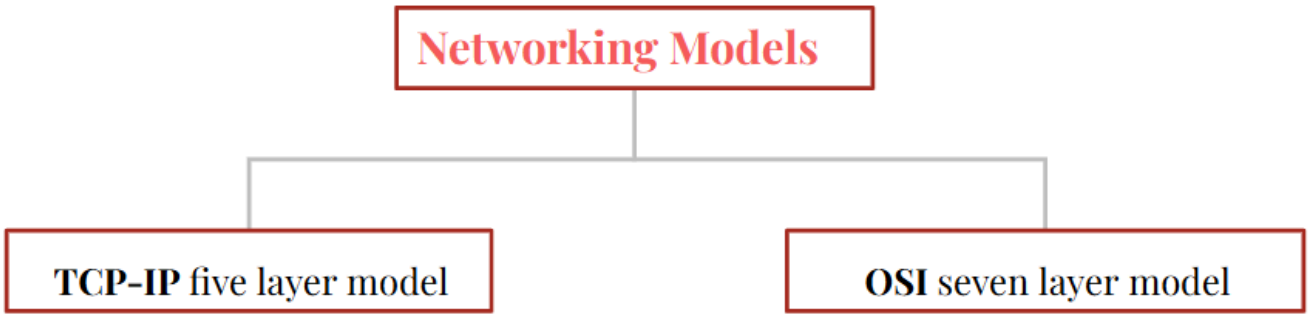
The Bits and Bytes of Computer Networking - Notes

Module 1: Introduction to Networking

Networking ensures that computers can communicate effectively by adhering to these protocols.
Protocol: defined set of standards that computers MUST follow to communicate efficiently.



- **Each layer carries data for the layers above it.**
- Networking involves multiple **protocols simultaneously**; from a single networking cable to global internet communication

The Five Layers of the TCP/IP Model

5	Application	HTTP, SMTP, etc...	Messages	n/a
4	Transport	TCP/UDP	Segment	Port #'s
3	Network	IP	Packet / Datagram	IP Address
2	Data Link	Ethernet, Wi-Fi	Frames	MAC Address
1	Physical	n/a	Bits	n/a

1 **Physical** Layer:

- Represents physical components like **cables, connectors, and networking hardware**.
- Defines how signals are transmitted over connections.

2 **Data Link** Layer:

- Also called the **Network Interface Layer** or **Network Access Layer**.
- Defines how signals are interpreted so devices can communicate.
- Common protocols: **Ethernet (wired)** and **Wi-Fi (wireless)**.
- **Ensures data is transferred between nodes on the same network.**

3 **Network** Layer:

- Also called **Internet Layer.**
- Allows different networks to communicate with each other through ROUTERS.
- Common protocols: **Internet Protocol (IP)**
- **Forms internetwork (a connection of network connected together through routers)**

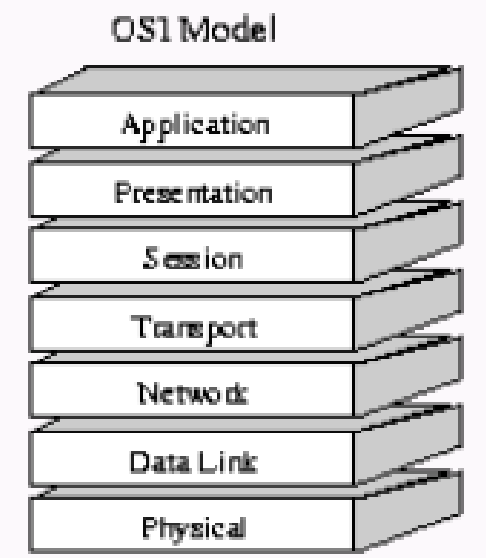
4 **Transportation** Layer:

- Ensures data is delivered to the correct applications on a device.
- Common protocols:
  - **TCP (Transmission Control Protocol) -** ensures reliable data delivery
  - **UDP (User Datagram Protocol)** - prioritises speed over reliability

5 **Application** Layer:

- Directly interacts with user applications.
- Contains application-specific protocols, such as **HTTP (web browsing)** and **SMTP (email)**.

The Seven Layers of the OSI Model



(An acronym used to help remember the model from bottom to top is "**Please Do Not Throw Sausage Pizza Away.**" From top down the "**All People Seem To Need Data Processing**" acronym can be utilized.)

Upper Layers (Application-Specific Functions)

7 **Application** Layer:

- Provides network services to end-users.
- Functions: File Transfer (FTP), Virtual Terminal (VT), Email, Directory Services.

6 **Presentation** Layer:

- Manages data formatting, encryption, and compression.
- Converts data into a common format for communication.

5 **Session** Layer:

- Establishes, maintains, and terminates network sessions.
- Manages dialogue control (one-way or two-way communication).
- Provides synchronization (checkpoints for recovery).

Lower Layers (Network-Specific Functions)

4 **Transport** Layer:

- Ensures reliable, error-free data delivery.
- Uses **acknowledgment & retransmission** for error correction.
- Manages **flow control & multiplexing** (multiple connections on one network).

3️⃣ **Network** Layer:

- Routes packets across networks (IP resides here).
- Provides **congestion control & logical addressing**.
- Used in routers.

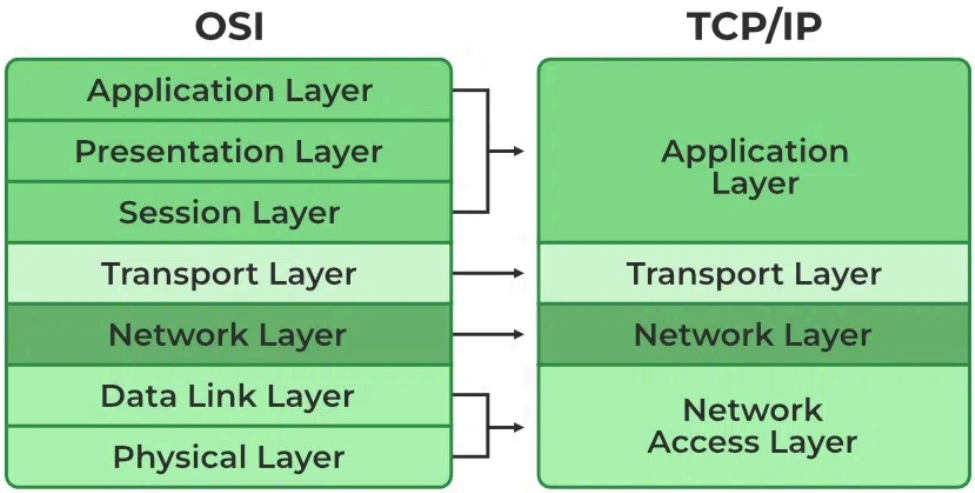
2️⃣ **Data Link** Layer:

- Converts raw transmission into frames and handles error detection.
- Manages MAC addresses and physical hardware addressing.
- Examples: Ethernet, PPP, Token Ring.

1️⃣ **Physical** Layer:

- Transmits raw bits over a communication medium.
- Defines voltage levels, network connectors, and cabling.
- Determines network topology (bus, star, ring).

Comparison: OSI Model vs. TCP/IP Model



✅ **Preferred Model: TCP/IP**

Reasons:

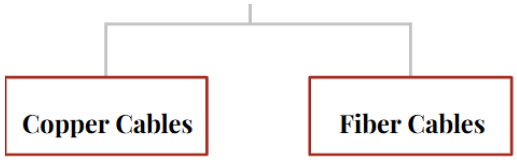
1. **Practicality**: The TCP/IP model is **designed for real-world networking** and powers the **Internet**.
2. **Simplicity**: It has only **4 layers**, making it easier to implement than the 7-layer OSI model.
3. **Widely Used**: Every modern network, including LANs, WANs, and the internet, is built on TCP/IP.

4. **Protocol Integration**: TCP/IP includes commonly used protocols like HTTP, FTP, SMTP, and DNS.
5. **Scalability**: It is **highly scalable** and works efficiently across large networks.

The basics of Networking Devices

Cables

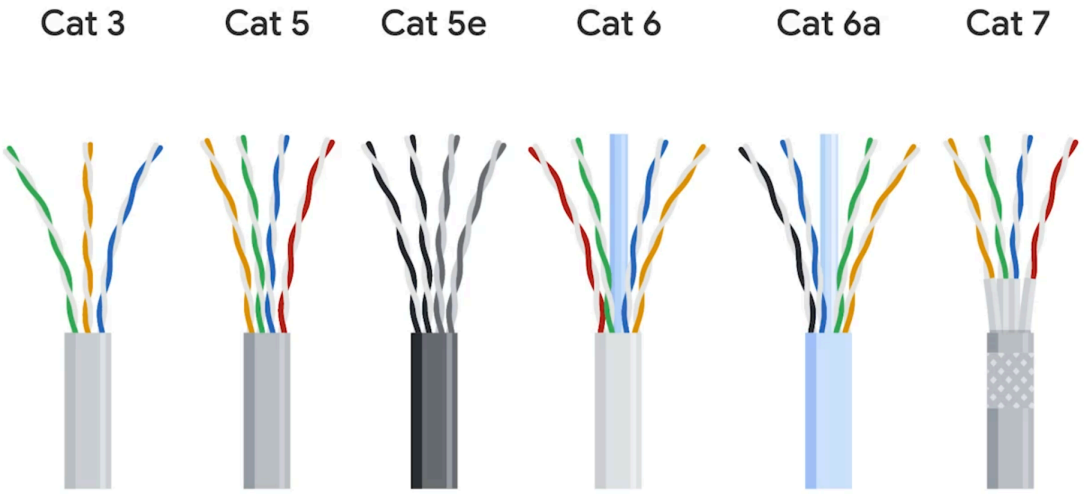
Networking cables enable data transmission between devices in a wired network.



1. Copper Cables

- Contain multiple **pairs of copper wires** inside plastic insulation.
- Transmit data using voltage changes to represent binary (1s and 0s).
- Susceptible to **crosstalk** (interference between wires), which can cause errors.

Types of Copper Cables:



- ➔ **Cat 5** – Older standard, mostly replaced.
- ➔ **Cat 5e (Enhanced)** – Reduces crosstalk, making data transfer more reliable.
- ➔ **Cat 6** – Higher data transfer rates, even stricter crosstalk reduction, but shorter maximum distance at higher speeds.

2. Fiber Cables

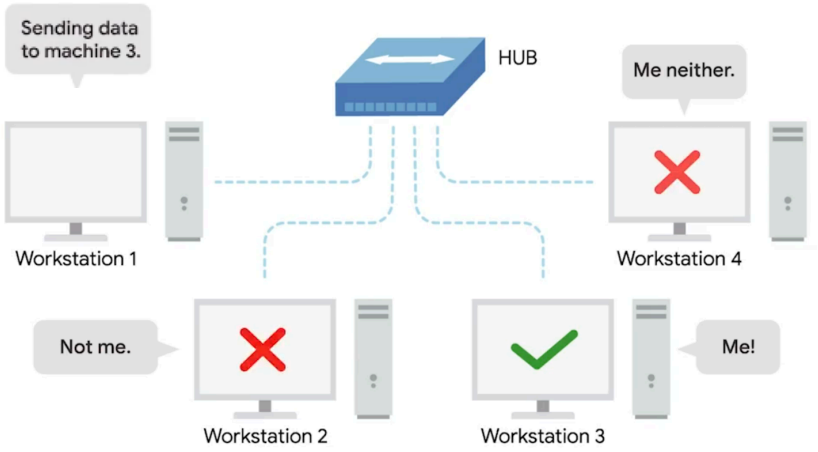
- Made of glass tubes as thin as a human hair.

- Uses **pulses of light** instead of electrical signals for data transmission.
- Advantages over Copper cables:
  - Faster data transfer speeds.
  - Longer transmission distances without data loss.
  - Resistant to **electromagnetic interference (EMI)**.
- More expensive and fragile than copper wires.

Hubs and Switches

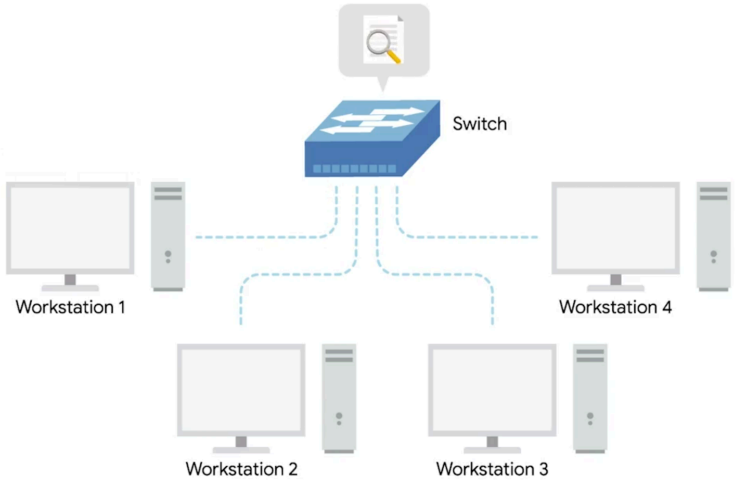
Because cable is not practical for large networks with many computers.

- A hub is a physical layer device that allows for connections from many computers at once - sends incoming data to **all connected devices**; each device decides if the data is meant for it.



Causes **collision domains** where multiple devices transmitting at the same time interfere with each other; creating **network congestion** and **slows down communication.**

- A switch is a data-link layer device used to connect multiple devices - **inspects Ethernet data** to determine the intended recipient; sends data only to the **specific device** instead of broadcasting to all.



Reduces **collision domains-** leads to fewer retransmissions and a higher overall throughput

Routers

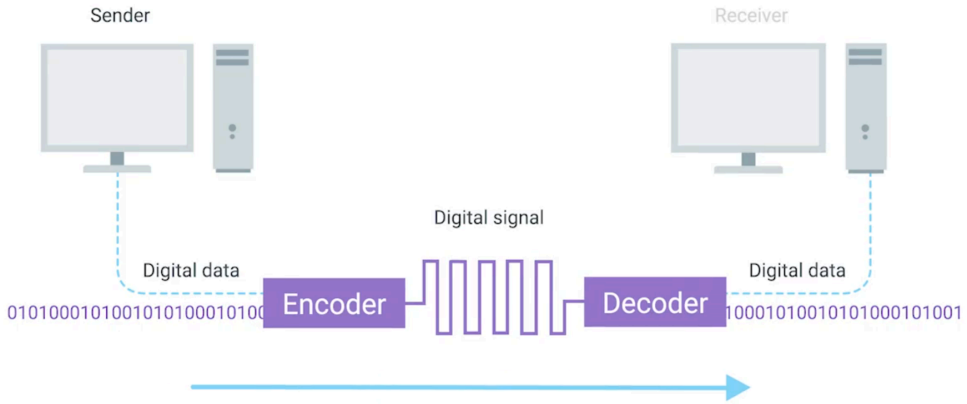
Because switches, which only forward data within a local network.

- A router is a network layer device used to connect different networks making global internet access possible - inspects **IP data** (not just Ethernet data like switches); uses **routing tables** to make decisions on where to send traffic.
- **Home routers** handle simple routing, mainly connecting LAN devices to an ISP.
- **Core ISP routers** manage global Internet traffic, using **BGP (Broader Gateway Protocol)** to share routing information and find the **best path** for traffic. Can **reroute traffic dynamically** based on congestion or failures.
- Data often travels through dozens of routers before reaching its destination.
- The Internet is incredibly large and complicated, and **routers are global guides** for getting traffic to the right places.

The Physical Layer

Moving bits across the wire

- Transmits **binary data (1s & 0s)** from one device to another.
- Modulation **(line coding)** encodes data into electrical signals.

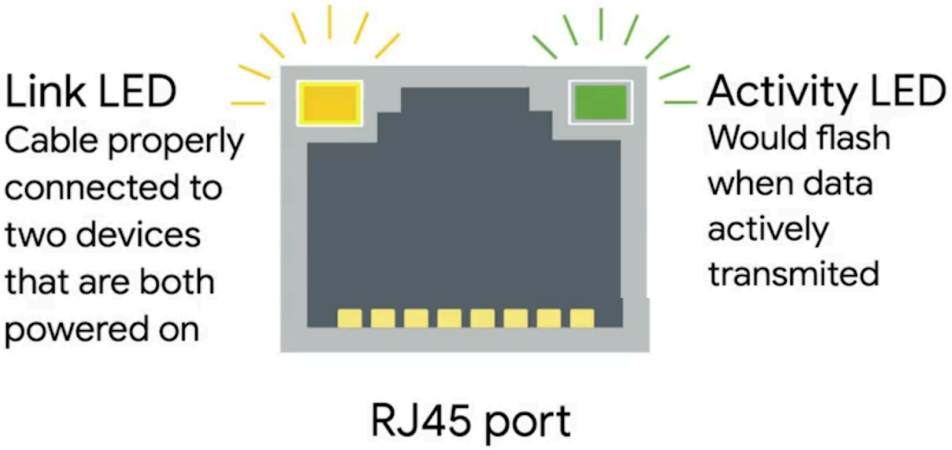


Twisted Pair Cabling and Duplexing

- **Pairs are assigned** for sending and receiving, enabling **full-duplex communication**.
  - Both devices can **send and receive data simultaneously**.
- If a **network issue** occurs, a connection may downgrade to **half-duplex mode**.
  - Devices **take turns sending and receiving data**.
- In some cases, **simplex communication** is used for one-way data flow.
  - Example: **Radio broadcasting**, where a sender transmits, and receivers only listen.

Network Ports and Patch Panels

- **The final steps of how the physical layer works take place at the endpoints of our network links.**
- **Network ports exist on devices like switches, servers, and desktops.**
- **LED indicators** on ports provide **troubleshooting information** (link status, activity, speed).



- **Patch panels** help **organize network cables** in structured cabling setups.

The Data Link Layer

Ethernet and the Need for a Data Link Layer

- **Ethernet** is the most widely used protocol for sending data across individual links
- The **data link layer** allows higher-level software to send and receive data without concern for hardware details

⬇ **But how do devices identify where to send data?**

MAC Addresses – Unique Identification in Ethernet

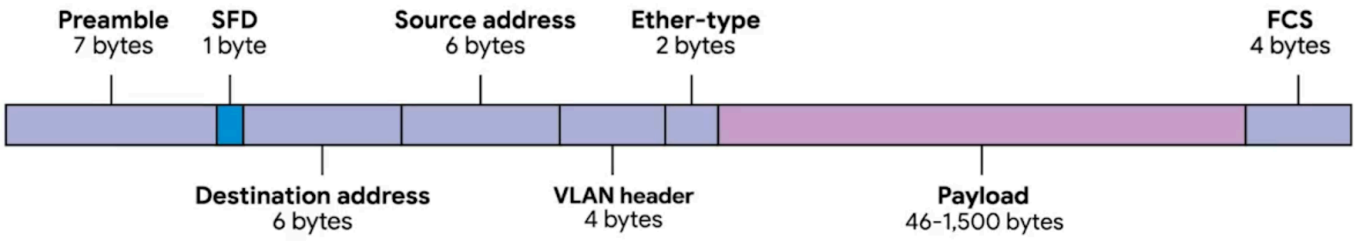
- Ethernet uses MAC addresses to determine **which device should receive a packet**.
- Every network device has a **MAC Address**, a globally unique **48-bit identifier** assigned to its network interface.
- **Structure of MAC Address:**
  - First 3 octets: **OUI (Organizationally Unique Identifier)** identifies the manufacturer.
  - **Last 3 octets**: Assigned by the manufacturer to ensure uniqueness.

CSMA/CD – Managing Collisions in Early Ethernet

- **Carrier Sense Multiple Access with Collision Detection**
- Devices listen before sending data.
- If a collision occurs, they stop, wait a random time, and retry transmission.
- While effective, this method was inefficient due to waiting times.
- Hence a **shift to Switches.**

Dissecting an Ethernet Frame

- **A structured collection of information used to transmit data across a network.**
- Helps convert raw binary data into meaningful communication between devices.
- Ensures **efficient**, **structured**, and **error-checked** communication in a network.



- **Preamble (8 bytes)**: Synchronizes sending and receiving devices.
- **Start Frame Delimiter (SFD, 1 byte)**: Signals the start of actual data.
- **Destination MAC Address**: Identifies the receiving device.
- **Source MAC Address**: Identifies the sender.
- **EtherType**: Defines the protocol used in the frame.
- **VLAN Header (Optional)**: Segregates different types of network traffic.
    - In a standard Ethernet frame (without VLAN tagging), the EtherType field directly follows the source MAC address. However, when VLAN tagging is applied, the **VLAN header** is inserted between them. The EtherType field then follows the VLAN header. This placement is necessary because the VLAN tag modifies how switches handle the frame, allowing them to segment network traffic before processing the payload.
- **Data Payload**: Contains actual data being transmitted.
- **Frame Check Sequence**: Used for error detection via **Cyclic Redundancy Check (CRC)**.
    - **Error Detection with CRC:** CRC ensures data integrity by checking if transmitted data has been corrupted. If the **checksum** doesn't match, the frame is discarded, and higher-level protocols decide if retransmission is needed.