OSI Model and Protocols Documentation

1. Introduction

- **What is the OSI Model?**

  The Open Systems Interconnection (OSI) Model is a conceptual framework that standardizes the functions of a telecommunication or computing system into seven abstract layers.

  Its goal is to guide product developers and facilitate interoperability between different communication systems.

- **Why is it important?**
  - Provides a universal language for network design.
  - Helps in troubleshooting by isolating issues at specific layers.
  - Ensures interoperability among hardware/software from different vendors.

2. Overview of the 7 Layers

Layer No.	Layer Name	Function Summary
7	Application Layer	Interface for end-user processes and network services
6	Presentation Layer	Data representation, encryption, compression
5	Session Layer	Establishes, manages, and terminates sessions
4	Transport Layer	Reliable delivery, flow control, error recovery
3	Network Layer	Logical addressing and routing
2	Data Link Layer	Physical addressing, error detection/correction
1	Physical Layer	Transmission of raw bits over physical medium

3. Detailed Explanation of Each Layer

Layer 1: Physical Layer

- **Role:** Transmits raw bit streams over the physical medium.
- **Functions:** Defines cables, connectors, signaling, voltage levels.
- **Examples:** Ethernet cables, fiber optics, hubs, repeaters.

Layer 2: Data Link Layer

- **Role:** Node-to-node data transfer and frame error detection.
- **Sub-layers:**
  - **LLC (Logical Link Control):** Flow control & error checking.
  - **MAC (Media Access Control):** Physical addressing.
- **Examples:** Ethernet (IEEE 802.3), PPP, Switches, MAC addresses.

Layer 3: Network Layer

- **Role:** Responsible for logical addressing and routing.
- **Functions:** Path selection, packet forwarding, fragmentation.
- **Examples:** IP (IPv4, IPv6), ICMP, Routers.

Layer 4: Transport Layer

- **Role:** Reliable or unreliable delivery of data.
- **Functions:** Segmentation, flow control, error recovery.
- **Protocols:**
  - **TCP (Transmission Control Protocol):** Reliable, connection-oriented.
  - **UDP (User Datagram Protocol):** Unreliable, connectionless.

Layer 5: Session Layer

- **Role:** Manages sessions between applications.
- **Functions:** Session establishment, maintenance, termination.
- **Examples:** Remote Procedure Calls (RPC), NetBIOS.

Layer 6: Presentation Layer

- **Role:** Data translation, encryption, and compression.
- **Functions:** Converts data formats for the Application layer.
- **Examples:** JPEG, MPEG, SSL/TLS (for encryption).

Layer 7: Application Layer

- **Role:** Provides network services directly to end-users.
- **Examples:** HTTP, FTP, SMTP, DNS, POP3, Telnet.

---

4. Common Protocols Mapped to OSI Layers

Layer	Protocols/Examples
7	HTTP, FTP, SMTP, DNS, Telnet, SNMP
6	SSL/TLS, JPEG, MPEG
5	NetBIOS, PPTP, RPC
4	TCP, UDP
3	IP, ICMP, IGMP, IPsec, RIP, OSPF
2	Ethernet, PPP, Frame Relay, ATM, ARP
1	Ethernet cables, Fiber optics, Hubs, DSL

---

5. OSI vs. TCP/IP Model

OSI Layer	TCP/IP Layer

Application	Application
Presentation	Application
Session	Application
Transport	Transport
Network	Internet
Data Link	Network Access
Physical	Network Access

6. Data Encapsulation in OSI Model

- **Encapsulation:** Each layer adds its own header (and sometimes trailer) as data passes down.
    - Application → Presentation → Session → Transport (Segment) → Network (Packet) → Data Link (Frame) → Physical (Bits)
- **Decapsulation:** The reverse process happens at the receiving end.

Advantages of the OSI Model

	Description
Standardization	Provides a universal standard for different systems to communicate, ensuring interoperability.
Modular Approach	Breaks down complex networking tasks into smaller, manageable layers.
Troubleshooting	Simplifies network troubleshooting by isolating issues to specific layers.
Flexibility	Allows development and upgrades of individual layers without affecting the entire architecture.
Vendor Neutral	Helps hardware/software from different vendors work together.
Educational Tool	Makes it easier for learners to understand how networking works step by step.

ENCAPSULATION OF TCP– Byte/Bits Growth

Layer	Data	Approx Size
App/Presentation/Session	`"Hi"`	2 bytes
Transport (TCP)	TCP Header (20 bytes) + "Hi"	22 bytes
Network (IP)	IP Header (20 bytes) + TCP Segment	42 bytes
Data Link (Ethernet)	Ethernet Header (14) + IP Packet + Trailer (4)	60 bytes
Physical	Converts into 480 bits (60 bytes × 8)	480 bits

Decapsulation (Receiver Side)

The receiver will **do the reverse**:

1. Physical Layer: Convert bits to frame
2. Data Link: Remove Ethernet header/trailer
3. Network: Remove IP header
4. Transport: Remove TCP header → get actual data
5. App Layer: Show `"Hi"` on screen

TCP - Socket

Socket Lifecycle (TCP)

1. **Server Side:**
    - Create socket
    - Bind to IP & port
    - Listen for connection
    - Accept client
2. **Client Side:**

- Create socket
- Connect to server IP & port

Advantages of TCP

	Description
Reliable Transmission	Guarantees delivery of data by using acknowledgments and retransmissions.
Connection-Oriented	Establishes a connection before data transfer (3-way handshake).
Ordered Data Transfer	Data packets arrive in order, even if sent out of order.
Error Detection & Recovery	Ensures data integrity through error checking and correction.
Flow & Congestion Control	Prevents network congestion and manages data flow to match receiver capability.

ENCAPSULATION OF UDP– Byte/Bits Growth

Layer	Payload Added	UDP Total Size
App	"Hi" (2 bytes)	2 bytes
Transport (UDP)	8 bytes	10 bytes
Network (IP)	20 bytes	30 bytes
Link (Ethernet)	18 bytes	**48 bytes**
Physical	→ Convert to 384 bits	

UDP Decapsulation

Layer	What Happens?	Example
1. Physical (Layer 1)	Receives **bits** (0s and 1s) from the cable/wireless.	384 bits → electrical signals
2. Data Link (Layer 2)	Checks **Ethernet header + CRC**. Strips off the Ethernet frame, passes the **IP**	Verifies MAC address, CRC OK

	packet to Network Layer.	
3. Network (Layer 3)	Reads **IP header** (20 bytes). Checks destination IP, strips IP header, passes **UDP segment** to Transport Layer.	IP dest = my IP? Yes → next
4. Transport (Layer 4)	Reads **UDP header** (8 bytes): source port, destination port, length, checksum. Passes **payload data** to the correct application **socket**.	Dest port = 5555? Deliver to listening app
5. Session / Presentation / Application (Layers 5–7)	Data reaches the app. If encrypted, decrypts; if compressed, decompresses; shows to the user.	Chat app displays `"Hi"`

UDP Socket Lifecycle

UDP is **connectionless**, so:

- No `listen()`
- No `accept()`
- No handshake
- Uses `sendto()` and `recvfrom()`

Advantages of UDP

	Description
Low Overhead	No connection establishment or acknowledgment, so it's faster and more lightweight.
Broadcast & Multicast Support	Ideal for applications like DNS queries, VoIP, video streaming, and online gaming.
Connectionless	Suitable for simple request-response protocols where reliability can be handled by the application.
Reduced Latency	No handshaking delays — useful for time-sensitive transmissions.
Stateless	The server does not maintain session state, which reduces resource usage for each client.