## Week 3

### 1. Overview of the Transport Layer

* The Transport Layer provides logical communication between processes running on different hosts. It ensures data is transferred reliably (if required) and efficiently between applications.

Key Responsibilities of the Transport Layer:

1. Multiplexing and Demultiplexing – Ensuring data reaches the correct application.
2. Reliable Data Transfer – Handling packet loss and errors.
3. Flow Control – Preventing sender from overwhelming the receiver.
4. Congestion Control – Avoiding network overload.
5. Connection Management – Establishing and closing connections.

Protocols in the Transport Layer

* TCP (Transmission Control Protocol)
  + Reliable, connection-oriented.
  + Ensures in-order delivery.
  + Uses flow control & congestion control.
* UDP (User Datagram Protocol)
  + Unreliable, connectionless.
  + Faster, used for real-time applications.

### 2. Multiplexing and Demultiplexing

Multiplexing allows multiple applications to share the network simultaneously. Demultiplexing ensures data is delivered to the correct application.

* Multiplexing (Sender Side): Adds transport-layer headers (with port numbers) to identify the destination application.
* Demultiplexing (Receiver Side): Uses port numbers to deliver data to the correct process.

Types of Demultiplexing

1. Connectionless Demultiplexing (UDP)

* Uses only the destination port number.
* No tracking of connections.

1. Connection-Oriented Demultiplexing (TCP)

* Uses four values (4-tuple):
  1. Source IP Address
  2. Source Port Number
  3. Destination IP Address
  4. Destination Port Number

1. Allows multiple connections from different clients to the same server.

### 3. UDP (User Datagram Protocol)

UDP is a simple, connectionless transport protocol that provides best-effort delivery.

Characteristics of UDP:

* No connection establishment (reduces delay).
* No reliability (packets may be lost or arrive out of order).
* No congestion control (can send data as fast as needed).
* Small header size (less overhead).

Common Uses of UDP:

* Streaming multimedia (loss-tolerant applications)
* DNS (Domain Name System)
* SNMP (Simple Network Management Protocol)
* HTTP/3

UDP Segment Structure

A UDP segment consists of:

1. Source Port – Identifies the sender’s application.
2. Destination Port – Identifies the recipient’s application.
3. Length – Indicates the size of the segment.
4. Checksum – Provides error detection.

Error Detection in UDP: Internet Checksum

* Sender: Computes checksum and includes it in the UDP header.
* Receiver: Computes checksum of received data and compares it with the sender’s checksum.
* Weakness: Cannot detect certain types of errors (e.g., swapped bits).

### 4. Principles of Reliable Data Transfer

Many applications need reliable communication, so protocols implement mechanisms to detect and correct errors.

Challenges in Reliable Data Transfer:

1. Packet Loss – Some packets may never arrive.
2. Packet Corruption – Some packets may get corrupted in transit.
3. Out-of-Order Packets – Packets may arrive in the wrong order.
4. Duplicate Packets – The sender may resend a packet that was actually received.

Reliable Data Transfer Protocols (RDT)

1. RDT 1.0 (Ideal Case)

* Assumes a perfect channel (no errors or loss).
* The sender sends data, and the receiver receives it without issues.

1. RDT 2.0 (Handles Bit Errors)

* Uses ACK (Acknowledgment) for correct packets.
* Uses NAK (Negative Acknowledgment) for corrupted packets.
* Sender retransmits if a NAK is received.

1. RDT 2.1 (Handles Corrupt ACKs/NAKs)

* Uses sequence numbers to differentiate retransmitted packets.
* Helps avoid duplicate packets.

1. RDT 3.0 (Handles Packet Loss)

* Uses timeouts to detect lost packets.
* If no ACK is received in time, the sender retransmits.

### 5. Stop-and-Wait vs. Pipelined Protocols

Stop-and-Wait Protocol

* The sender waits for an acknowledgment (ACK) before sending the next packet.
* Inefficient due to long waiting times.

Pipelined Protocols (Efficient Alternative)

* Sender can send multiple packets before receiving an ACK.
* Requires larger sequence numbers and buffering at the receiver.

Types of Pipelined Protocols

1. Go-Back-N (GBN)

* The sender can send up to N unacknowledged packets.
* If a packet is lost, the sender retransmits that packet and all subsequent packets.
* Simpler but less efficient.

1. Selective Repeat (SR)

* The sender only retransmits lost packets.
* The receiver buffers out-of-order packets.
* More efficient but requires extra memory.

### 6. TCP (Transmission Control Protocol)

TCP is a connection-oriented and reliable transport protocol used for most applications.

TCP Features:

1. Connection-Oriented

* Establishes a connection before data transfer.
* Uses the three-way handshake process.

1. Reliable Data Transfer

* Ensures correct order and error-free delivery.

1. Flow Control

* Ensures the sender does not overwhelm the receiver.

1. Congestion Control

* Avoids network congestion by adjusting sending speed.

1. TCP Segment Structure

A TCP segment consists of:

* Source Port & Destination Port – Identify the applications.
* Sequence Number – Helps order packets correctly.
* Acknowledgment Number – Confirms received data.
* Flags (e.g., SYN, ACK, FIN) – Control connection setup and termination.
* Window Size – Supports flow control.
* Checksum – Provides error detection.

### 7. TCP Connection Management

1. TCP connections go through three main phases:
2. Connection Establishment (Three-Way Handshake)

* Client sends SYN to initiate a connection.
* Server responds with SYN-ACK.
* Client sends ACK, and the connection is established.

1. Data Transfer

* Data is sent reliably using sequence numbers and ACKs.

1. Connection Termination (Four-Way Handshake)

* FIN-ACK-FIN-ACK exchange closes the connection.

### 8. TCP Congestion Control

* If too many packets are sent, the network gets overloaded.
* TCP adjusts the sending rate based on network conditions.

Congestion Control Mechanisms

1. Slow Start – Starts with a low transmission rate and increases exponentially.
2. Congestion Avoidance – Adjusts the rate carefully to prevent congestion.
3. Fast Retransmit – If three duplicate ACKs are received, TCP retransmits the lost packet immediately.

### 9. Summary

* Transport Layer handles end-to-end communication.
* TCP ensures reliable, ordered delivery with congestion and flow control.
* UDP is faster but does not guarantee delivery.
* Protocols like Go-Back-N and Selective Repeat improve efficiency in reliable communication.
* TCP congestion control prevents network overload.