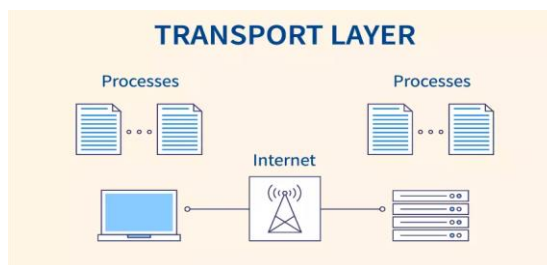The Transport Layer is responsible for process-to-process communication. It ensures complete data transfer and reliability between host systems.

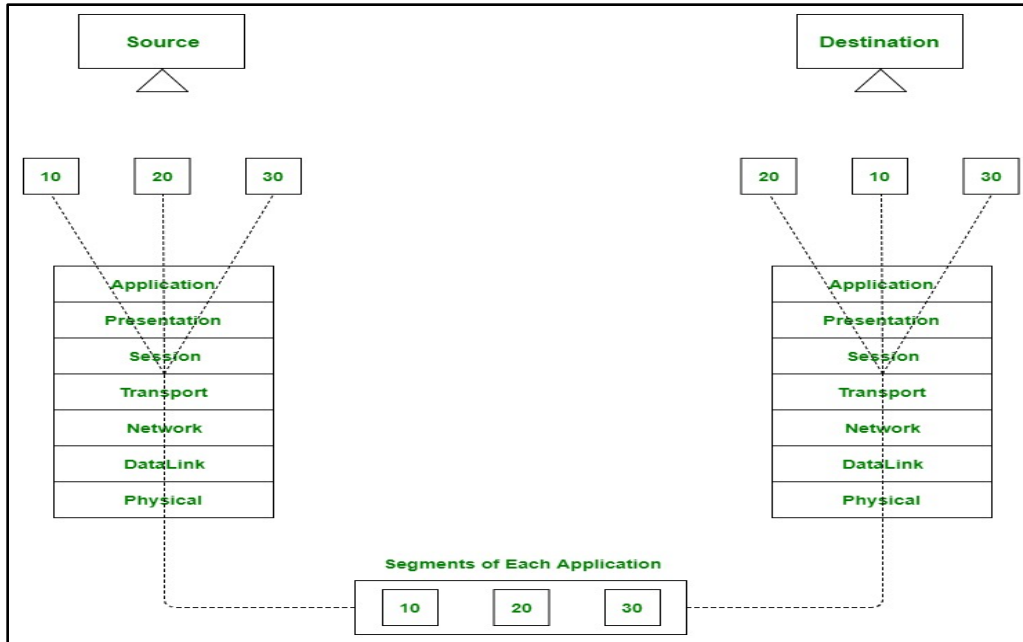**TCP vs UDP:**

On the Basis of	TCP	UDP
Connection Type	Connection-oriented	Connectionless
Reliability	Reliable	Not reliable
Acknowledgment	Required	Not used
Flow Control	Uses sliding window	Not available
Congestion Control	Yes	No

**Port Numbers, Multiplexing/Demultiplexing:**

1. **Port Numbers:**
   1) It is a logical address used to identify specific processes or services on a device for communication.
   2) Valid port numbers range from **0 to 65535**, categorized as **Well-known (0–1023)**, **Registered (1024–49151)**, and **Dynamic/Private (49152–65535).** E.g. SMTP-25, HTTP - 80, HTTPS - 443.

GLA UNIVERSITY
Accredited with A+ Grade by NAAC
Mathura | Greater Noida

Summer Immersion
Placement Program
SIPP 2025

2. **Multiplexing:** It allows multiple applications to share the network at the sender side.

3. **Demultiplexing:** It delivers incoming data to the correct application at the receiver side

TCP Three-way Handshake:
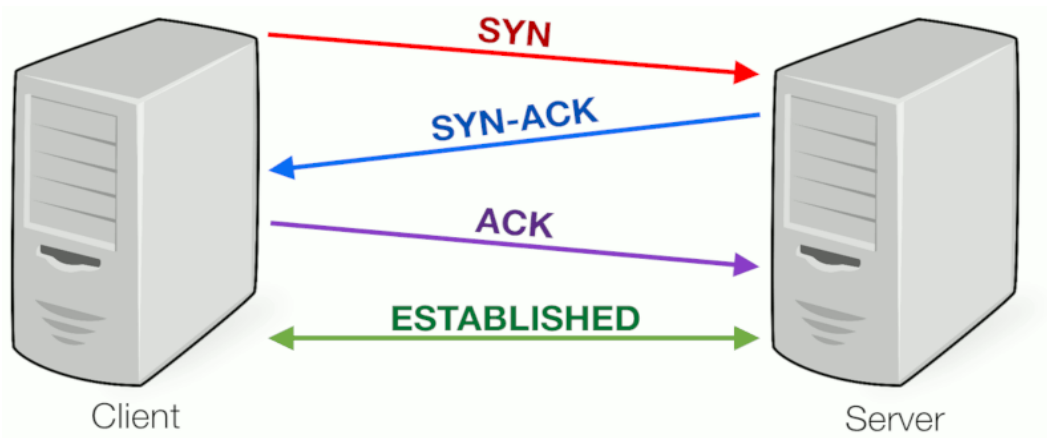
- ❖ **SYN:** Client sends a connection request.
- ❖ **SYN-ACK:** Server acknowledges and sends back a connection request.
- ❖ **ACK:** Client acknowledges and the connection is established.

Transport Layer Protocols:

- ❖ **TCP:** Reliable, connection-oriented, used for file transfer, email, web browsing.

- ❖ **UDP:** Unreliable, connectionless, used in streaming, DNS, VoIP.

- ❖ **SCTP (Stream Control Transmission Protocol):** Combines features of both TCP and UDP. Supports multi-streaming and multi-homing.

TCP Segment Structure:
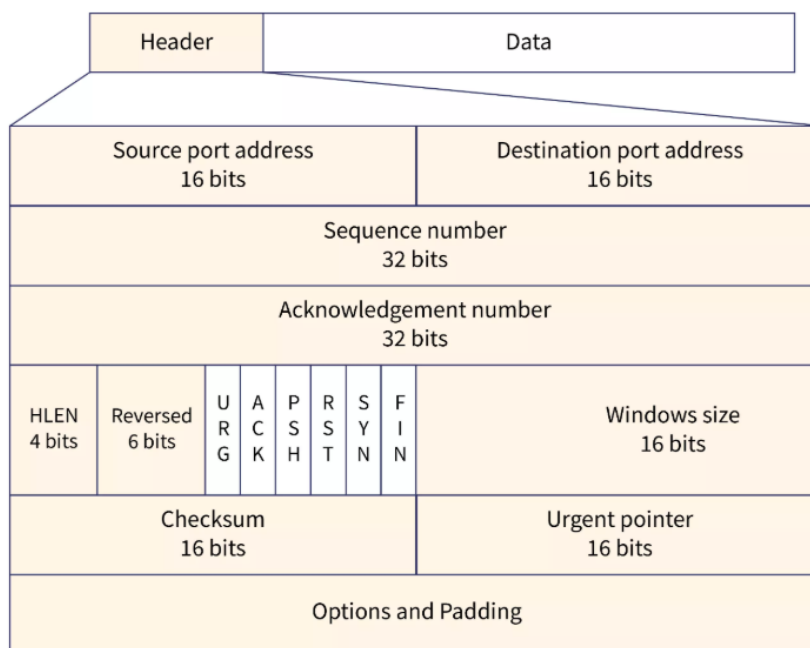
- ❖ **Source Port and Destination Port:** Identify sending and receiving applications.

- ❖ **Sequence Number:** Keeps track of data segments sent.

- ❖ **Acknowledgment Number**: Confirms receipt of data.

- ❖ **Header Length:** Indicates the start of data in the segment.

- ❖ **Flags:** Control bits such as SYN, ACK, FIN, RST, PSH, URG.

- ❖ **Window Size:** Indicates how much data the receiver can accept.

- ❖ **Checksum:** Used for error-checking.

- ❖ **Urgent Pointer:** Used if the URG flag is set.

UDP Segment Structure:
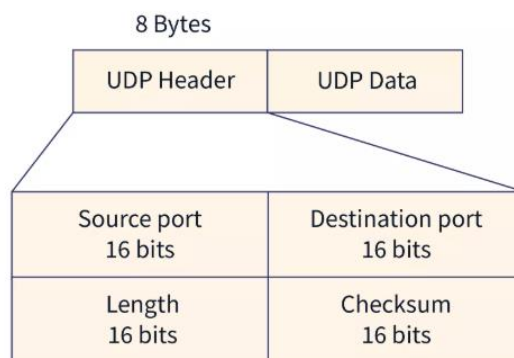
❖ **Source Port and Destination Port:** Identify sending and receiving applications.

❖ **Length:** Total length of UDP segment.

❖ **Checksum:** Used for error detection, optional in IPv4.

Note : However, in **IPv6**, the UDP checksum is **mandatory** to ensure data integrity.
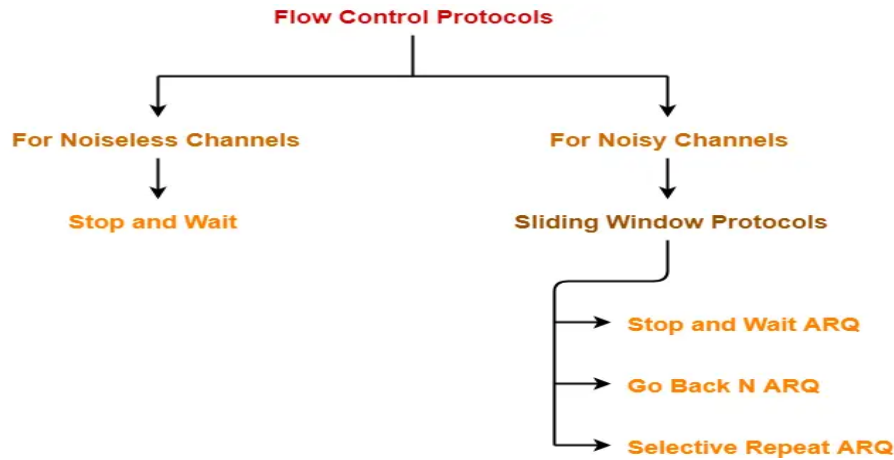
TCP Connection Termination (Four-way & Simultaneous):
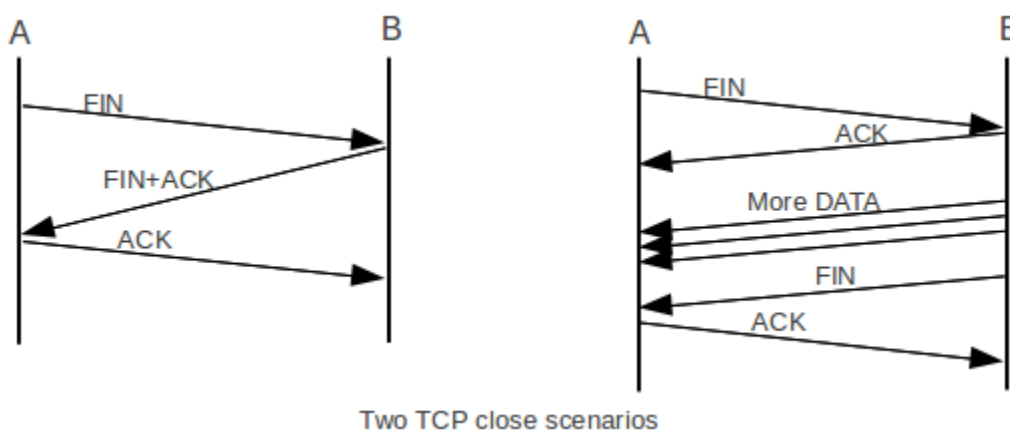
Two TCP close scenarios

❖ **FIN:** Sender initiates connection termination.

❖ **ACK:** Receiver acknowledges FIN.

❖ **FIN:** Receiver sends FIN to terminate connection from its side.

❖ **ACK:** Sender acknowledges FIN and connection is closed.

TCP Flow Control Mechanism :

❖ It is a mechanism that prevents the sender from **overwhelming** the receiver with too much data at once. It ensures efficient data transmission by managing the data rate between sender and receiver.

Flow Control Protocols

- **For Noiseless Channels**
  - **Stop and Wait**
- **For Noisy Channels**
  - **Sliding Window Protocols**
    - → Stop and Wait ARQ
    - → Go Back N ARQ
    - → Selective Repeat ARQ

TCP Congestion Control Algorithms:

❖ It is a technique to prevent network overload by regulating data flow.

TCP Congestion Control
- Slow Start
- Congestion Avoidance
- Congestion Detection
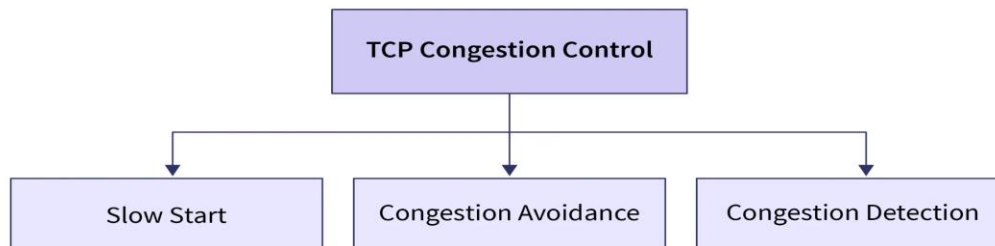
❖ **Slow Start:** Starts with a small congestion window, increases exponentially.

❖ **Congestion Avoidance**: After threshold, window grows linearly.

❖ **Congestion Detection:** TCP identifies this either by a **timeout** (no ACK received) or **3 duplicate ACKs**, and then it reduces the **congestion window (cwnd)** to control the data flow and avoid further congestion.

SCTP (Stream Control Transmission Protocol):

❖ It is a transport layer protocol that ensures reliable, message-oriented communication between endpoints.
   Features:
   o **Multi-streaming:** Multiple streams in a single connection to prevent head-of-line blocking.
   o **Multi-homing:** Supports multiple IP addresses for redundancy and failover.