

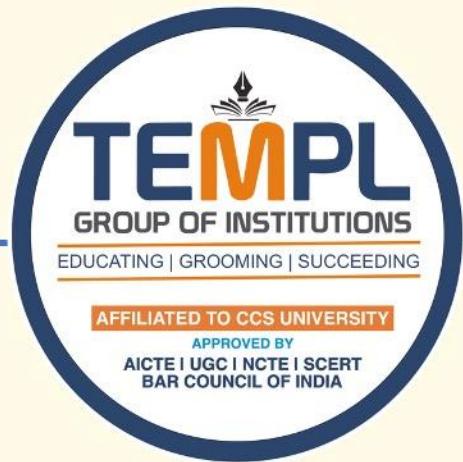
UNIT-V Transport and upper layers in OSI Model:

Transport layer functions,

connection management,

functions of session layers,

presentation layer and application layer.



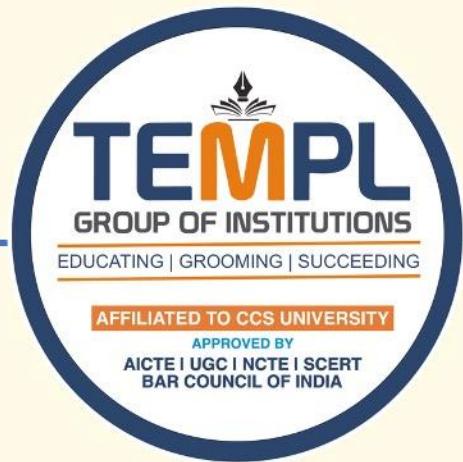
Transport Layer (Layer 4 of OSI Model)

The **Transport Layer** provides **reliable, transparent transfer of data** between two end systems (host to host).

It ensures that **data is delivered error-free, in sequence, and without losses or duplication**.

Transport Layers Functions

- Addressing
- Connection Management
- Flow Control
- Multiplexing
- Crash Recovery



1. Addressing

- The Transport Layer uses **Port Numbers** to identify **specific processes or applications** running on a host.
- This is known as **Process-to-Process Communication**.
- Example:
 - Web browser → Port **80 (HTTP)**
 - Email → Port **25 (SMTP)**

❑ **IP address** identifies the computer,

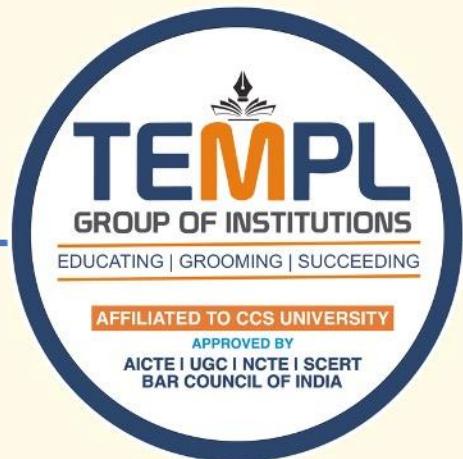
❑ **Port number** identifies the specific process within that computer.



2. Connection Management

- The Transport Layer is responsible for **establishing, maintaining, and terminating** logical connections between devices.
- Two types:
 - **Connection-oriented (TCP):**
Involves three phases – *Connection Establishment, Data Transfer, Connection Termination* (e.g., TCP handshake).
 - **Connectionless (UDP):**
No connection setup; data is sent directly.

❑ Ensures smooth start, communication, and end of a session.



1. Connection Establishment

- Also called "Setup Phase".
- Ensures both devices are **ready to send and receive data**.
- Common method: **Three-Way Handshake** (used in TCP).

Steps:

1. **SYN**: Sender requests a connection.
2. **SYN-ACK**: Receiver acknowledges and agrees.
3. **ACK**: Sender confirms — connection established.

Now data transfer can begin.



2. Connection Release

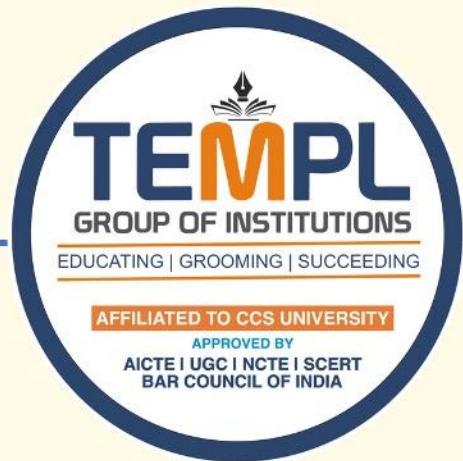
- Also called "**Teardown Phase**".
- Used to **close** the connection after transmission ends.
- Common method: **Four-Way Handshake** (in TCP).

Steps:

1. Sender sends **FIN** (finish request).
2. Receiver sends **ACK** (acknowledgment).
3. Receiver also sends **FIN** when ready to close.
4. Sender replies with **ACK** — connection closed.

Communication channel is released properly.

| Stage | Purpose | Example in TCP |
|---------------------------------|---------------------|-------------------------------------|
| Connection Establishment | Start communication | 3-way handshake (SYN, SYN-ACK, ACK) |
| Connection Release | End communication | 4-way handshake (FIN, ACK) |



3. Flow Control

- Ensures the **sender does not overwhelm the receiver** by sending too much data at once.
- Maintains a **balanced speed** between sender and receiver.
- Example:
 - **TCP uses sliding window protocol** for flow control.
- ☒ Prevents **data loss and congestion** in the network.



4. Multiplexing and Demultiplexing

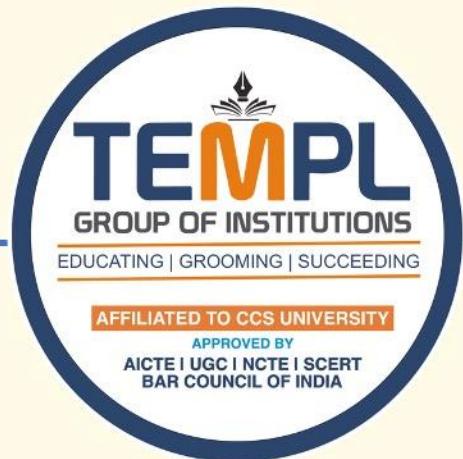
- **Multiplexing:**

Allows multiple applications to share the same network connection simultaneously.
(e.g., browsing + downloading + emailing at the same time)

- **Demultiplexing:**

At the receiver's end, it delivers received data to the correct application based on **port numbers**.

❑ This helps in **efficient use of network resources**.



5. Crash Recovery

- Ensures data is not lost if a connection **fails or system crashes** during transmission.
- **TCP** uses **acknowledgments and retransmission** mechanisms to recover from failures.
- After recovery, communication resumes from the **last acknowledged point**.
- Maintains **data integrity and reliability** even during failures.



| Function | Description |
|------------------------------|--|
| Addressing | Identifies sending and receiving processes using port numbers. |
| Connection Management | Establishes, maintains, and ends communication sessions. |
| Flow Control | Balances data rate between sender and receiver. |
| Multiplexing | Enables multiple applications to use one network connection. |
| Crash Recovery | Recovers lost data after connection failure or crash. |



Transport Layer Protocols:

There are **two main protocols** used at the Transport Layer:

1. TCP (Transmission Control Protocol)

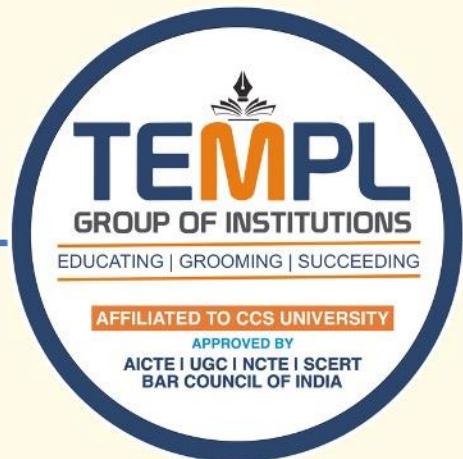
Type:

Connection-oriented protocol

Features:

1. **Reliable Communication** – Ensures error-free and ordered delivery.
2. **Connection Establishment** – Uses **3-way handshake** before data transfer.
3. **Flow Control** – Prevents data overflow using **Sliding Window protocol**.
4. **Error Control** – Detects and retransmits lost or damaged segments.
5. **Congestion Control** – Controls data flow when network is busy.
6. **Segmentation and Reassembly** – Divides data into segments, reassembles at receiver.

Example Applications: File Transfer (FTP)



2. UDP (User Datagram Protocol)

Type:

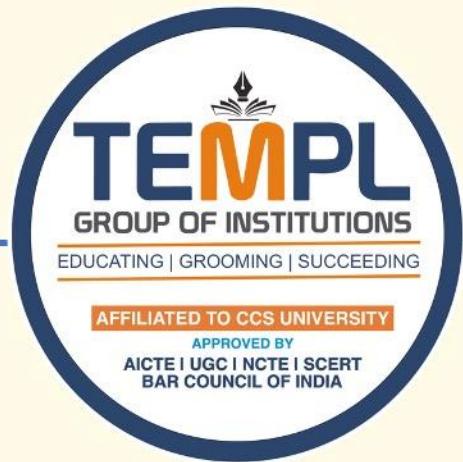
Connectionless protocol

Features:

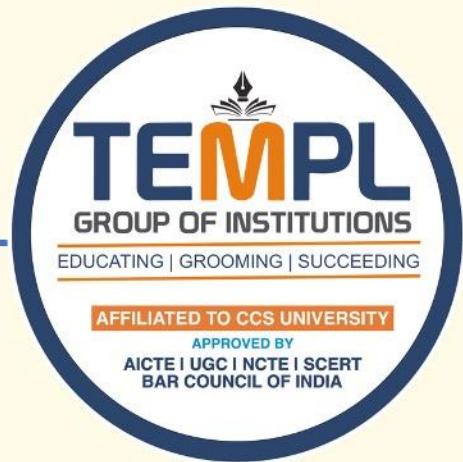
1. **No Connection Setup** – Sends data directly without handshake.
2. **Unreliable Communication** – No acknowledgment, no retransmission.
3. **Fast Transmission** – Less delay and overhead.
4. **No Flow or Error Control** – Simple, lightweight protocol.
5. **Used for Real-Time Applications** where speed is more important than reliability.

Example Applications:

- Online Gaming



| Feature | TCP | UDP |
|---------------|--------------------------------|---------------------|
| Type | Connection-oriented | Connectionless |
| Reliability | Reliable (ACK, retransmission) | Unreliable |
| Speed | Slower | Faster |
| Flow Control | Yes | No |
| Error Control | Yes | No |
| Use Case | Email, Web, FTP | Video, Audio, Games |
| Overhead | High | Low |



Session Layer (Layer 5 of OSI Model)

Definition:

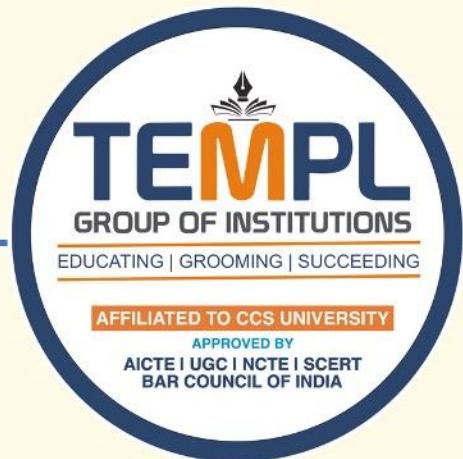
The **Session Layer** is the **fifth layer** of the OSI model.

It is responsible for **establishing, managing, and terminating sessions (connections)** between two communicating devices or applications.

It acts as a **dialog controller**, keeping track of whose turn it is to send or receive data.



| Function | Description |
|---------------------------------|--|
| Session Establishment & Release | Starts and ends the session between two devices. |
| Data Exchange | Manages orderly flow of data during the session. |
| Interaction Management | Controls dialog (who sends/receives and when). |
| Session Recovery | Restores connection after failure using checkpoints. |
| Exception Reporting | Handles and reports errors during session. |

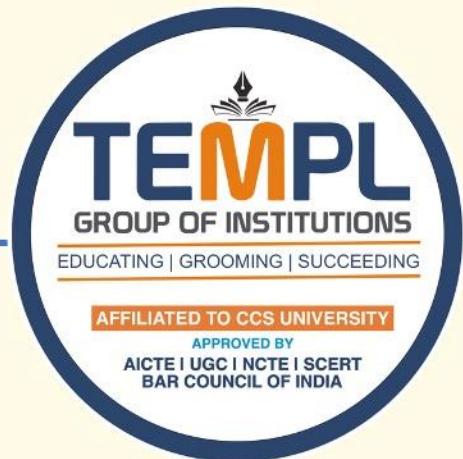


Presentation Layer (Layer 6 of OSI Model)

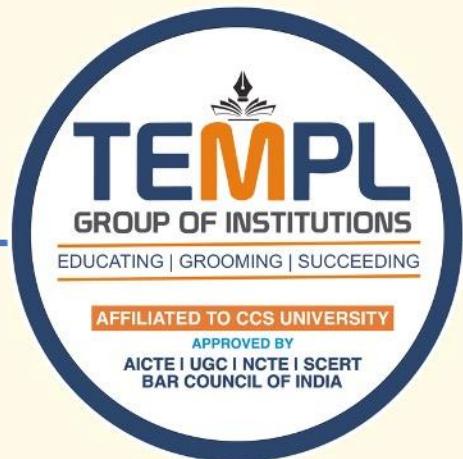
The Presentation Layer is the **sixth layer** of the OSI Model.

It acts as a **translator** between the **Application Layer** and the **Network**.

Its main job is to **format, translate, encrypt, and compress data** so that the data sent by one system's application layer can be understood by the other system's application layer.



| Function | Description |
|-------------------------|--|
| Translation | Converts data from application to network format and vice versa. |
| Encryption / Decryption | Secures data during transmission. |
| Compression | Reduces data size for faster transmission. |
| Formatting | Defines structure and representation of data. |
| Code Conversion | Converts different character encoding systems. |



1. Data Translation (Format Conversion)

- Converts data from the **application layer format** into a **common format** suitable for transmission.
- Ensures that data sent from one system can be understood by another, even if they use different data formats.

Example:

Converting between ASCII (used by PCs) and EBCDIC (used by mainframes).



Data Encryption and Decryption (Security)

- **Encryption:** Converts plain text data into **coded form** before sending, to protect it from unauthorized access.
- **Decryption:** Converts coded data back into **readable form** at the receiver end.

Example:

Secure web communication using **SSL (Secure Sockets Layer)** or **TLS**.



Symmetric (Secret-Key) Cryptography

Symmetric Cryptography, also called **Secret-Key Cryptography**, is a method of encryption and decryption where **the same key** is used for both operations.

That means **both sender and receiver share one common secret key**.

- **Sender** encrypts the plain text using a **secret key** → produces **cipher text**.
- **Receiver** decrypts the cipher text using **the same secret key** to get back the **original message**.

Plain Text → [Encryption + Secret Key] → Cipher Text
Cipher Text → [Decryption + Same Key] → Plain Text

Example:

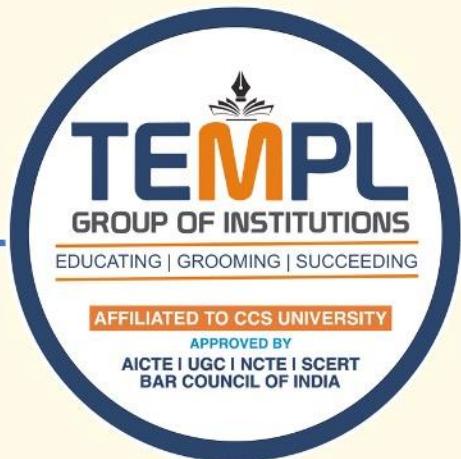
- Suppose the key = 7
- Message: “HELLO” → encrypted using key 7 → “OLSSV”
- Receiver uses the same key (7) to decrypt back to “HELLO”.

 **Asymmetric Cryptography (Public Key Cryptography)****Definition:**

Asymmetric cryptography is a method of encryption that uses **two different keys** —

- 👉 **Public Key** (shared with everyone)
- 👉 **Private Key** (kept secret by the owner)

It is also called **Public Key Cryptography**.



| Type of Key | Purpose | Who Has It |
|--------------------|--|---------------------------|
| Public Key | Used for encryption or verifying a signature | Shared openly with anyone |
| Private Key | Used for decryption or creating a signature | Kept secret by the owner |

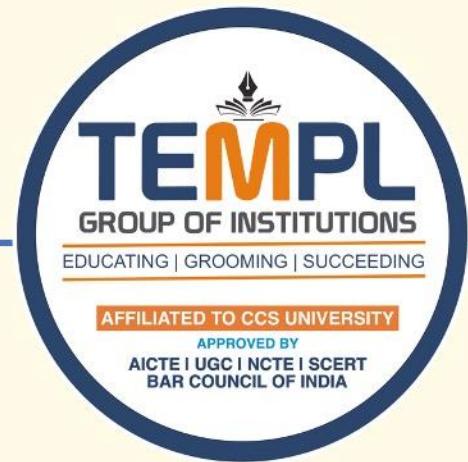


Application Layer (Layer 7 of OSI Model)

The **Application Layer** is the **topmost layer** (Layer 7) of the **OSI Model**.

It provides **services directly to the user or application software** to access network resources.

In short — it's the layer where **users interact with the network** through applications like browsers, email, or file transfer tools.



| Function | Description |
|--|---|
| 1. Network Virtual Terminal | Allows a user to log on to a remote host as if it were local (used in Telnet). |
| 2. File Transfer, Access, and Management (FTAM) | Enables users to access, read, write, or manage files on a remote computer. |
| 3. Mail Services | Provides email forwarding, storage, and access (used in SMTP, POP3, IMAP). |
| 4. Directory Services | Provides access to global information about network resources (like DNS or LDAP). |
| 5. Resource Sharing | Helps share printers, files, and other network services. |