



# Lecture 2 – TCP/IP, UDP, TCP 3 Way Handshakes & OSI Model

---

## 1. Network Protocols

A **network protocol** is a set of rules that define how data is communicated between devices over a network.

The two main transport protocols are **TCP/IP** and **UDP**.

---

### A. TCP/IP (Transmission Control Protocol / Internet Protocol)

#### What is TCP/IP?

- A **connection-oriented** protocol.
- Connection is established *before* sending data.
- Breaks data into **segments**.
- Performs **error checking** and **retransmission** if any data is lost.

#### How It Works?

1. Creates a reliable connection between sender and receiver.
2. Sends data in small segments.
3. Checks whether data reached correctly.
4. If any segment is lost, it is resent.

#### Why Is It Used?

- Because it is **reliable**, **accurate**, and **secure**.
- Ensures that data arrives correctly and in order.

#### Daily Life Example

- Uploading a video on YouTube.  
(You can't afford missing video data, so reliability is important.)
-

## B. UDP (User Datagram Protocol)

### What is UDP?

- A **connectionless** protocol.
- No handshake, no guarantee.
- Sends data without checking errors.

### How It Works?

- Sends packets called **datagrams** without confirming delivery.
- No retransmission even if data is lost.

### Why Is It Used?

- Fast, low-latency communication is required.
- Some data loss is acceptable.

### Daily Life Example

- Live streaming a cricket match.
- Online gaming.
- Video calls.

(If a few packets drop, your stream may blur for a second, but it continues without lag.)

---

## 2. TCP 3-Way Handshake

Before sending data, TCP creates a **secure and reliable connection** using a 3-step process.

### Steps in 3-Way Handshake

#### Step 1 – SYN (Synchronize)

- Client sends a **SYN packet** to the server.
- Includes an initial sequence number, e.g.,  **$x = 10$** .

#### Step 2 – SYN + ACK (Synchronize + Acknowledge)

- Server receives SYN  $\rightarrow$  increases the value by 1  $\rightarrow x + 1$ .
- Server also sends its own sequence number, e.g.,  **$y = 20$** .
- Sends back: **SYN + ACK**.

#### Step 3 – ACK (Acknowledge)

- Client receives SYN+ACK  $\rightarrow$  increases  $y$  by 1  $\rightarrow y + 1$ .
- Sends ACK to server.

**Now the connection is established.**

### Why Used?

- To create a reliable, synchronized connection between client and server.
- Ensures both sides are ready to send/receive data.

### Daily Life Example

- When you open Instagram or Google  $\rightarrow$  your device first establishes a TCP connection using this handshake.
-

### 3. OSI Model (7 Layers)

OSI = **Open Systems Interconnection** model.

It explains **how data travels from sender to receiver** in 7 steps.

---



**Sender Side (Data going from User → Network → Cable)**

---

#### 1. Application Layer

**What?**

- The layer where humans interact with software.
- Browsers, email apps, WhatsApp, etc.

**How it Works?**

- User types a request like google.com.
- Application layer prepares the request but **cannot understand network language**.

**Example:**

- Opening a website in Chrome.
- 

#### 2. Presentation Layer

**What?**

- Converts data into a standard format.
- Responsible for **encryption, compression, and translation**.

**How it Works?**

- Sender side: **Encrypts data**
- Receiver side: **Decrypts data**

**Example:**

- HTTPS encryption while logging into Gmail.
-

### 3. Session Layer

#### What?

- Maintains the **communication session** between devices.

#### How it Works?

- Ensures that the session does not break while data transfer is happening.

#### Example:

- Your Zoom call staying connected for hours.
- 

**These top 3 layers = Software Layers**

---

### 4. Transport Layer

#### What?

- Breaks big data into small pieces (segments).
- Adds **sequence numbers**.

#### How it Works?

- If you send a 10MB file, it is divided into many segments.
- Each segment gets a unique sequence number to reassemble later.

#### Example:

- Sending a file on WhatsApp.
- 

### 5. Network Layer

#### What?

- Works with **IP addresses**.
- Determines the best path for data.

#### How it Works?

- Router operates here.

- Adds sender and receiver **IP addresses**.

**Example:**

- Your phone sends data to YouTube servers using IP routing.
- 

## **6. Data Link Layer**

**What?**

- Deals with **MAC addresses** and link-level communication.

**How it Works?**

- Converts IP packets → Frames.
- Adds MAC address for local delivery.

**Example:**

- Switch forwarding your laptop's data using MAC table.
- 

## **7. Physical Layer**

**What?**

- Final layer where data is converted into **binary (0s and 1s)**.
- Travels through cable, Wi-Fi signals, fiber optics, etc.

**Example:**

- Electrical signals in LAN cable.
- 
-



## Receiver Side (Data coming from Cable → User)

The same layers work **in reverse** on the receiver side:

---

### 1. Physical Layer

- Converts electrical/wireless signals back to binary.

### 2. Data Link Layer

- Checks the **MAC address**.
- Ensures data is delivered to the correct device.

### 3. Network Layer

- Finds receiver's **IP address**.

### 4. Transport Layer

- Reassembles segments using **sequence numbers**.

### 5. Session Layer

- Maintains the communication session.

### 6. Presentation Layer

- Decrypts and translates data.

### 7. Application Layer

- Displays the final output to the user (webpage, video, message, etc.)