OSI Model

Open Systems Interconnection Model:

1. The OSI Model is a conceptual framework that helps us understand how data moves across a network, step by step, layer by layer.

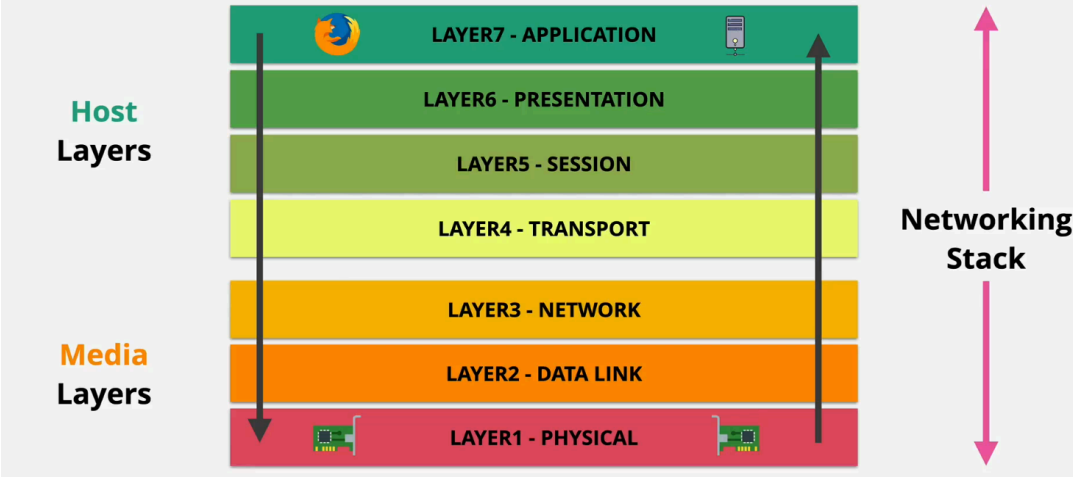
2. **Purpose:**
    - Developed by the International Organization for Standardization (ISO) and published in 1984 (ISO 7498).
    - Its goal is to standardize network communication, making it easier to design, troubleshoot, and build interoperable systems.
    - Think of it as the universal language for how different devices talk to each other over a network.

3. **Structure:**
    - Divides network functions into 7 distinct layers, stacked one above the other, each with its own specific responsibilities.
    - These layers range from physical transmission of raw data (Layer 1) all the way up to user-facing software interactions (Layer 7).
    - Communication flows down the stack on the sender's side and up the stack on the receiver's side.

4. **Why it's Useful:**
    - Simplifies troubleshooting by allowing network pros to isolate issues to specific layers.
    - Encourages modular design, hardware and software vendors can build components that focus on one layer and still work together.
    - Helps explain networking in a way that's easier to teach, learn, and document.

5. **Layers:** The OSI model is divided into seven layers, each specifying particular network functions:
    - **Physical Layer:** Deals with the physical connection between devices, including cables, switches, and other hardware. It is responsible for the transmission and reception of raw bit streams over a physical medium.
    - **Data Link Layer:** Provides node-to-node data transfer and handles error detection and correction from the physical layer. It is responsible for MAC addresses and switching.
    - **Network Layer:** Responsible for data transfer between different networks. It handles routing, forwarding, and addressing, typically using IP addresses.
    - **Transport Layer:** Ensures complete data transfer with error checking and data flow control. Protocols like TCP and UDP operate at this layer.
    - **Session Layer:** Manages sessions between applications. It establishes, manages, and terminates connections between local and remote applications.
    - **Presentation Layer:** Translates data between the application layer and the network, handling data encryption, compression, and translation.
    - **Application Layer:** Closest to the end user, it interacts with software applications that implement a communicating component, such as browsers, email clients, and file transfer programs.

A Layer X device means all layers less than X are also incorporated in it.



Layer - 1 Physical Layer

The Physical Layer is responsible for transmitting raw bits (0s and 1s) over a physical medium such as cables or radio waves. It deals entirely with hardware, not the meaning of data, just moving it across network links. This layer defines voltage levels, transmission rates, connectors, and pinouts.

Key Responsibilities:

- Converts digital data into electrical/optical/radio signals
- Defines cables, connectors, and transmission media
- Manages bit synchronization and data rates
- Supports physical topologies (bus, star, ring, etc.)

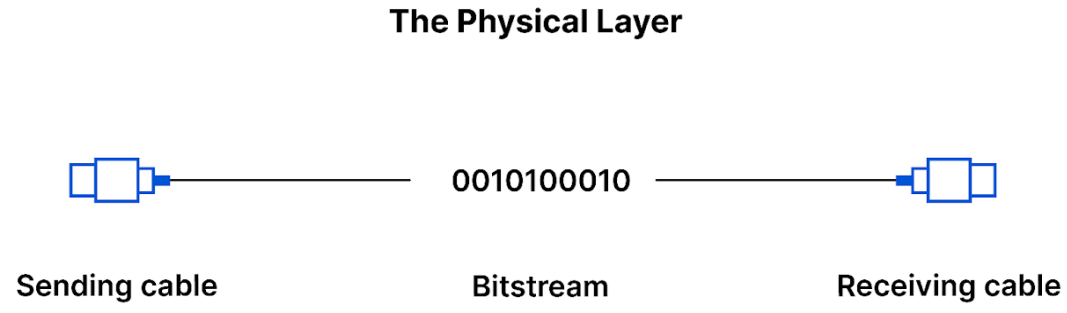
Examples:

- **Devices:** Hubs, repeaters, network cables (Cat6, fiber)
- **Protocols/Standards:** Ethernet (physical aspects), USB, RS-232, DSL, Bluetooth (RF layer)

Limitations:

- It does not understand frames, IP addresses, or any logical data
- NO error detection or correction

Analogy: It's like a fiber optic wire or copper cable, just delivering electricity or light pulses, with no clue of what's inside itself.



Layer - 2
Data Link Layer

This layer ensures reliable data transfer across a physical link by organizing bits into **frames**, managing access to the physical medium, and handling errors from Layer 1. It adds MAC addressing to identify devices on a local network segment.

Key Responsibilities:

- **Framing:** Encapsulates raw bits into frames
- **Error** detection/correction (via CRC)
- MAC addressing and media access control
- Controls how devices share the medium

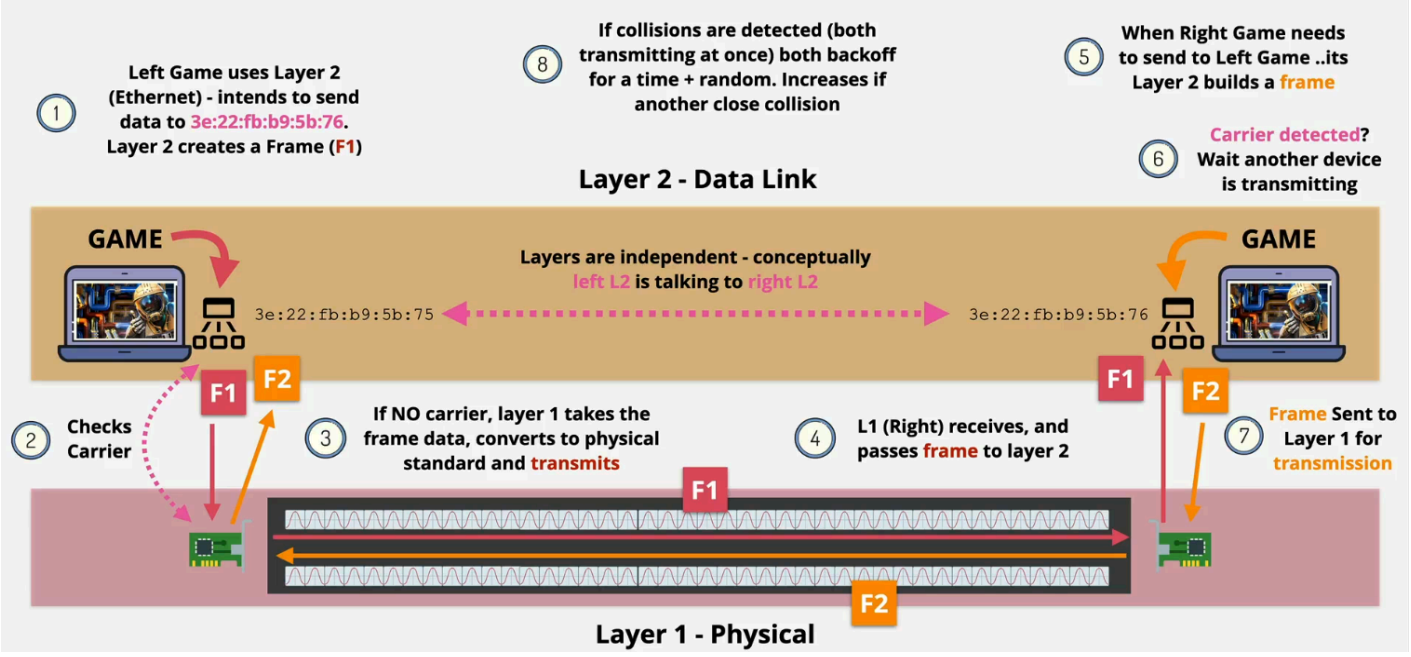
Examples:

- **Protocols:** Ethernet (IEEE 802.3), PPP, HDLC, ARP, Frame Relay
- **Sub-layers:** MAC (Media Access Control), LLC (Logical Link Control)
- **Devices:** Switches, bridges

Limitations:

- Not capable of IP routing or end-to-end communication

Analogy: It's like traffic cops directing cars (frames) through intersections, deciding who goes when and where.



Layer - 3
Network Layer

The Network Layer determines the **best path** for data to travel between devices on different networks. It handles logical addressing (e.g., IP addresses), routing, and packet forwarding.

Key Responsibilities:

- Logical addressing and routing (IP)
- Packet forwarding between networks
- Fragmentation and reassembly of packets
- Congestion control and error handling (basic)

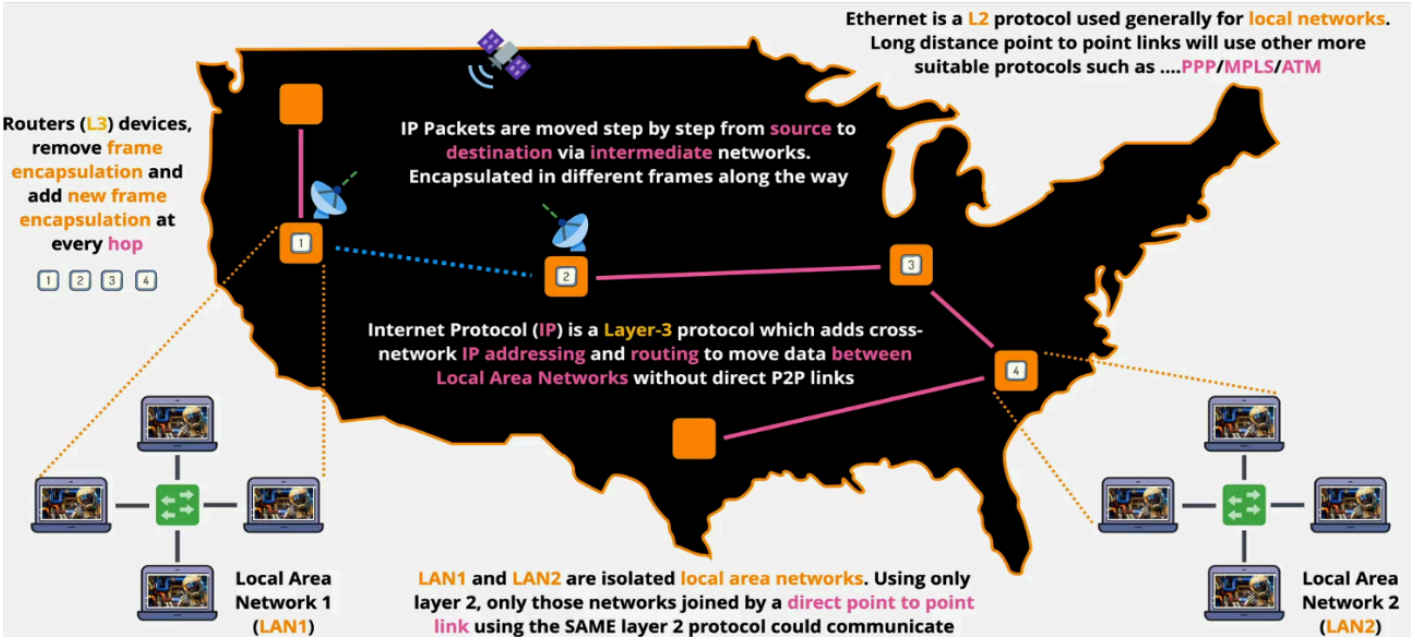
Examples:

- Protocols: IP (IPv4/IPv6), ICMP, IPX, AppleTalk
- Devices: Routers, Layer 3 switches

Limitations:

- It does not guarantee proper delivery of packets
- Does not handle the flow control or session management

Analogy: It's like a GPS navigation, it finds the most efficient route for your data to reach the destination.



Layer - 4
Transport Layer

This layer ensures **reliable data delivery** between devices, whether local or remote. It breaks large chunks into manageable segments, handles error recovery, and manages flow control.

Key Responsibilities:

- End-to-end delivery and reliability
- Segmentation and reassembly
- Flow control and congestion avoidance
- Error detection and correction

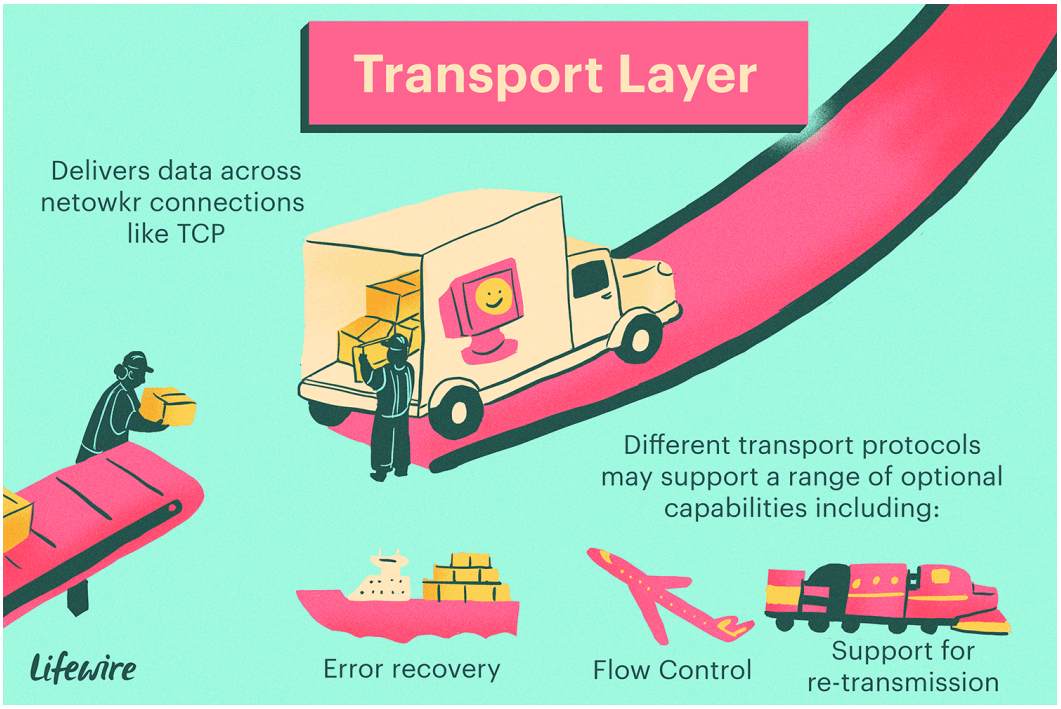
Examples:

- Protocols: TCP (reliable, connection-based), UDP (faster, connectionless), SPX

Limitations:

- Cannot handle routing or physical transmission

Analogy: Like a courier service ensuring all packages arrive complete, in order, and without damage.



Layer - 5
Session Layer

The Session Layer **establishes, manages, and terminates sessions** between two applications. It also adds structure to conversations, like setting checkpoints during large file transfers.

Key Responsibilities:

- Session initiation, maintenance, and termination
- Authentication and authorization at session level
- Synchronization (checkpoints for recovery)

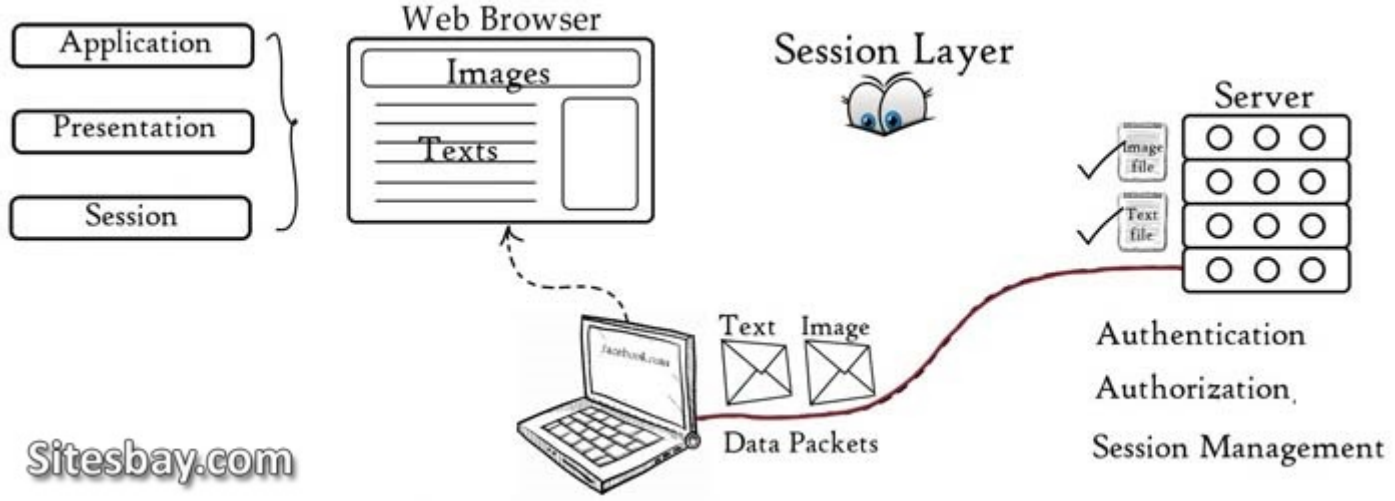
Examples:

- Protocols/APIs: NetBIOS, RPC, PPTP, SMB session services

Limitations:

- Send raw data
- Cannot handle transmission errors

Analogy: Like a moderator in a panel discussion—starts the conversation, ensures it flows, and ends it properly.



Layer - 6
Presentation Layer

This layer **translates data formats** between the application and network. It ensures that data sent from one system is readable by another. It also handles **encryption**, **compression**, and **serialization**.

Key Responsibilities:

- Data translation between formats (e.g., EBCDIC ↔ ASCII)
- Encryption and decryption
- Compression and decompression
- Character encoding conversion

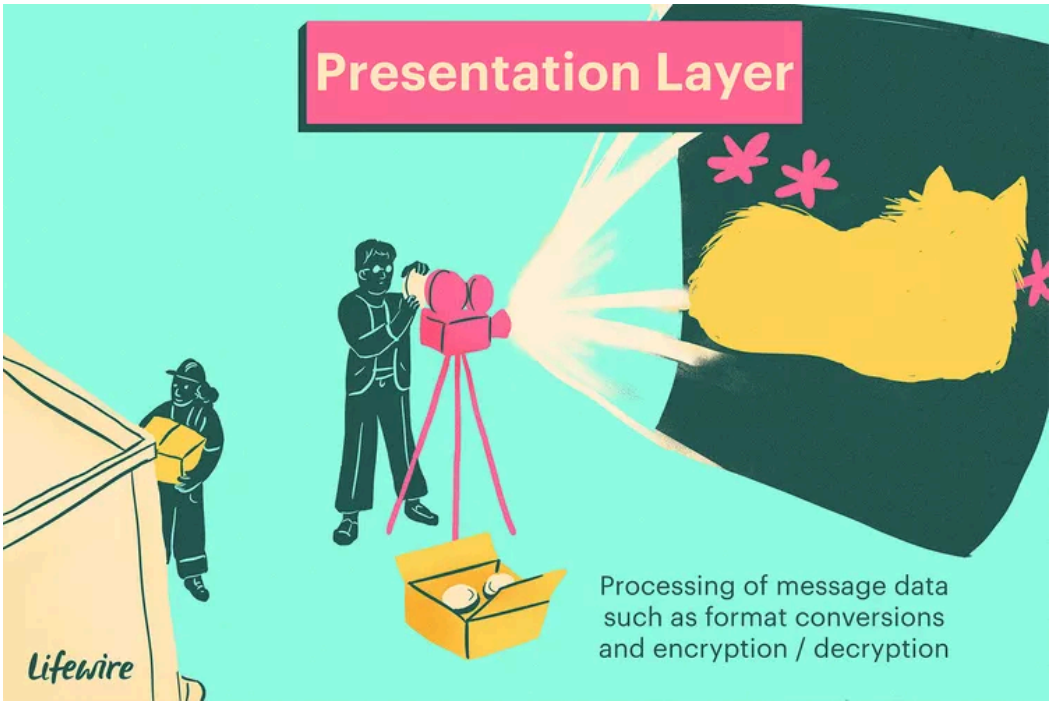
Examples:

- Formats: JPEG, MPEG, GIF, ASCII, EBCDIC
- Protocols: TLS/SSL (can span both Layer 6 & 7)

Limitations:

- Not capable of maintaining sessions or establishing connections

Analogy: Like a translator converting a movie into subtitles you can understand.



Layer - 7
Application Layer

This is the layer **closest to the end user**. It enables software applications to communicate over a network using protocols suited for their function (e.g., browsing, email, file sharing).

Key Responsibilities:

- Provides network services to user applications
- Interfaces directly with software (e.g., browser, email client)
- Identifies communication partners and resource availability

Examples:

- Protocols: HTTP, FTP, SMTP, POP3, DNS, SNMP
- Applications: Web browsers, email clients, file transfer tools

Does NOT:

- Translate or encrypt data itself (that's Layer 6's job)

Analogy: Like the app interface you interact with—it's how you "speak" to the network.