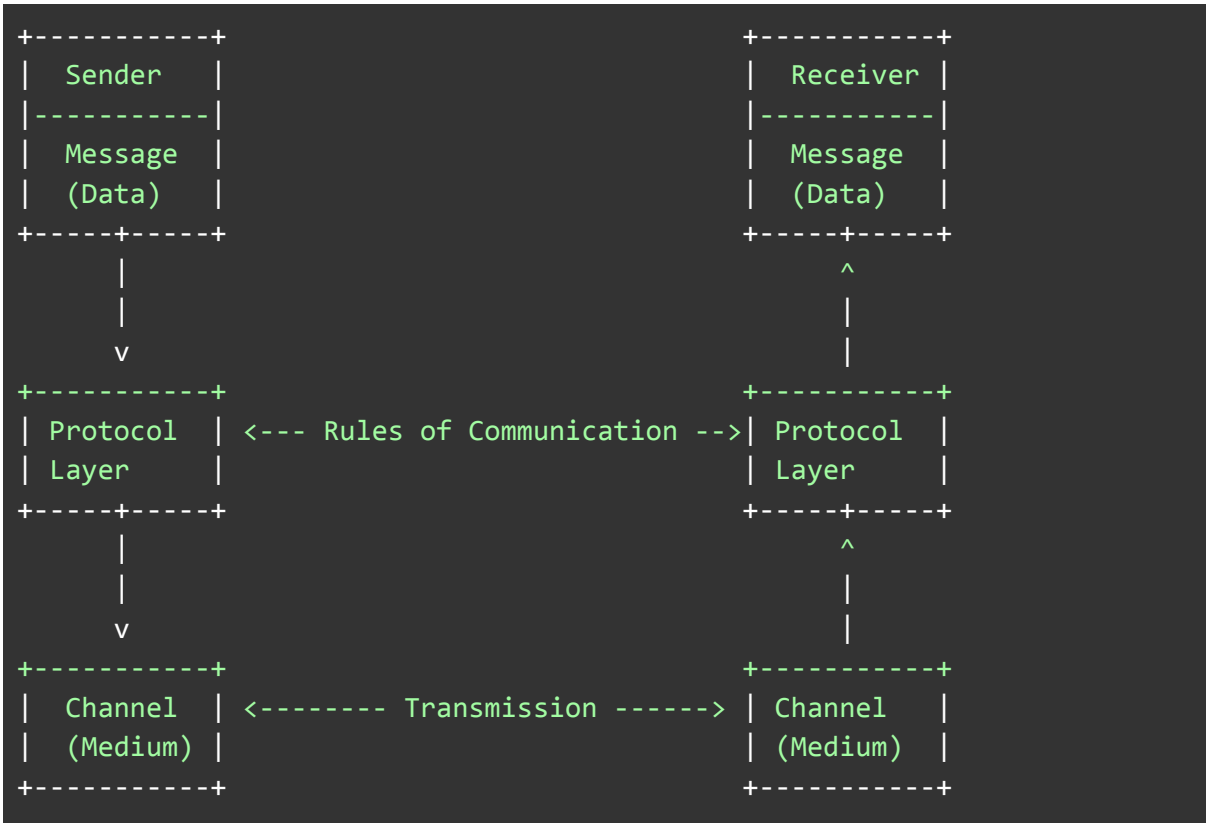
📌 Common Types of Protocols

Protocol	Use Case
HTTP/HTTPS	Web browsing aur secure communication
FTP	File transfer
SMTP/IMAP/POP3	Email communication
TCP/IP	Internet ke backbone protocols
DNS	Domain name ko IP address mein convert karta hai
Ethernet	Local area network communication

⚡ Quick Recap

- **Protocol = Rules of communication in computer networks.**
- Without protocols, devices ek dusre ko samajh hi nahi paate.
- Har protocol ek specific purpose serve karta hai (web, email, file transfer, etc.).

👉 *"Protocol ek set of rules hai jo define karta hai ki data kaise transmit aur receive hoga network mein. Ye ensure karta hai ki alag-alag devices ek dusre se reliably communicate kar saken, chahe unka hardware/software different ho."*



🔎 Explanation

- **Sender:** Data generate karta hai (jaise ek email, file, ya request).
- **Protocol Layer:** Data ko format karta hai, packet banata hai, aur rules apply karta hai (TCP/IP, HTTP, etc.).
- **Channel (Medium):** Ye physical ya wireless medium hota hai (Ethernet cable, Wi-Fi, fiber optic).
- **Receiver:** Protocol ke rules follow karke data ko decode karta hai aur original message samajhta hai.

🌐 OSI Model (Open Systems Interconnection)

- **Definition:** OSI model ek **standardized framework** hai jo explain karta hai ki computers ek dusre se kaise communicate karte hain. Ye ISO (International Organization for Standardization) ne develop kiya tha.
- **Layers (7 total):**
  - **Physical Layer** – Bits ko physical medium (cable, Wi-Fi) par bhejna.
  - **Data Link Layer** – Frames, error detection, MAC addresses.
  - **Network Layer** – IP addressing, routing.
  - **Transport Layer** – Reliable delivery (TCP/UDP).
  - **Session Layer** – Session management (login/logout).
  - **Presentation Layer** – Data translation, encryption, compression.
  - **Application Layer** – End-user services (HTTP, FTP, SMTP).

- **Purpose:**
    - Ek **reference model** hai jo communication ko samajhne aur design karne mein help karta hai.
    - Har layer ek specific responsibility rakhti hai.

🌍 TCP/IP Model (Transmission Control Protocol / Internet Protocol)

📡 Protocol Suite – Easy Explanation

- **Definition:** *Protocol suite* ka matlab hai ek **collection of related protocols** jo ek saath milkar complete communication system banate hain. Matlab ek single protocol kaam nahi karta, balki ek **group of protocols** ek coordinated way mein kaam karte hain.

- **Example:**
    - **TCP/IP Suite**: Internet chalane ke liye sabse famous protocol suite.
        - TCP (Transmission Control Protocol) → Reliable data delivery.
        - IP (Internet Protocol) → Addressing aur routing.
        - HTTP, FTP, SMTP, DNS → Application-level protocols. Ye sab ek suite ke andar aate hain aur ek dusre ke saath integrate hote hain.

- **Definition:** TCP/IP ek **protocol suite** hai jo real-world internet communication ke liye use hota hai. Ye OSI se simpler hai aur 4 layers mein divide hota hai.

- **Layers (4 total):**
    - **Application Layer** – User-level protocols (HTTP, FTP, SMTP).
    - **Transport Layer** – TCP/UDP for reliable/unreliable delivery.
    - **Internet Layer** – IP addressing, routing.
    - **Network Access Layer** – Physical transmission (Ethernet, Wi-Fi).

- **Purpose:**
    - Internet backbone protocols (TCP + IP) provide karta hai.
    - Data ko packets mein tod kar safely destination tak pahunchata hai.

🔑 Difference Between OSI and TCP/IP

Feature	OSI Model (7 Layers)	TCP/IP Model (4 Layers)
Type	Conceptual framework	Practical implementation
Layers	7 (Physical → Application)	4 (Network Access → Application)
Use	Teaching, design reference	Real-world internet communication
Developer	ISO	DARPA/US Department of Defense
Flexibility	Detailed, theoretical	Simplified, widely used

👉 ***"OSI aur TCP/IP model mein kya difference hai?"***
To tum concise answer de sakte ho: *"OSI ek 7-layer conceptual model hai jo communication ko samjhata hai, jabki TCP/IP ek 4-layer practical model hai jo internet mein use hota hai. OSI mainly reference ke liye hai, TCP/IP real-world implementation hai."*

📦 Packets in Computer Networks

- **Definition:** Packet ek **small unit of data** hota hai jo network par bheja jaata hai. Jab tum ek bada message (jaise WhatsApp text, email, ya video stream) bhejte ho, wo pura ek saath nahi jaata — usse **chhote-chhote packets** mein tod diya jaata hai.
- **Structure of a Packet:** Har packet ke andar do cheezein hoti hain:
  - **Header** → Control information (source IP, destination IP, sequence number, protocol type).
  - **Payload** → Actual data (message ka ek part).
- **Why Packets?**
  - Efficient transmission (large data ko tod kar bhejna easy hota hai).
  - Error detection (agar ek packet corrupt ho jaye to sirf wahi resend hota hai).
  - Multiplexing (ek hi network par multiple users ke packets mix hoke ja sakte hain).
- Multiplexing ek technique hai jisme **ek hi communication channel** par **multiple users ke data packets ek saath bheje jaate hain**. Matlab ek hi "road" par alag-alag

“vehicles” (packets) chal rahe hote hain, aur sab apne-apne destination tak pahunch jaate hain.

🤝 TCP 3-Way Handshake – Easy Way

1. **Step 1 – SYN (Synchronize)**
   - Client bolta hai: *“Main connection start karna chahta hoon.”*
   - Ye ek request hoti hai connection banane ke liye.
2. **Step 2 – SYN-ACK (Synchronize + Acknowledge)**
   - Server reply karta hai: *“Theek hai, main ready hoon. Tum bhi confirm karo.”*
   - Ye ek accept + confirm message hota hai.
3. **Step 3 – ACK (Acknowledge)**
   - Client bolta hai: *“Confirm, ab data bhejna start karte hain.”*
   - Ye final confirmation hota hai.

👉 Ab dono ke beech **connection establish** ho jaata hai, aur data safely transfer ho sakta hai.

📊 Simple Diagram

```
Client (Sender)                  Server (Receiver)
   | SYN (start request)  ----->     |
   |                                 |
   | <----- SYN-ACK (ready + confirm)|
   |                                 |
   | ACK (final confirm)  ---------> |
```

✅ Recap

- **SYN** = Request to start connection
- **ACK** = Confirmation
- **SYN-ACK** = Server ready + confirmation
- Ye 3 steps complete hote hi connection ban jaata hai.

🌐 TCP (Transmission Control Protocol)

- **Type:** Connection-oriented protocol
- **Reliability:** Reliable communication (acknowledgment, retransmission, error checking)
- **Ordering:** Packets hamesha correct order mein deliver hote hain
- **Handshake:** 3-way handshake se connection establish hota hai
- **Speed:** Thoda slow (extra checks ke wajah se)
- **Use Cases:** Web browsing (HTTP/HTTPS), Email (SMTP, IMAP), File transfer (FTP)

👉 **Interview line:** *"TCP ek connection-oriented protocol hai jo reliable aur ordered communication ensure karta hai, lekin thoda slow hota hai."*

⚡ UDP (User Datagram Protocol)

- **Type:** Connectionless protocol
- **Reliability:** No guarantee (no acknowledgment, no retransmission)
- **Ordering:** Packets out-of-order aa sakte hain
- **Handshake:** No handshake, direct send
- **Speed:** Fast (low overhead, small header)
- **Use Cases:** Video streaming, Online gaming, VoIP calls, DNS queries

👉 **Interview line:** *"UDP ek connectionless protocol hai jo fast communication provide karta hai, lekin unreliable hai."*

Feature	TCP	UDP
Connection	Connection-oriented	Connectionless
Reliability	Reliable (ACK, retransmit)	Unreliable (no ACK)
Ordering	Maintains order	No order guarantee
Speed	Slower	Faster
Use Cases	Web, email, file transfer	Streaming, gaming, VoIP, DNS

🎯 Interview-Ready Summary

👉 *"TCP aur UDP dono transport layer protocols hain. TCP connection-oriented hai, reliable aur ordered communication deta hai, jabki UDP connectionless hai, fast hai lekin unreliable. TCP use hota hai jab accuracy important ho, aur UDP use hota hai jab speed important ho."*

🌐 Networking (Computer Networks)

- **Definition:** Networking ka matlab hai **computers aur devices ko connect karna** taaki wo ek dusre ke saath data share kar saken.
- **Purpose:** Communication, resource sharing (printers, files), internet access.
- **Types:**
    - LAN (Local Area Network) → ek building ya campus ke andar.
    - WAN (Wide Area Network) → large area, jaise internet.
    - MAN (Metropolitan Area Network) → ek city level.

🌐 Addressing System in Computer Networks

- **Definition:** Addressing system ek **method hai jisme har device ko ek unique identity di jaati hai** taaki data sahi destination tak pahunch sake. Network mein do major types ke addresses hote hain:
    1. **Logical Address (IP Address)** → Software-level identity, change ho sakta hai.
    2. **Physical Address (MAC Address)** → Hardware-level identity, permanent hota hai.

💡 Bit

- **Bit** ka matlab hai **Binary Digit**.
- Ye computer ka **sabse chhota data unit** hai.
- Sirf **0** (off) ya **1** (on) hota hai.
- 8 bits = 1 byte.
- Example: `1010` binary = $1\times 2^3+0\times 2^2+1\times 2^1+0\times 2^0=10\ decimal$.

📊 Types of Addressing

1. **MAC Address (Physical Address)**
    - 48-bit hardware address, NIC card par fixed hota hai.
    - Format: Hexadecimal (e.g., `00:1A:2B:3C:4D:5E`).
    - Used in **Data Link Layer**.

2. **IP Address (Logical Address)**
    - Software-level address jo network par device ko identify karta hai.
    - Used in **Network Layer**.
    - Two versions: IPv4 (32-bit) and IPv6 (128-bit).

📦 Octet in Networking

- **Definition:** Octet = **8 bits ka group**. (1 bit = 0 ya 1, aur 8 bits = 1 byte).
- Networking mein jab hum **IPv4 address** likhte hain, wo 32 bits ka hota hai. Usko 4 parts mein divide kiya jaata hai → har part = **8 bits = 1 octet**.

👉 Matlab IPv4 address = 4 octets (4 × 8 bits = 32 bits).

🔎 Example: IPv4 Address

IPv4 address: `192.168.1.1`

- Binary form mein:

```
192 = 11000000
168 = 10101000
  1 = 00000001
  1 = 00000001
```

- Har number (192, 168, 1, 1) ek **octet** hai.

📊 Text-Based Diagram

```
IPv4 Address: 192.168.1.1
+---------+---------+---------+---------+
| Octet 1 | Octet 2 | Octet 3 | Octet 4 |
|  192    |  168    |    1    |    1    |
+---------+---------+---------+---------+

Each Octet = 8 bits (e.g., 192 = 11000000)
Total = 32 bits (4 × 8 bits)
```

📊 Text Diagram

```
IPv6 Address Example: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

+-------+-------+-------+-------+-------+-------+-------+-------+
| Block1| Block2| Block3| Block4| Block5| Block6| Block7| Block8|
| 2001  | 0db8  | 85a3  | 0000  | 0000  | 8a2e  | 0370  | 7334  |
+-------+-------+-------+-------+-------+-------+-------+-------+

Each Block = 16 bits = 2 octets
Total = 8 blocks × 16 bits = 128 bits = 16 octets
```

📊 IPv4 vs IPv6 – Important Interview Points

Feature	IPv4	IPv6
Address Size	32-bit (4 octets)	128-bit (16 octets)
Format	Dot-decimal (e.g., 192.168.1.1)	Colon-hexadecimal (e.g., 2001:db8::1)
Total Addresses	~4.3 billion (2^32)	~3.4 × 10^38 (2^128) virtually unlimited
Configuration	Manual / DHCP	Auto-configuration supported (SLAAC)
Security	Optional (IPSec not mandatory)	Built-in (IPSec mandatory)
Broadcast	Supported	Not supported (uses multicast)
Usage	Still widely used, limited	Modern backbone, solves IPv4 exhaustion

🌐 DHCP in Computer Networks – Detailed Explanation

1. Definition

- DHCP = **Dynamic Host Configuration Protocol**.
- Ye ek **client-server protocol** hai jo devices (PC, phone, printer, etc.) ko automatically assign karta hai:
    - **IP Address**
    - **Subnet Mask**
    - **Default Gateway**
    - **DNS Server Address**
    - Aur other TCP/IP settings.

👉 Matlab: Jab tum Wi-Fi connect karte ho, tumhe IP aur settings automatically mil jaati hain — ye kaam DHCP karta hai.

🌐 Subnet Mask – Definition

- **Subnet Mask ek 32-bit number hota hai (IPv4 ke liye)** jo batata hai ki IP address ka kaunsa part **Network ID** hai aur kaunsa part **Host ID** hai.
- Matlab: Ye ek **filter** ki tarah kaam karta hai jo IP address ko do parts mein todta hai:
  - **Network Portion** → identify karta hai network.
  - **Host Portion** → identify karta hai device (host) us network ke andar.

🔎 Example

IP Address: `192.168.1.10` Subnet Mask: `255.255.255.0`

- Binary form:

```
IP Address:    11000000.10101000.00000001.00001010
Subnet Mask:   11111111.11111111.11111111.00000000
```

- Result:
  - **Network Part:** 192.168.1
  - **Host Part:** 10

👉 Matlab: Network = `192.168.1.0` aur host = `10`.

📊 Subnet Mask Common Values

- `255.0.0.0` → Class A (large networks)
- `255.255.0.0` → Class B (medium networks)
- `255.255.255.0` → Class C (small networks)

📦 Text-Based Diagram

```
IP Address:    192.168.1.10
Subnet Mask:   255.255.255.0

+------------------+---------------+
|   Network Part   |   Host Part   |
|   192.168.1      |       10      |
+------------------+---------------+
```

🌐 Subnet Mask ka Role

- **Network Part:** Batata hai ki device kis **network** ke andar hai.
- **Host Part:** Batata hai ki us network ke andar **kaunsa device** hai.

👉 Matlab: Network part = colony ka naam, Host part = ghar ka number.

📱 Real-World Example (Wi-Fi Router)

Maan lo tumhare ghar ka Wi-Fi router hai:

- Router ka **IP Address:** `192.168.1.1`
- Subnet Mask: `255.255.255.0`

Ab tumhare ghar ke devices:

- Laptop: `192.168.1.5`
- Phone: `192.168.1.10`
- Smart TV: `192.168.1.20`

Breakdown:

- **Network Part (192.168.1)** → sab devices ek hi Wi-Fi network mein hain.
- **Host Part (5, 10, 20)** → alag-alag devices ko identify karta hai.

👉 Isse router ko pata chalta hai ki ye sab ek hi network ke andar hain, aur har device ka apna unique host number hai.

✅ Interview-Ready Line

“Subnet Mask ek number hai jo IP address ko network aur host part mein divide karta hai. Ye mostly automatically assign hota hai DHCP ke through, jaise Wi-Fi connect karte hi tumhe IP ke saath subnet mask bhi mil jaata hai.”

🌐 Default Gateway – Definition

- **Default Gateway ek device (usually router) hota hai jo ek network ke devices ko dusre networks ya internet tak pahunchne ka rasta deta hai.**
- Agar tumhara computer ek IP packet bhejna chahta hai jo **local network ke bahar** hai, to wo packet **default gateway** ko forward kar diya jaata hai.

👉 Matlab: Default Gateway = **network ka exit door**.

🔎 Real-World Example

Maan lo tumhare ghar ka Wi-Fi network hai:

- Laptop ka IP: `192.168.1.5`
- Phone ka IP: `192.168.1.10`
- Router ka IP: `192.168.1.1`

Yahaan:

- Agar laptop phone ko data bhejta hai → direct local communication hota hai (same network).
- Agar laptop Google.com open karta hai → wo local network ke bahar hai.
    - Laptop packet ko **Default Gateway (192.168.1.1)** bhejta hai.
    - Router (gateway) us packet ko internet par forward karta hai.

📊 Text-Based Diagram

```
[ Laptop 192.168.1.5 ] ----\
                            \
[ Phone 192.168.1.10 ] ------> Local Network (192.168.1.x)
                            /
[ Smart TV 192.168.1.20 ]--/

         |
         |  Default Gateway (Router 192.168.1.1)
         v
      Internet (Google, YouTube, etc.)
```

🎯 Interview-Ready Line

👉 *"Default Gateway ek router hota hai jo local network ke devices ko dusre networks ya internet tak connect karta hai. Agar destination local network ke bahar hai, packet default gateway ko bheja jaata hai jo usse forward karta hai."*

🌐 DNS – Full Concept Explained

1. Definition

- DNS = **Domain Name System**.
- Ye ek **hierarchical aur distributed naming system** hai jo domain names ko IP addresses mein translate karta hai.
- Har device internet par ek unique IP address rakhta hai, aur DNS us IP tak reach karne ka rasta banata hai.

2. Why DNS is Needed

- Humans ke liye names yaad rakhna easy hai (`youtube.com`).
- Machines ke liye IP addresses easy hote hain (`142.250.190.14`).
- DNS bridge ka kaam karta hai → **Name → IP conversion**.

👉 Without DNS, tumhe har website ka IP yaad rakhna padta.

🌐 DNS Working – Easy Flow

1. **User Request:** Tum browser mein `www.google.com` type karte ho.
2. **Local Check:** Browser/OS pehle apna **cache** check karta hai (agar pehle IP resolve hua ho).
3. **DNS Resolver (ISP ka server):** Agar cache mein nahi mila, tumhari request tumhare **ISP (Internet Service Provider)** ke DNS resolver tak jaati hai.
    - **DNS Resolver mtlb:** Ek server jo tumhari query ko process karta hai aur IP address dhoondta hai.
4. **Root Server:** Resolver root DNS server se poochta hai → “Google.com kahan hai?”
5. **TLD Server (Top Level Domain):** Root server usse `.com` ke **TLD (Top Level Domain)** server ka address deta hai.
    - **TLD mtlb:** Wo server jo `.com`, `.org`, `.net` jaise extensions handle karta hai.
6. **Authoritative Server:**
    - Ye **final server hota hai jisme domain ka actual IP address stored hota hai**.
    - Example: `google.com` ka IP (142.251.223.110) yahin se milta hai.
7. **Response:** Authoritative server IP address return karta hai (e.g., `142.251.223.110`).
8. **Browser Connects:** Browser ab us IP ke server se connect karke website load karta hai.

📊 Easy Text Diagram

```
User (Browser)
      ↓
DNS Resolver (ISP server)
      ↓
Root Server
      ↓
TLD Server (.com)
      ↓
Authoritative Server (google.com)
      ↓
Returns IP
      ↓
Browser connects to Website
```

🌐 VPN – Detailed Concept in Computer Networks

1. Definition

- VPN = **Virtual Private Network**.
- Ye ek **encrypted connection** banata hai tumhare device aur internet ke beech.
- Isse tumhari identity aur data secure rehti hai.

👉 Matlab: VPN ek **private tunnel** hai jo public internet ke upar banaya jaata hai.

🔐 Encrypted Connection – Simple Meaning

- **Encryption** = Data ko ek **secret code** mein convert karna.
- Jab tum internet par koi information bhejte ho (password, message, payment details), wo **plain text** mein na jaake **scrambled form** mein jaati hai.
- Sirf sender aur receiver ke paas **key** hoti hai jo us scrambled data ko wapas normal form mein convert kar sakti hai.

👉 Matlab: Agar koi beech mein data intercept kare, to usse sirf **random symbols** dikhenge, actual message nahi.

🌐 VPN Context

- Jab tum VPN use karte ho, tumhara **internet traffic encrypt ho jaata hai**.
- Matlab tum jo websites visit karte ho, passwords, ya data bhejte ho → sab ek **secret tunnel** ke andar jaata hai.
- ISP (Internet Service Provider) ya hackers public Wi-Fi par tumhara actual data nahi dekh sakte.

VPN server kisi company ka hota hai, to kya wo hamara data dekh sakta hai?

🌐 VPN Server Reality

- **Yes, technically possible:** VPN server tumhara traffic decrypt karke internet par bhejta hai. Matlab agar VPN provider chahe, to wo tumhari browsing activity dekh sakta hai.
- **But encryption ka role:** Tumhare device se VPN server tak jo data jaata hai wo **encrypted tunnel** mein hota hai.
    - ISP (Internet Service Provider) ya hackers public Wi-Fi par tumhara data **nahi dekh sakte**.
    - Lekin VPN server ke paas tumhara decrypted traffic aa sakta hai.

🎯 Interview-Ready Line

👉 *"VPN ek Virtual Private Network hai jo encrypted tunnel banata hai device aur internet ke beech. Ye IP address hide karta hai, privacy protect karta hai, aur restricted content access karne deta hai. VPN widely use hota hai corporate remote access aur secure browsing ke liye."*

🌐 Router vs Gateway – Difference

1. Router

- Router ek **network device** hai jo **do alag networks ko connect karta hai** (mostly LAN ↔ Internet).
- Ye **packets ko forward** karta hai based on **IP addresses** aur routing tables.
- Example: Tumhare ghar ka Wi-Fi router → ek taraf tumhara local network (192.168.1.x), dusri taraf ISP/internet.

👉 Router = **device** jo traffic ko route karta hai.

2. Gateway

- Gateway ek **logical concept** hai, jo ek network ka **exit point** hota hai.
- Jab tumhara device ko local network ke bahar jaana ho (internet ya dusre subnet), wo packet **default gateway** ko bhejta hai.
- Gateway zaroori nahi ki hamesha router ho — kabhi kabhi firewall, proxy server, ya ek special device bhi gateway ban sakta hai.

👉 Gateway = **exit door** of the network.

🎯 Interview-Ready Line

👉 *"Router ek physical device hai jo networks ko connect karta hai aur packets route karta hai. Gateway ek logical exit point hai jo batata hai ki local network ke bahar traffic kahan bhejna hai. Ghar ke network mein router ka IP hi default gateway hota hai."*

🌐 Firewall – Definition

- **Firewall ek security system hai jo network traffic ko filter karta hai.**
- Ye decide karta hai ki **kaunsa data andar aane de** aur **kaunsa block kare**.
- Isse tumhara network **unauthorized access** aur **malicious traffic** se safe rehta hai.

👉 Matlab: Firewall ek **security guard** hai jo tumhare network ke gate par khada hai.

🔎 Firewall ka Role

1. **Packet Filtering:** Har packet check hota hai ki allowed hai ya nahi.
2. **Access Control:** Sirf trusted IPs, ports, aur protocols ko allow karta hai.
3. **Protection:** Hackers, malware, aur suspicious traffic ko block karta hai.
4. **Monitoring:** Network activity ko continuously monitor karta hai.

📱 Real-World Example

- Tumhare ghar ka Wi-Fi router mein ek **basic firewall** hota hai.
- Agar koi unknown device tumhare network ko access karna chahe, firewall usse block kar deta hai.
- Office mein firewall aur advanced hota hai → wo decide karta hai ki employees sirf company-approved websites hi access kar paayein.

🎯 Interview-Ready Line

👉 *"Firewall ek security system hai jo network traffic ko filter karta hai aur unauthorized access ko block karta hai. Ye ek security guard ki tarah kaam karta hai jo decide karta hai ki kaunsa data andar aayega aur kaunsa nahi."*

🌐 *Network Topology – Definition*

- ***Topology*** *= Network ka* ***structure*** *(jaise ghar ka map).*
- *Do tarah ke hote hain:*
  - ***Physical Topology:*** *Devices aur cables ka actual layout.*
  - ***Logical Topology:*** *Data ka flow kaise hota hai, chahe physical layout kuch bhi ho.*

📊 Types of Network Topology

1. Bus Topology

- Ek single backbone cable hoti hai, sab devices usse connect hote hain.
- Agar cable fail ho jaaye → pura network down.

```
Device -- Device -- Device -- Device
          |
       Main Cable (Bus)
```

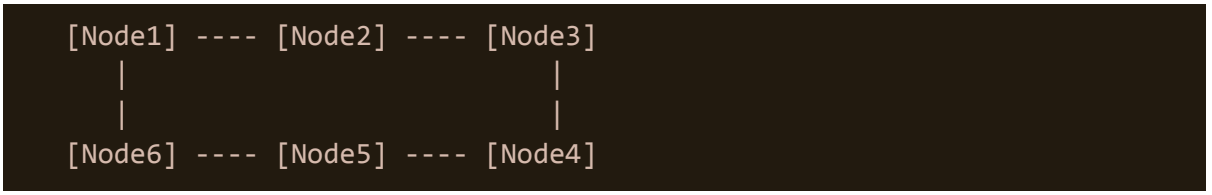
2. Star Topology

- Sab devices ek **central hub/switch** se connect hote hain.
- Easy to manage, but agar hub fail ho jaaye → pura network down.

```
           Device
             |
Device -- Hub/Switch -- Device
             |
           Device
```

3. Ring Topology

- Devices ek **circular loop** mein connected hote hain.
- Data ek direction mein circulate karta hai (clockwise ya anticlockwise).
- Agar ek link fail ho jaaye, pura ring disturb ho sakta hai (unless dual ring use ho).

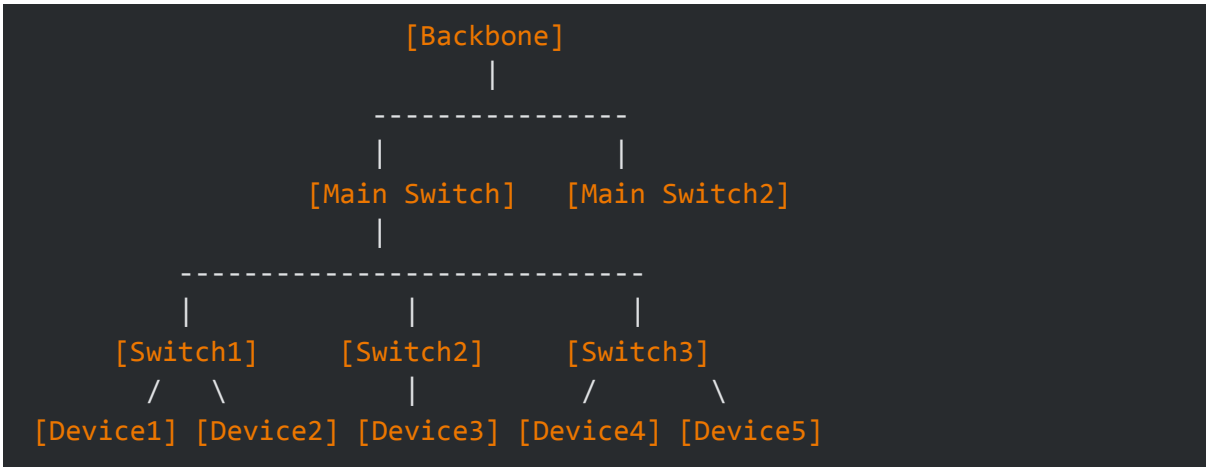


4. Mesh Topology

- Har device directly har dusre device se connected hota hai.
- High reliability, but costly (bohot cables chahiye).

5. Tree Topology

- Combination of **Star + Bus**.
- Ek main backbone hota hai, aur usse multiple star networks connect hote hain



🔎 Explanation

- **Backbone:** Central line/network jisme sab switches connect hote hain.
- **Main Switches:** Backbone se directly connected hote hain, aur niche ke switches ko manage karte hain.
- **Switches:** Har switch apne devices ko connect karta hai.
- **Devices:** End nodes (PCs, laptops, mobiles, printers) jo actual users hote hain.

👉 Ye ek **hierarchical structure** hai → backbone → switches → devices.

6. Hybrid Topology

- Jab ek network mein multiple topologies mix hoti hain (jaise star + ring).
- Flexible aur scalable

✅ Interview-Ready Line

👉 *"Topology network ka layout hota hai. Common types hain Bus, Star, Ring, Mesh, Tree, aur Hybrid. Har topology ke apne advantages aur disadvantages hote hain, aur choice depend karti hai network size aur requirement par."*

🌐 Hub vs Switch – Easy Explanation

🔹 Hub

- **Layer:** Physical Layer (OSI Layer 1).
- **Working:** Jo bhi data aata hai, wo sabhi ports par broadcast kar deta hai.
- **Efficiency:** Kam efficient, kyunki unnecessary traffic generate hota hai.
- **Security:** Low, kyunki data sabko visible hota hai.
- **Use Case:** Small, simple networks (ab modern networks mein kam use hota hai).

🔹 Switch

- **Layer:** Data Link Layer (OSI Layer 2).
- **Working:** Data ko destination device ke **MAC address** ke basis par forward karta hai.
- **Efficiency:** High, kyunki sirf intended device ko data milta hai.
- **Security:** Better, kyunki unnecessary devices ko data nahi milta.
- **Use Case:** Modern LANs, offices, homes.

📊 Tabular Comparison

Feature	Hub	Switch
OSI Layer	Physical Layer (Layer 1)	Data Link Layer (Layer 2)
Data Transfer	Broadcast to all devices	Forward to specific MAC address
Efficiency	Low (extra traffic)	High (optimized traffic)
Security	Weak (sabko data visible)	Stronger (sirf target device)
Cost	Cheaper	Thoda costly but efficient
Use Today	Rarely used	Widely used in modern networks

🎯 Interview-Ready Line

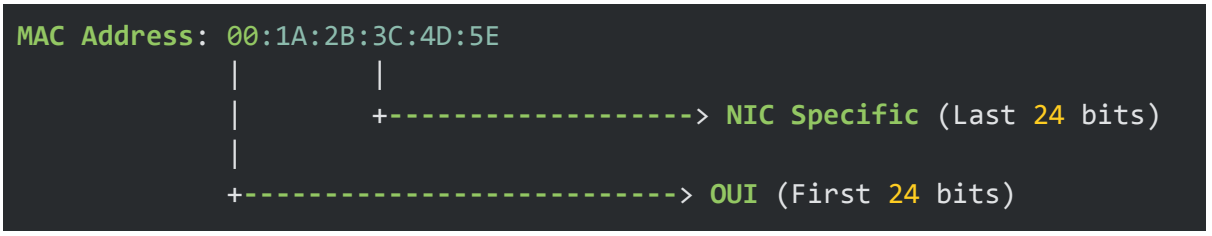
👉 *"Hub ek simple device hai jo data sabhi devices ko broadcast karta hai, jabki Switch ek intelligent device hai jo MAC address ke basis par sirf intended device ko data forward karta hai. Isliye modern networks mein Switch use hota hai."*

🌐 MAC Address – Detailed Concept

1. Definition

- **MAC = Media Access Control Address**.
- Ye ek **48-bit (6 bytes)** ka unique identifier hota hai jo har NIC (Network Interface Card) ke saath manufacturer assign karta hai.
- Format: **Hexadecimal** (base 16) → Example: `00:1A:2B:3C:4D:5E`.

🔐 Structure of MAC Address (48-bit = 6 Bytes)



🔎 Breakdown

- **OUI (Organizationally Unique Identifier)** → First 24 bits (3 bytes)
  - Manufacturer ka code hota hai.
  - Example: `00:1A:2B` → Dell ka OUI.
- **NIC Specific (Device Identifier)** → Last 24 bits (3 bytes)
  - Manufacturer apne har device ko unique serial number assign karta hai.
  - Example: `3C:4D:5E` → Specific NIC ka unique ID.

🎯 Interview-Ready Line

"MAC address 48-bit ka hota hai jisme pehle 24 bits OUI (manufacturer code) hote hain aur last 24 bits NIC-specific unique serial number hote hain. Is combination ki wajah se har device ka MAC globally unique hota hai."

Why Both IP & MAC Are Needed

- **MAC Address:** Permanent hardware identity (Layer 2).
- **IP Address:** Logical, changeable identity (Layer 3).
- Together → Device ko uniquely identify aur route karne ke liye dono zaroori hain

🔐 Role of MAC Address in Networking

1. LAN Communication

- Jab ek LAN (Local Area Network) mein devices connected hote hain, **Switch** apni **MAC Address Table** maintain karta hai.
- Ye table batata hai ki **kaunsa MAC kis port par connected hai**.
- Jab ek packet aata hai, switch uska **destination MAC** check karta hai aur packet **sirf us port** par bhejta hai jahan wo device connected hai. 👉 Isse unnecessary broadcast avoid hota hai aur network fast banta hai.

2. ARP (Address Resolution Protocol)

- Devices communication ke liye IP address use karte hain, lekin LAN mein delivery ke liye **MAC address chahiye hota hai**.
- ARP ka role:
    - Agar Laptop ko Mobile ka IP pata hai (`192.168.1.10`), to wo ARP query bhejta hai: *"Is IP ka MAC address kya hai?"*
    - Mobile reply karta hai: *"Mera MAC = FF:EE:DD:CC:BB:02"*.
    - Ab Laptop packet ko us MAC par bhej sakta hai. 👉 ARP = IP → MAC conversion for local delivery.

3. Security (MAC Filtering)

- Router/switch par **MAC filtering** enable kiya ja sakta hai.
- Matlab: Sirf specific MAC addresses ko network access allowed hoga.
- Example: Agar tum apne Wi-Fi par MAC filtering lagao, to sirf tumhare laptop aur mobile connect kar paayenge, baaki devices block ho jaayenge.

👉 Ye ek **basic access control** technique hai.

🎯 Interview-Ready Line

👉 *"MAC address networking mein teen major roles play karta hai: LAN communication (switches MAC table use karte hain), ARP (IP ko MAC mein convert karta hai for local delivery), aur Security (MAC filtering se access control hota hai)."*

📊 Example Table (Switch MAC Table)

```
+-----------+---------------------------+
| Port No.  | MAC Address               |
+-----------+---------------------------+
| Port 1    | AA:BB:CC:DD:EE:01 (Laptop) |
| Port 2    | FF:EE:DD:CC:BB:02 (Mobile) |
| Port 3    | 11:22:33:44:55:66 (Printer)|
+-----------+---------------------------+
```

🔎 Port ka Matlab

- **Switch ke physical connectors (Ethernet jahan cable lagti hai)** ko **ports** kehte hain.
- Har port par ek **device** connect hota hai (jaise Laptop, Mobile, Printer).
- Jab device connect hota hai, uska **MAC address** switch seekh leta hai aur apni **MAC table** mein note kar leta hai.

👉 Matlab: **Port = Switch ka connection point** aur **MAC = us port par connected device ka hardware address**.

- **Port 1** → Laptop ka MAC
- **Port 2** → Mobile ka MAC
- **Port 3** → Printer ka MAC

🎯 Interview-Ready Line

👉 *"Switch apni MAC table mein store karta hai ki kaunsa MAC address kis physical port par connected hai. Port ka matlab hai switch ka connection point, aur MAC us port par connected device ka unique hardware address."*

📊 Common Network Protocols – Table Format

Protocol	Layer (OSI)	Working / Function	Use Case
TCP (Transmission Control Protocol)	Transport Layer	Connection-oriented, reliable delivery, 3-way handshake, retransmission of lost packets	Web browsing, emails, file transfer
UDP (User Datagram Protocol)	Transport Layer	Connectionless, fast, no guarantee of delivery	Video streaming, online gaming, VoIP
IP (Internet Protocol)	Network Layer	Provides addressing & routing, packets forwarded based on IP addresses	Internet backbone communication
HTTP (Hypertext Transfer Protocol)	Application Layer	Client (browser) requests data, server responds with HTML/CSS/JS	Web browsing
FTP (File Transfer Protocol)	Application Layer	Client-server model, files uploaded/downloaded	Website file management, data sharing
SMTP (Simple Mail Transfer Protocol)	Application Layer	Sends emails from client → mail server → destination server	Email sending
DNS (Domain Name System)	Application Layer	Converts domain names into IP addresses	Accessing websites by name instead of IP

🎯 Interview-Ready Line

👉 ***"Protocols ek set of rules hote hain jo network communication ko define karte hain. TCP reliable hai, UDP fast hai, IP addressing karta hai, HTTP web browsing ke liye, FTP file transfer ke liye, SMTP email sending ke liye, aur DNS domain ko IP mein convert karta hai."***

🌐 HTTP (HyperText Transfer Protocol)

- **Layer: Application Layer protocol.**
- **Working:**
  - **Browser (client) server ko request bhejta hai (GET/POST).**
  - **Server response bhejta hai (HTML, CSS, JS).**
- **Nature: Connection stateless hota hai (har request independent hoti hai).**
- **Use Case: Web browsing.**
- **Example: Jab tum `http://example.com` open karte ho, data bina encryption ke transfer hota hai.**

🔒 HTTPS (HyperText Transfer Protocol Secure)

- **Layer: Application Layer + Security (TLS/SSL).**
- **Working:**
  - **Same as HTTP, but data transfer encrypted hota hai.**
  - **TLS/SSL certificates use hote hain jo ensure karte hain ki communication secure hai.**
- **Nature: Secure, prevents eavesdropping & tampering.**
- **Use Case: Secure web browsing, online banking, shopping.**
- **Example: Jab tum `https://example.com` open karte ho, browser aur server ke beech data encrypted hota hai.**

👉 Difference: HTTP = normal communication, HTTPS = secure communication with encryption.

📊 Easy Text Diagram

HTTP vs HTTPS

```
Browser ----HTTP----> Web Server   (Data in plain text)
Browser ----HTTPS---> Web Server   (Data encrypted with SSL/TLS)
```

🔒 SSL/TLS – Short Explanation

- **SSL (Secure Sockets Layer) aur TLS (Transport Layer Security) dono encryption protocols hain jo internet communication ko secure banate hain.**
- **Ye ensure karte hain ki data encrypted form mein transfer ho, taaki koi attacker usse read ya modify na kar sake.**
- **TLS SSL ka updated aur more secure version hai (aajkal mostly TLS hi use hota hai).**
- **Example: Jab tum `https://example.com` website open karte ho, browser aur server ke beech communication SSL/TLS ke through encrypted hota hai.**

🎯 Interview-Ready Line

👉 ***"SSL/TLS ek encryption protocol hai jo client aur server ke beech secure communication provide karta hai. HTTPS websites SSL/TLS use karti hain taaki data safe rahe."***

🌐 OSI Model – 7 Layers Detailed Explanation

1. Physical Layer

- **Kaam: Bits (0s & 1s) ko physical medium (cable, fiber, Wi-Fi signals) par bhejna.**
- **Example: Tumhare laptop se Wi-Fi router tak electrical signals / radio waves jaate hain.**
  👉 ***Socho wire ek road hai aur bits chhoti gaadiyan jo us road par chalti hain.***

📦 Frame (Data Link Layer ka unit)

🔎 Definition

- **Frame = Data Link Layer ka packet**.
- Ye ek **structured container** hota hai jisme data + addressing + error-checking information hoti hai.
- Physical layer bits bhejti hai, lekin Data Link layer un bits ko **frame** ke form mein organize karti hai.

📊 Text Diagram – Frame Structure

```
+-----------+-----------+-----------+
| Header    | Payload   | Trailer   |
+-----------+-----------+-----------+
```

Breakdown:

- **Header:** Source MAC, Destination MAC, control info
- **Payload:** Actual data (from upper layers)
- **Trailer:** Error detection bits (CRC)

Short Example

Socho tum courier bhejte ho:

- **Header** = Sender & Receiver address likha hua.
- **Payload** = Parcel (actual data).
- **Trailer** = Seal/checksum jo ensure kare ki parcel damage nahi hua.

Interview-Ready Line

👉 *"Frame ek Data Link Layer ka unit hai jisme header (MAC addresses), payload (data), aur trailer (error-checking) hota hai. Ye ensure karta hai ki data LAN mein sahi device tak aur bina error ke pahunche."*

🔎 CRC (Cyclic Redundancy Check)

- **Definition:** CRC ek **error detection technique** hai jo Data Link Layer mein use hoti hai.
- **Kaam:** Jab data (frame) bheja jaata hai, sender ek **mathematical calculation (polynomial division)** karke ek **CRC value (checksum)** generate karta hai aur frame ke trailer mein attach kar deta hai.
- **Receiver side:** Jab data receive hota hai, receiver bhi wahi calculation karta hai.
    - Agar calculated CRC = received CRC → Data sahi hai.
    - Agar mismatch → Error detect ho gaya.

📊 Text Diagram – CRC Flow

```
Sender Side:
[Data] ---> CRC Calculation ---> [Data + CRC bits] ---> Send

Receiver Side:
[Data + CRC bits] ---> CRC Recalculation ---> Compare
        | Match? Yes → Accept
        | Match? No  → Error Detected
```

2. Data Link Layer

- **Kaam:** Frames banata hai, error detection karta hai, aur MAC addresses use karta hai.
- **Example:** Switch apni MAC table ke basis par decide karta hai ki packet kis port par bhejna hai.

3. Network Layer

- **Kaam:** IP addressing aur routing karta hai.
- **Example:** Router tumhare packet ko internet par sahi path se forward karta hai (source IP → destination IP).
  👉 *Socho courier company jo city-to-city address ke basis par parcel deliver karti hai.*

4. Transport Layer

- **Kaam:** Reliable delivery (TCP) ya fast delivery (UDP).
- **Example:**
  - TCP: Agar tum file download karte ho → har packet verify hota hai, missing packet resend hota hai.
  - UDP: Agar tum YouTube video dekhte ho → fast stream hota hai, missing packet ignore ho jaata hai.

5. Session Layer

- **Kaam:** Session establish, maintain aur terminate karta hai.
- **Example:** Jab tum Gmail login karte ho → ek session create hota hai, logout karte hi session close ho jaata hai. 👉 *Socho ek meeting room jisme entry aur exit ka record rakha jaata hai.*

6. Presentation Layer

- **Kaam:** Data translation, encryption, compression.
- **Example:**
  - Jab tum HTTPS site open karte ho → data encrypt hota hai (SSL/TLS).
  - Jab tum image bhejte ho → compression hota hai (JPEG/PNG).

7. Application Layer

- **Kaam:** End-user services provide karta hai (HTTP, FTP, SMTP, DNS).
- **Example:**
  - HTTP → Web browsing
  - FTP → File transfer
  - SMTP → Email sending
  - DNS → Domain name → IP conversion
    👉 *Socho tumhara mobile app jo directly tumhe service deta hai.*

📊 Easy Text Diagram (Flow of Data)

```
User (Application Layer: HTTP Request)
        ↓
Presentation Layer (Encrypt/Compress Data)
        ↓
Session Layer (Manage Login Session)
        ↓
Transport Layer (TCP/UDP Packets)
        ↓
Network Layer (IP Addressing & Routing)
        ↓
Data Link Layer (Frames + MAC Address)
        ↓
Physical Layer (Bits on Cable/Wi-Fi)
```

🎯 Interview-Ready Line

👉 *"OSI Model ek 7-layer architecture hai jo network communication ko step-by-step define karta hai. Physical layer bits bhejti hai, Data Link frames aur MAC handle karta hai, Network IP addressing karta hai, Transport reliable delivery karta hai, Session login/logout manage karta hai, Presentation data translate/encrypt/compress karta hai, aur Application end-user services provide karta hai. Ek HTTP request browser se server tak jaane ke liye ye 7 layers sequentially kaam karti hain."*

📡 Data Transmission

- **Definition:** Data transmission ka matlab hai ek device se dusre device tak information bhejna (bits/frames/packets).
- Ye transmission **direction** aur **timing** ke basis par alag-alag modes mein hota hai.

🔀 Modes of Data Transmission

1. Simplex Mode

- **Direction:** One-way communication (sirf ek taraf data flow).
- **Example:**
    - Keyboard → Computer (keyboard sirf input bhejta hai, receive nahi karta).

```
[Sender] ---> [Receiver]
```

2. Half-Duplex Mode

- **Direction:** Dono taraf data ja sakta hai, lekin **ek time par sirf ek taraf**.
- **Example:**
    - Walkie-Talkie (ek bolta hai, dusra sunta hai; phir role change hota hai).

```
[Device A] ---> [Device B]
[Device A] <--- [Device B]   (but not at same time)
```

3. Full-Duplex Mode

- **Direction:** Dono taraf data **simultaneously** ja sakta hai.
- **Example:**
    - Telephone call (tum bolte ho aur saamne wala ek saath sunta + bolta hai).

```
[Device A] <----> [Device B]   (same time both ways)
```

🎯 Interview-Ready Line

👉 *"Data transmission ke 3 modes hote hain: Simplex (one-way), Half-Duplex (two-way but one at a time), aur Full-Duplex (two-way simultaneously). Modern networks full-duplex use karte hain taaki efficiency aur speed high ho."*

📡 Types of Data Transmission

📊 Comparison Table

Feature	Serial Transmission	Parallel Transmission
Data Flow	One bit at a time	Multiple bits at a time
Speed	Slower	Faster
Cost	Cheaper (less wires)	Costly (more wires)
Distance	Long distance (less error)	Short distance (signal skew)
Examples	USB, Internet	CPU ↔ RAM, Printer cable

🔎 Signal Skew (short distance issue in parallel transmission)

- **Definition:** Jab **parallel transmission** mein multiple bits ek saath alag-alag wires par bheje jaate hain, to sab bits ek hi time par receiver tak pahunchne chahiye.
- **Problem (Skew):** Har wire ki **length, resistance, capacitance** thodi alag ho sakti hai → is wajah se kuch bits thoda jaldi aur kuch thoda late receiver tak pahunch jaate hain.
- **Result:** Receiver ko ek hi clock cycle mein sab bits ek saath milne chahiye, lekin agar timing mismatch ho jaaye to **error** ho sakta hai.

📊 Text Diagram – Signal Skew

Sender Side (Parallel Transmission)

```
Sender sends bits simultaneously:
Bit1  ------------------------------>
Bit2  ------------------------------>
Bit3  ------------------------------>
```

👉 Ideally sab bits ek hi time par nikalte hain aur ek saath receiver tak pahunchne chahiye.

Receiver Side (Arrival Times)

```
Receiver receives:
Bit1  -----> arrives at t = 10 ns
Bit2  -----> arrives at t = 12 ns
Bit3  -----> arrives at t = 11 ns
```

Result

```
Expected: All bits arrive together
Actual:   Bits arrive at different times
=> Timing mismatch = Signal Skew
```

👉 *"Signal skew ek timing mismatch hai jo parallel transmission mein hota hai jab alag-alag wires par bits ek hi time par receiver tak nahi pahunchte. Isliye parallel transmission short distance ke liye hi reliable hai."*

🔄 NAT – Easy Explanation (Network Address Translation)

- **Problem:** Internet par har device ko ek **unique public IP** chahiye hota hai. Lekin ghar/office ke andar devices ke paas **private IP** hota hai (192.168.x.x, 10.x.x.x).
- **Solution:** NAT router ke andar hota hai jo **private IP → public IP** conversion karta hai.
- Matlab: Tumhare ghar ke 5 devices ek hi **public IP** use karke internet access karte hain.

📊 Text Diagram – NAT Flow

```
Laptop (192.168.1.5) -----\
Phone  (192.168.1.10) ----> Router NAT ----> Internet (Public IP: 203.0.113.25)
TV     (192.168.1.20) ----/
```

- **Inside Home:** Devices ke paas private IP hai (192.168.x.x).
- **Router NAT:** Sabko ek hi public IP assign karta hai (203.0.113.25).
- **Internet Side:** Bahar se lagta hai ki ek hi device hai, but andar multiple devices chal rahe hote hain.

Interview-Ready Line

👉 *"NAT ek technique hai jo private IP addresses ko public IP mein translate karti hai. Isse ek hi public IP ke through multiple devices internet access kar sakte hain. Ghar ke Wi-Fi router NAT use karta hai."*

🔢 Port Number – Easy Definition

- **Port number** ek **logical gate** hai jo batata hai ki ek IP address ke andar **kaunsi service/application** chal rahi hai.
- Ek IP address ek ghar ka address jaisa hai, aur port number us ghar ke andar **specific room/service** jaisa hai.

🎯 Real-World Example Flow (Web Browsing)

```
Browser (Client)
   |
   |  Request to Server IP: 142.250.190.14
   |  Port: 443 (HTTPS)
   v
Google Server (listening on port 443)
   |
   |  Encrypted response (HTML, CSS, JS)
   v
Browser displays Google homepage
```

👉 Agar tum `https://google.com` open karte ho, tumhara browser **IP + Port 443** par connect karta hai. Agar port 443 block ho jaaye, tum secure site open nahi kar paoge.

Interview-Ready Line

👉 *"Port number ek service identifier hai jo ek IP address ke andar specific application ko target karta hai. Jaise web browsing ke liye port 80/443, DNS ke liye port 53, aur email sending ke liye port 25 use hota hai."*

📊 Common Port Numbers with Real Use Cases

Port	Protocol	Real Use Case
80	HTTP	Jab tum **`http://example.com`** open karte ho, browser server ke port 80 par request bhejta hai. Ye normal (unencrypted) web browsing hai.
443	HTTPS	Jab tum **`https://google.com`** open karte ho, browser server ke port 443 par secure request bhejta hai. Ye encrypted communication hai (SSL/TLS).
53	DNS	Jab tum **`www.youtube.com`** type karte ho, tumhara system DNS server ke port 53 par query bhejta hai taaki domain → IP conversion ho.
25	SMTP	Jab tum Gmail se email bhejte ho, wo SMTP server ke port 25 par jaata hai. Ye email sending ke liye standard port hai.
22	SSH	Jab tum remote Linux server par login karte ho (secure shell), connection port 22 par establish hota hai.
21	FTP	Jab tum ek website ke files upload karte ho via FileZilla, wo FTP server ke port 21 par connect hota hai.

🔐 Proxy Server vs Firewall – Clear Difference

🌐 Proxy Server

- **Definition:** Proxy ek **intermediate server** hota hai jo client aur destination server ke beech kaam karta hai.
- **Roles:**
  - Request forward karna (client → proxy → internet).
  - Caching (frequently used data fast serve karna).
  - Security (hide client IP, filter traffic).
- **Example:** Office network mein proxy server use hota hai taaki employees sirf approved websites access kar saken.
  👉 *Interview line:* "Proxy server ek middleman hai jo client requests ko forward karta hai, caching aur security provide karta hai."

🎯 Interview-Ready Line

👉 *"Proxy server ek middleman hai jo client requests ko forward karta hai aur authorized sites filter karta hai, jabki firewall ek security guard hai jo packets inspect karke unauthorized traffic block karta hai. Proxy application-level par kaam karta hai, firewall network-level par."*

📊 Comparison Table

Feature	Proxy Server	Firewall
Layer	Application Layer (Layer 7)	Network Layer (Layer 3/4)
Focus	Content filtering, caching, privacy	Packet filtering, access control
Decision	Based on URL/domain	Based on IP, port, protocol
Use Case	Restrict websites, hide identity	Block unauthorized traffic, secure LAN
Analogy	Middleman who forwards requests	Security guard at the gate

🛡️ Firewall – Example

Scenario: Tumhare office network mein ek firewall laga hai.

- **Ek hacker bahar se tumhare LAN ke ek computer ko Telnet (port 23) se access karna chahta hai.**
- **Firewall ke rules kehte hain: "Port 23 block karo, sirf Port 80/443 allow karo."**
- **Result: Hacker ka packet firewall par hi block ho jaata hai, wo andar ghus hi nahi paata.**

👉 Firewall ka focus hai network ke andar unauthorized entry rokna. Ye IP, port, aur protocol ke basis par decide karta hai.

🌐 Proxy Server – Example

Scenario: Tumhare office mein employees internet use karte hain, aur ek proxy server configured hai.

- **Ek employee browser mein facebook.com open karta hai.**
- **Proxy server ke rules kehte hain: "Sirf Gmail aur company portal allow karo, Facebook block karo."**
- **Result: Employee ko "Access Denied" message milta hai, kyunki proxy ne request forward hi nahi ki.**

👉 Proxy ka focus hai application/content level filtering. Ye URL/domain ke basis par decide karta hai ki kaunsi website access hogi.

🔎 Telnet (Telecommunication Network)

- **Definition: Telnet ek network protocol hai jo remote computer/server par login karne ke liye use hota tha.**
- **Layer: Application Layer protocol (OSI Layer 7).**
- **Working:**
  - **Tum apne computer se ek remote machine ke command line par connect kar sakte ho.**
  - **Port number: 23 (default).**
- **Nature: Plain text communication → no encryption (isliye insecure).**

Done 🥀💅💅💅