Computer Networks: Transport Layer

The **Transport Layer** is a critical layer in network architecture. Its main job is to ensure the **end-to-end transportation of data** between the applications running on two different hosts. This layer takes care of all the necessary steps to make sure data is delivered correctly and efficiently to the intended service.

---

1. Core Functions of the Transport Layer

The functions described at the Transport Layer manage data preparation, flow, and final delivery to the application.

i. Identifying the Service (Port Numbers)

Services on a host are identified at this layer using **Port Numbers**. Both TCP and UDP use these numbers to ensure incoming data is directed to the correct application process.

ii. Segmentation, Sequencing, and Reassembling

When the application layer needs to send a large amount of data, the Transport Layer breaks it down into smaller pieces called segments—a process known as **Segmentation**. On the receiving side, the segments are ordered correctly using **Sequencing** and then put back together to reconstruct the original data, which is **Reassembling**.

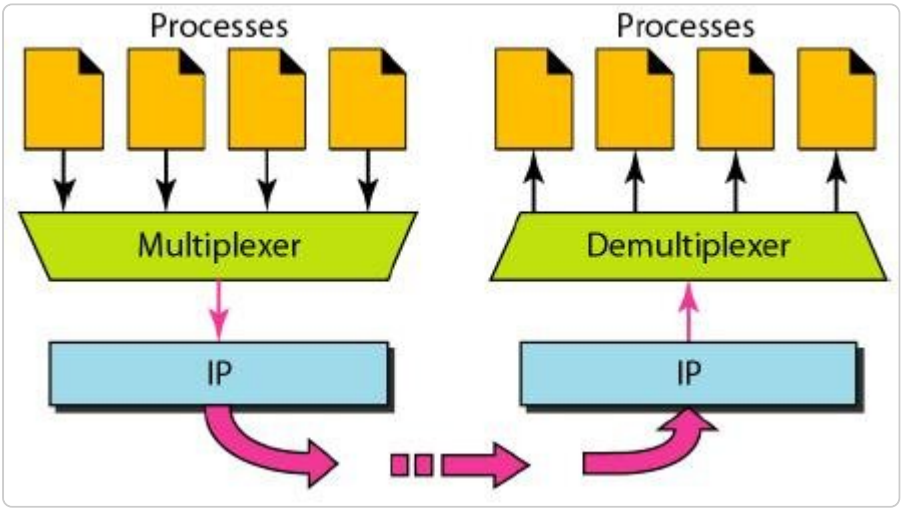

Figure 1: Illustration of data flow, showing Segmentation, Multiplexing, and Demultiplexing processes.

iii. Multiplexing and De-multiplexing

Multiplexing occurs at the sending host, allowing segments from multiple applications to be transmitted over a single network connection. **De-multiplexing** is the reverse process at the receiving host, where incoming segments are sorted and delivered to their appropriate application based on their port numbers.

- **Flow Control**: This function controls the rate of data transmission to prevent the sender from overwhelming the receiver's capacity.

- **Error Correction**: Mechanisms, primarily within TCP, are used to detect and handle data errors or loss during transmission.
- **Protocols**: The major protocols at this layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

---

2. Transport Layer Protocols: TCP and UDP

TCP and UDP are the main protocols that manage data transportation. The key difference lies in reliability and speed.

i. Transmission Control Protocol (TCP)

TCP is a **Connection-Oriented** protocol, meaning it establishes a logical connection before transferring data. It is a **Reliable communication** protocol because it uses **Acknowledgments (Ack's)** to confirm receipt of data. This reliability makes TCP data transportation **Slower**. TCP has a higher overhead with a larger header and the source holds data until it receives acknowledgment. Examples of applications using TCP include HTTP, FTP, and MTP.

ii. Connection Establishment (Three-Way Handshake)

TCP uses a process called the **three-way handshake** to establish a connection or virtual-circuit with the destination. This handshake uses the **SYN** (Synchronization) and **ACK** (Acknowledgement) flags found in the Code Bits section of the TCP header. The process is essential for initializing the sequence and acknowledgement number fields.

TCP Congestion Control (Simplified Window Reduction):

If segments are lost due to congestion at the receiver, the receiver acknowledges the last received sequential segment and replies with a ***reduced window size****.*

$$Window\ Size_{new} \leq Window\ Size_{old}$$

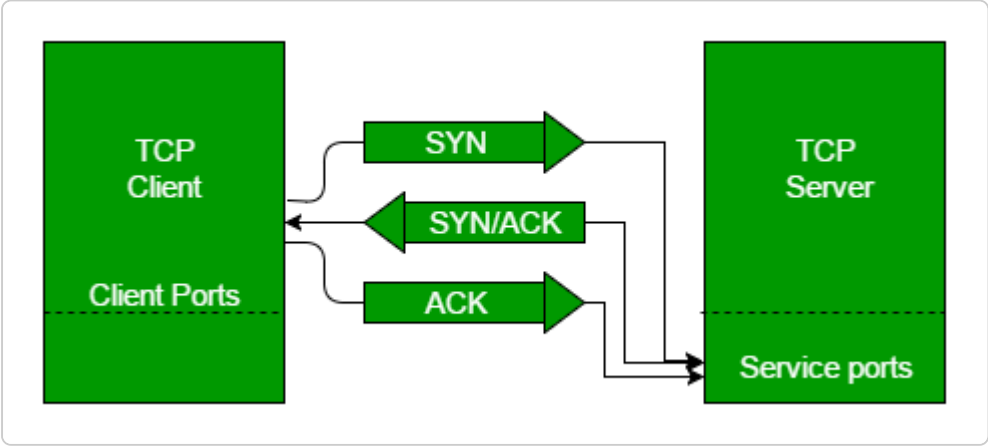

Figure 2: The three-way handshake sequence using SYN and ACK messages for establishing a connection.

- **TCP Header**: Includes fields like Source Port (16 bits), Destination Port (16 bits), Sequence Number (32 bits), Acknowledgement Number (32 bits), and Window size (16 bits).

- **Flow Control Mechanism**: TCP uses a dynamic window size. If segments are lost because of congestion, the receiver will acknowledge the last received sequential segment and reply with a reduced window size.
- **Window Size Benefit**: Larger window sizes increase communication efficiency.

---

3. User Datagram Protocol (UDP) and Comparison

UDP is designed for speed and simplicity, making trade-offs in reliability.

i. User Datagram Protocol (UDP)

UDP is a **Connection-Less** protocol. It provides **Unreliable communication** because it does not use Acknowledgments. This lack of overhead allows for **Faster data transportation**. Applications that can handle reliability themselves or don't strictly require it, such as DNS, DHCP, and TFTP, use UDP.

ii. UDP Header Structure

The UDP header is very simple and small. It contains only four parameters: **Source Port, Destination Port, Length, and Checksum**. The total size of the UDP header is only **8 bytes**.

UDP Header Size (in bits):

Source Port (16 bits) + Destination Port (16 bits) + Length (16 bits) + Checksum (16 bits) = ***64 bits (8 bytes)***

iii. Protocol Comparison Summary

Feature	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Connection Type	Connection Oriented	Connection Less
Reliability	Reliable communication (with Ack's)	Unreliable communication (no Ack's)
Speed	Slower data Transportation	Faster data Transportation
Header Overhead	Higher Overhead	Lower Overhead (8 bytes)

- **Commonality**: The only thing common between TCP and UDP is that they both use **port numbers** to transport traffic.
- **Voice/Video Applications**: Some applications, especially those that deal with voice and video, require fast transport and often use UDP, taking care of the reliability themselves at the application layer.
- **Foundation**: TCP is one of the original protocols designed in the TCP/IP suite, giving the model its name.

Study hard! This note covers all the key concepts.