What is OSI and TCP/IP Model?

The **OSI** (Open Systems Interconnection) model and the **TCP/IP** (Transmission Control Protocol/Internet Protocol) model are both conceptual frameworks used to understand and design how different networking protocols interact in a networked communication system.

🌐 OSI Model (Open Systems Interconnection)

The **OSI model** is a **theoretical** framework developed by the **ISO (International Organization for Standardization)** to standardize networking functions into **seven layers**.

🔹 **7 Layers of the OSI Model:**

Layer	Name	Description
7	**Application**	End-user services (e.g., HTTP, FTP, SMTP)
6	**Presentation**	Data formatting, encryption, compression
5	**Session**	Establishing, managing sessions
4	**Transport**	Reliable data transfer (e.g., TCP, UDP)
3	**Network**	Path selection, IP addressing (e.g., IP)
2	**Data Link**	MAC addressing, error detection (e.g., Ethernet)
1	**Physical**	Hardware transmission (e.g., cables, signals)

🌐 TCP/IP Model

The **TCP/IP model** (also called the **Internet Protocol Suite**) is a **practical model** developed by the **U.S. Department of Defense** for designing and implementing real-world internet communication protocols.

🔹 **4 Layers of the TCP/IP Model:**

Layer	Corresponding OSI Layers	Description
Application	7, 6, 5	All end-user protocols (HTTP, DNS, FTP)

Transport	4	Reliable/unreliable communication (TCP, UDP)
Internet	3	Routing, IP addressing (IP, ICMP)
Network Access	2, 1	Hardware communication (Ethernet, Wi-Fi)

📊 Comparison Summary:

Feature	**OSI Model**	**TCP/IP Model**
Layers	7	4
Developed By	ISO	DARPA/DoD
Used For	Teaching/Reference	Internet design
Protocol Dependency	Protocol-independent	Protocol-specific (TCP/IP)
Structure	More rigid	More flexible

✅ Key Points:

- The **OSI model** is **idealized** and great for understanding and teaching network concepts.
- The **TCP/IP model** is **practical** and reflects the actual protocols used on the Internet.
- In practice, networking uses the **TCP/IP model**, but understanding both helps troubleshoot and design networks effectively.

🕰️ Networking Before the OSI Model (Pre-1979)

Before the OSI model was developed, there was **no standard framework** for how computers should communicate over a network. Each computer manufacturer—such as Dell, HP, IBM, etc.—often created its **own proprietary communication protocols**. As a result:

- **Devices from the same vendor** could typically communicate with each other because they used the same internal design and protocols.

- **Devices from different vendors** often **could not communicate**, even if physically connected, because their communication methods were incompatible.
- Networking was **limited, fragmented, and vendor-dependent**, making system integration and expansion complex and costly.

This lack of standardization **hindered the growth of computer networking**, especially as the need for interoperability between systems became more important.

🔧 Why Standardization Was Needed

To solve these problems, there was a need for a **universal model** that would allow any two systems—regardless of their manufacturer—to communicate effectively. This led to the development of the **OSI model**, which introduced a **layered architecture** and **standard communication protocols**.

🌐 OSI Model with Protocols and Functions

The data transmission process is divided into **seven layers**, each with **distinct responsibilities** on both the **sending** and **receiving** ends.

🔸 Upper Layers (Sender Side to Receiver Side)

1. **Application Layer**
    - **Function**: Provides services directly to the user.
    - **Examples**: Protocols like **HTTP, FTP, HTTPS**.
    - **Sender**: Initiates communication (e.g., web request).
    - **Receiver**: Receives the application data (e.g., webpage).
2. **Presentation Layer**
    - **Function**: Data formatting, encryption, decryption, compression.
    - **Sender**: Encrypts or compresses data.
    - **Receiver**: Decrypts or decompresses data.

3. **Session Layer**
    - **Function**: Establishes, manages, and terminates sessions.
    - **Sender**: Sets up and maintains the session.
    - **Receiver**: Uses the same session for consistent communication.

🔹 Middle Layer (Transport)

4. **Transport Layer**
    - **Function**: Reliable or unreliable data transport between host systems.
    - **Protocols**: **TCP, UDP**.
    - **Sender**: Breaks data into segments and adds port information.
    - **Receiver**: Reassembles data and ensures reliable delivery.

🔻 Lower Layers (Networking & Physical Transmission)

5. **Network Layer**
    - **Function**: Routing and logical addressing.
    - **Protocols**: **IP**.
    - **Sender**: Adds source and destination IP addresses.
    - **Receiver**: Uses IP address to direct data to the correct device.
6. **Data Link Layer**
    - **Function**: Responsible for MAC addressing and framing.
    - **Technologies**: **LLC (Logical Link Control), MAC (Media Access Control)**.
    - **Sender**: Adds MAC addresses for local delivery.
    - **Receiver**: Uses MAC to identify correct interface.

7. **Physical Layer**
    - **Function**: Transmits raw bits over a physical medium.
    - **Sender**: Converts data into signals (electrical, optical, etc.).
    - **Receiver**: Converts signals back into binary data.

🔁 Flow Summary

- Data flows **down** from the application to the physical layer on the **sender's side**.
- It flows **up** from the physical to the application layer on the **receiver's side**.
- Each layer on one side communicates with its **corresponding layer** on the other side using specific protocols.

For the send part, which layer first comes? For the receiver Part, which layer comes first?

📤 Sender Side – First Layer:

- **The first layer involved when sending data is the Application Layer (Layer 7).**
- **Data flows from top to bottom:**
  `Application → Presentation → Session → Transport → Network → Data Link → Physical`

This means the sender starts at the top (Application) and moves down the OSI stack to the Physical Layer, which sends the actual bits over the medium.

📥 Receiver Side – First Layer:

- **The first layer involved when receiving data is the Physical Layer (Layer 1).**

- **Data flows from bottom to top:**
  `Physical → Data Link → Network → Transport → Session → Presentation → Application`

This means the receiver starts at the bottom (Physical Layer receives the bits) and moves up the OSI stack to the Application Layer, where the user finally sees the data.