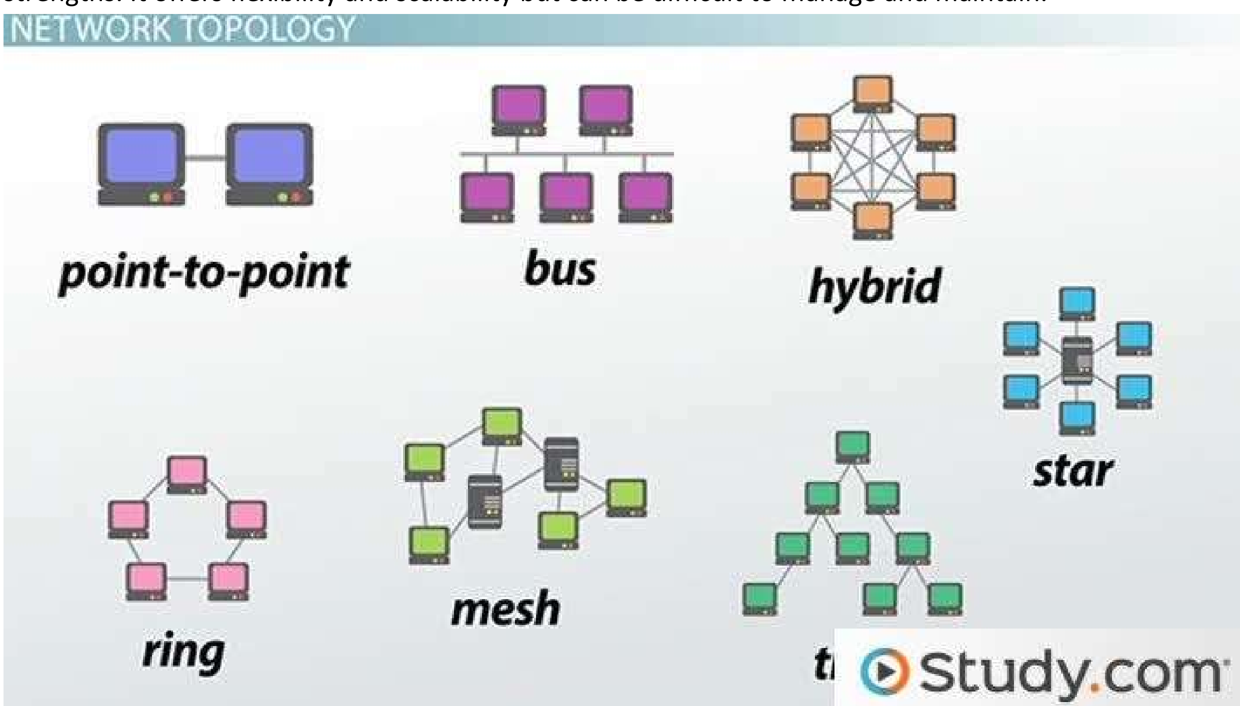
1. Different Types of Network Topologies

Network topologies define the layout or structure of devices and cables within a network. Common topologies include:

- **Bus Topology**: All devices are connected to a single central cable (the bus). Data is broadcast to all devices, and only the recipient processes it. It's easy to set up and cost-effective but becomes inefficient with many devices. A failure in the central cable disrupts the entire network.
- **Star Topology**: Devices are connected to a central hub or switch. The hub manages communication between devices. It’s easy to expand and isolates device failures, as they don't affect the rest of the network. However, if the hub fails, the entire network goes down.
- **Ring Topology**: Devices are connected in a closed loop, where each device has two neighbors. Data travels in one direction, passing through each device until it reaches the destination. While it’s inexpensive and simple, a single device failure can bring down the network unless redundancy is built in.
- **Mesh Topology**: Each device is directly connected to every other device in the network, providing multiple paths for data. This ensures high reliability and redundancy, but it’s costly and complex to install due to the many connections required.
- **Tree Topology**: Combines elements of bus and star topologies. Groups of devices are connected in star fashion, and these groups are linked together using a backbone (a bus). It’s scalable and hierarchical but can be impacted if the backbone fails.
- **Hybrid Topology**: This combines two or more topologies to take advantage of their respective strengths. It offers flexibility and scalability but can be difficult to manage and maintain.



2. OSI Model

The OSI (Open Systems Interconnection) Model is a framework that standardizes the functions of a network into seven layers, from physical transmission to application interaction:

1. **Physical Layer**: Deals with the transmission of raw bits over a physical medium (cables, switches). It includes the electrical signals and the hardware involved.
2. **Data Link Layer**: Responsible for node-to-node data transfer, error detection, and correction. It handles the framing of data and access to the physical medium.
3. **Network Layer**: Manages routing, forwarding, and logical addressing (IP addresses). It ensures data reaches its destination across different networks.
4. **Transport Layer**: Provides end-to-end communication, reliability, flow control, and error recovery (TCP, UDP). It ensures the complete and correct delivery of data.
5. **Session Layer**: Manages sessions between communicating devices. It ensures the establishment, maintenance, and termination of connections.
6. **Presentation Layer**: Translates data into a format that the application layer can understand. It handles data encryption, compression, and translation between different formats.
7. **Application Layer**: Provides network services to end users and applications, such as file transfer (FTP), email (SMTP), and web browsing (HTTP).

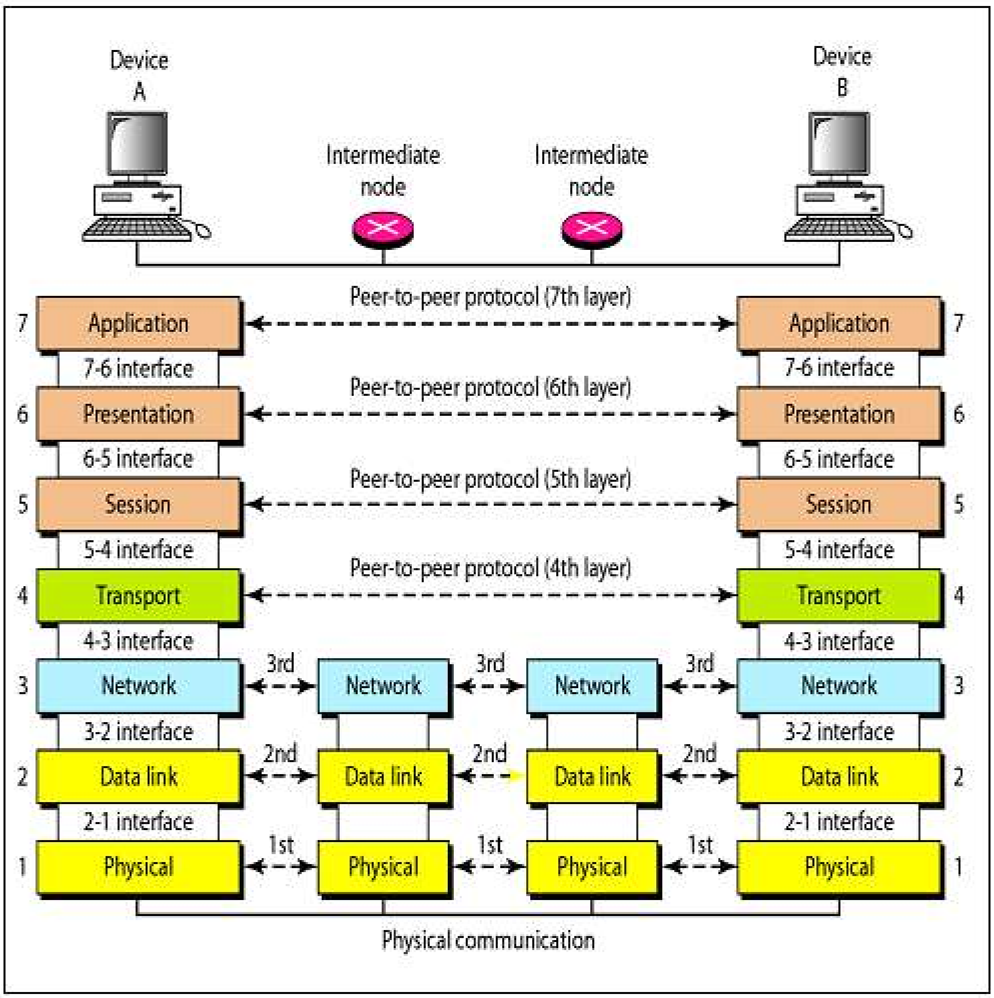


Fig: Communication & Interfaces in the OSI model

3. Multiplexing

Multiplexing refers to the technique of combining multiple signals over a shared communication medium. Three common types of multiplexing are:

- **Frequency Division Multiplexing (FDM)**: In FDM, the available bandwidth is divided into smaller frequency bands, and each signal is transmitted within its allocated band. This method is commonly used in radio and television broadcasting.
- **Time Division Multiplexing (TDM)**: TDM divides the time available on a channel into small time slots, allocating each signal a specific time slot for transmission. It's used in digital communication, such as telephony and data networks, ensuring multiple signals share the same transmission medium.
- **Wavelength Division Multiplexing (WDM)**: This is a form of multiplexing used in fiber optic communications. WDM uses different wavelengths (or light frequencies) to transmit multiple signals simultaneously over a single optical fiber. It significantly increases the capacity of fiber optic networks.

Multiplexing enables efficient use of the transmission medium by allowing multiple signals to coexist without interference. Each type of multiplexing is suited to different technologies based on the nature of the medium, such as frequency, time, or wavelength.

4. Introduction to Wired and Wireless LAN

A **Wired Local Area Network (LAN)** uses physical cables, typically Ethernet, to connect devices within a small geographical area like an office or home. It provides high-speed data transfer, secure connections, and low latency. However, it has limitations in mobility since devices must remain connected through cables.

A **Wireless Local Area Network (Wi-Fi)**, on the other hand, uses radio waves to connect devices, offering flexibility and mobility. Wi-Fi networks allow devices to connect to the internet or other network resources without physical connections. While it's convenient and allows easy expansion, Wi-Fi can suffer from interference, security risks, and lower speeds compared to wired LANs. Both wired and wireless LANs are commonly used for home, office, and campus networking.

5. TCP/IP Model

The **TCP/IP Model** is the protocol stack that underlies the internet and most networks. It has four layers:

1. **Link Layer**: Corresponds to the OSI's Data Link and Physical layers. It handles the physical transmission of data and the addressing within the network segment (Ethernet, Wi-Fi).
2. **Internet Layer**: Responsible for addressing and routing data between different networks (IP protocol). It includes the Internet Protocol (IP), which routes packets to the correct destination.
3. **Transport Layer**: Ensures reliable data transfer between devices (TCP/UDP). TCP provides error recovery and ensures data integrity, while UDP is faster but less reliable, often used for real-time applications like streaming.
4. **Application Layer**: The top layer that interacts directly with end-user applications, offering services like file transfer (FTP), email (SMTP), and web browsing (HTTP).

The TCP/IP model is widely adopted in networking because of its simplicity and scalability.

6. Error Detection & Error Correction

Error Detection involves identifying errors that occur during data transmission. Common techniques include:

- **Parity Bit**: Adds a bit to data to ensure it has an even or odd number of 1's. If the parity doesn't match, an error is detected.
- **Checksums**: A mathematical function that produces a sum for transmitted data. The receiver compares the checksum to verify data integrity.
- **Cyclic Redundancy Check (CRC)**: A more robust method, CRC uses polynomial division to detect errors. It is commonly used in Ethernet and disk drives.

Error Correction is the process of correcting errors found during transmission. Methods include:

- **Hamming Code**: Adds extra bits to the data to detect and correct errors by examining patterns.
- **Automatic Repeat Request (ARQ)**: If an error is detected, the receiver requests the sender to retransmit the data. Protocols like TCP use ARQ to ensure reliable communication.

Error detection and correction ensure data integrity and reliable communication across networks.

7. Stop & Wait ARQ and Go Back-N ARQ

- **Stop & Wait ARQ**: In this method, the sender sends one data packet at a time and waits for an acknowledgment from the receiver before sending the next packet. While simple and easy to implement, it can be inefficient, especially in high-latency networks, as the sender is idle while waiting for acknowledgments.
- **Go Back-N ARQ**: This method allows the sender to send multiple packets without waiting for an acknowledgment for each one. If any packet is lost or contains an error, the receiver requests a retransmission of that packet and all subsequent packets. This approach increases efficiency compared to Stop & Wait, but it can result in the retransmission of multiple packets, even if only one was lost.

Both protocols are used for ensuring reliable communication, with Go Back-N being more efficient in certain conditions.

8. Multiple Access Protocols

- **Pure ALOHA**: In Pure ALOHA, a device transmits data whenever it has information to send. If the transmission collides with another, both devices must wait a random amount of time before retransmitting. This method is simple but inefficient, especially with a high number of devices, as it has a high collision probability.
- **Slotted ALOHA**: A refinement of Pure ALOHA, Slotted ALOHA divides time into equal slots and only allows transmissions at the beginning of a slot. This reduces the chances of collisions, making it more efficient than Pure ALOHA, though it still faces performance issues when the network is busy.

9. CSMA/CD & CSMA/CA Protocol

- **CSMA/CD (Carrier Sense Multiple Access with Collision Detection)**: Used in Ethernet networks, CSMA/CD allows devices to listen to the channel before transmitting. If a collision is detected, devices stop transmitting, wait a random time, and retry. It's efficient in wired environments but becomes less effective as network size grows due to increased collisions.
- **CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)**: Used in wireless networks like Wi-Fi, CSMA/CA avoids collisions by having devices wait for the channel to be clear before transmitting. It also uses backoff mechanisms to reduce collision probability, but it's less efficient than CSMA/CD in terms of throughput.

10. Elements of Transport Protocol

The transport protocol is responsible for ensuring reliable data transmission between devices. Key elements include:

- **Segmentation and Reassembly**: Data is broken into smaller segments to ensure proper transmission and reassembled at the destination.
- **Flow Control**: This prevents congestion by regulating the rate of data transmission between sender and receiver.
- **Error Detection and Correction**: The transport layer ensures that data is accurately received, retransmitting if necessary.
- **Connection Establishment and Termination**: Before data transfer, a connection is established between devices, and it is terminated when

the communication is complete.

- **Multiplexing**: This allows multiple applications to use the transport layer simultaneously by distinguishing between different communication streams.

11. Internet Transport Protocols (UDP & TCP)

- **TCP (Transmission Control Protocol)**: TCP is a connection-oriented protocol, meaning it ensures reliable data transmission by acknowledging the receipt of data, sequencing packets, and requesting retransmissions in case of errors. It’s used in applications requiring reliability, like web browsing and email.
- **UDP (User Datagram Protocol)**: UDP is a connectionless protocol that doesn’t guarantee delivery, order, or error correction. It’s faster and more efficient than TCP, making it ideal for applications like live streaming and online gaming, where speed is crucial and occasional data loss is acceptable.

Here are the headings and subheadings for easy remembrance:

1. Different Types of Network Topologies

- Bus Topology
- Star Topology
- Ring Topology
- Mesh Topology
- Tree Topology
- Hybrid Topology

2. OSI Model

- Physical Layer

- Data Link Layer
- Network Layer
- Transport Layer
- Session Layer
- Presentation Layer
- Application Layer

3. Multiplexing (FDM, TDM, WDM)

- Frequency Division Multiplexing (FDM)
- Time Division Multiplexing (TDM)
- Wavelength Division Multiplexing (WDM)

4. Introduction to Wired and Wireless LAN

- Wired LAN
- Wireless LAN (Wi-Fi)

5. TCP/IP Model

- Link Layer
- Internet Layer
- Transport Layer
- Application Layer

6. Error Detection & Error Correction

- Error Detection
    - Parity Bit
    - Checksums
    - Cyclic Redundancy Check (CRC)
- Error Correction
    - Hamming Code
    - ARQ (Automatic Repeat Request)

7. Stop & Wait ARQ and Go Back-N ARQ

- Stop & Wait ARQ
- Go Back-N ARQ

8. Multiple Access Protocols

- Pure ALOHA
- Slotted ALOHA

9. CSMA/CD & CSMA/CA Protocol

- CSMA/CD
- CSMA/CA

10. Elements of Transport Protocol

11. Internet Transport Protocols (UDP & TCP)

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)