Module 3 - The Transport Layer.

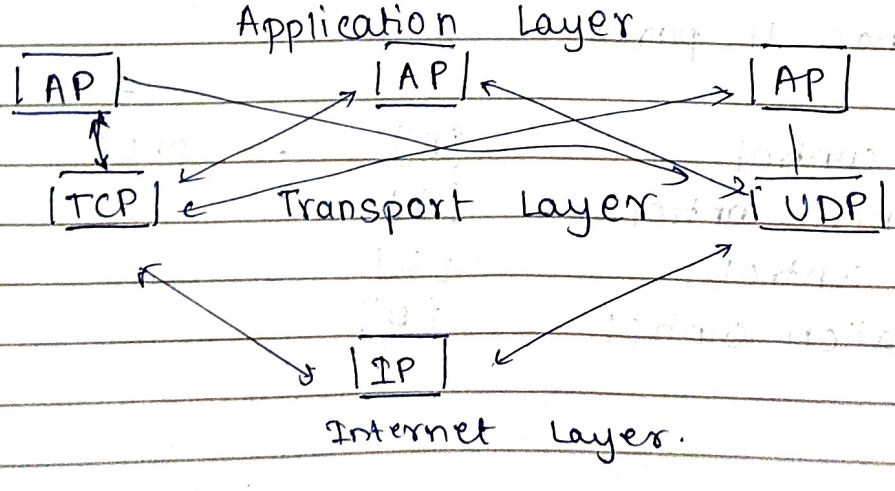
- The transport layer is the 4th layer within a computer network.

- It provides a variety of services but the main role of the transport layer is to provide communication services to application processes running on different hosts.

- The transport layer provides a logical connection between application processes from different hosts since none of them are physically connected to each other.

- The transport layer has two protocols, - TCP and UDP.

- Each of the applications in the application layer has the ability to send a message using either TCP or UDP. The applications can read and write to the transport layer. Hence we say this communication is a two-way process.



* Services of Transport Layer.

- Services provided by transport layer are divided into 5 categories:-

1] End-to-end delivery — E
2] Addressing — A — ['DRAFE']
3] Reliable delivery — R
4] Flow control — F
5] Multiplexing. — D

* End-to-end delivery.

The transport layer transmits the entire message from source to destination. Providing end-to-end delivery of the message.

* Reliable delivery.

Retransmission of lost and damaged packets is providing reliability of message delivery.

It has 4 parts.

1] Error control — [E] → 'DELS'
2] Sequence control. — [S]
3] Loss control — [L]
4] Duplication control. — [D]

* Error control.

→ No transmisson is error-free. So the transport layer protocols are designed to provide error-free transmission.

→ The error-free experience is the main role of the data link layer but it only ensures node-to-node error correction. It doesnt ensure end-to-end error free reliance.

→ If there is an error in the router, the data-link layer cannot detect it. It can only detect errors in the beginning and at the end of the link. But the transport layer makes sure the message arrives correctly.

---

* Sequence control.

→ This aspect makes sure that the packets recieved from the upper layers are accessible to the lower layers.

→ Also the segments can be re-arranged according to the priority provided by the user.

---

* Loss control.

→ Makes sure all the messages sent from the source are recieved at the destination, not some of them. Every single message is given a sequence number, allowing to identify any missing segment.

* Duplication control.

→ Guarentees that data being recieved at the destination is not duplicated preventing wastage of storage space. Sequence number can also identify lost packets.

* Flow control.

→ Used to prevent the sender from overwhelming the reciever.

→ If the reciever is overloaded with packets, then it discards the packets which need retransmission.

→ It uses the sliding window protocol that makes data transmission more efficient as well as controls the flow of data.

* Multiplexing

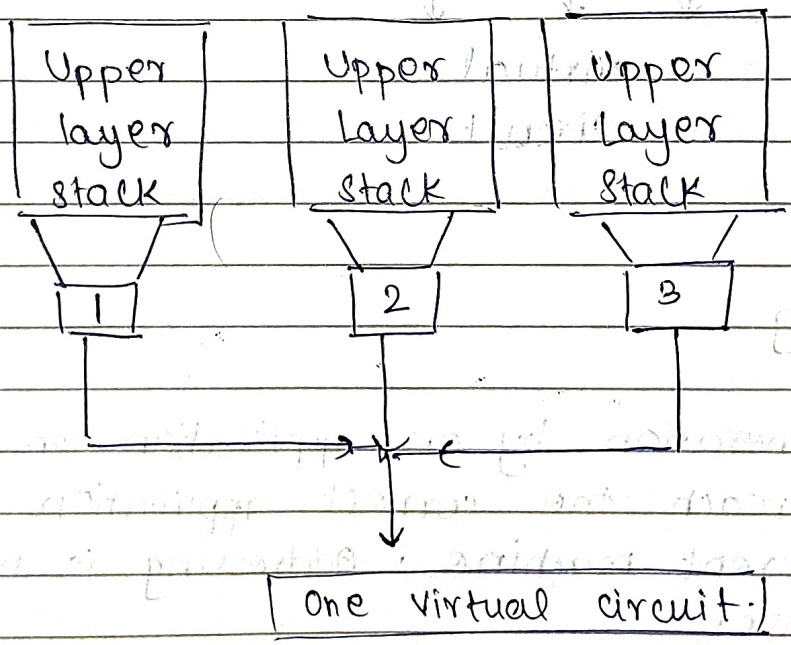
→ Can be of two types :-

1] Upward M.P
2] Downward M.P.

→ Upward Multiplexing

This means multiple transport layer connections use the same network connection.

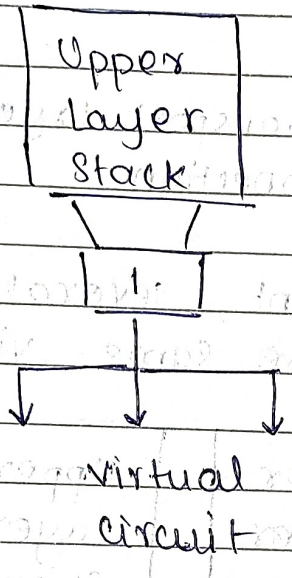
Making this cost efficient, several transmissions are sent through the same virtual circuit



→ Downward Multiplexing.

When one transport layer connection uses multiple network connections.

This allows to split connection between several paths to improve accuracy and throughput.



---

* Addressing

→ Data generation by an application on one machine should reach the correct application on same or different machine. Addressing is helpful in these cases.

→ The T.L provides the user address which is specified as the station or port.

This port variable represents a T.S user of a station called Transport service access point. (TSAP).

* Transport layer protocols.
* ~~UDP~~ UDP

→ UDP (User Data gram Protocol) is a simple protocol providing non-sequential transmission functionality.

→ UDP is a connectionless protocol.

→ UDP focuses on speed and size efficiency rather than reliability and security.

→ This type of protocol adds transport-level addresses, checksum error control and length into to the data from the upper layer.

* format of UDP packets.

Source port address (16-bits)	Dest. port address (16-bits)
Total length. (16-bits)	Checksum (16-bits)
Data.	

In this diagram we see.

1] Source port address - Defines the address of the application process that has delivered the message.

2] Destination Port address - It defines the address of the application process which will recieve the message.

3] Total length - defines the total length of the user datagram (16-bit)

4] Checksum - Used in error-detection.

* Applications of UDP.

1] Lossless Data transmission.

2] Gaming, voice calls and video calls.

3] Multicasting and routing updates

4] Fast applications.

* Disadvantages of UDP.

1] Doesnt provide sequencing or re-ordering of user datagrams.

2] Doesnt give specifications about which data packet is lost during loss control. There are no sequence numbers here.

---

* TCP (Transmission control Protocol)

→ TCP stands for Transmission control Protocol.

→ It is a connection oriented protocol which means a connection is established between both ends of the transmission.

→ TCP generates a virtual circuit between sender and reciever, for the duration of transmission.

---

* Features of the TCP protocol.

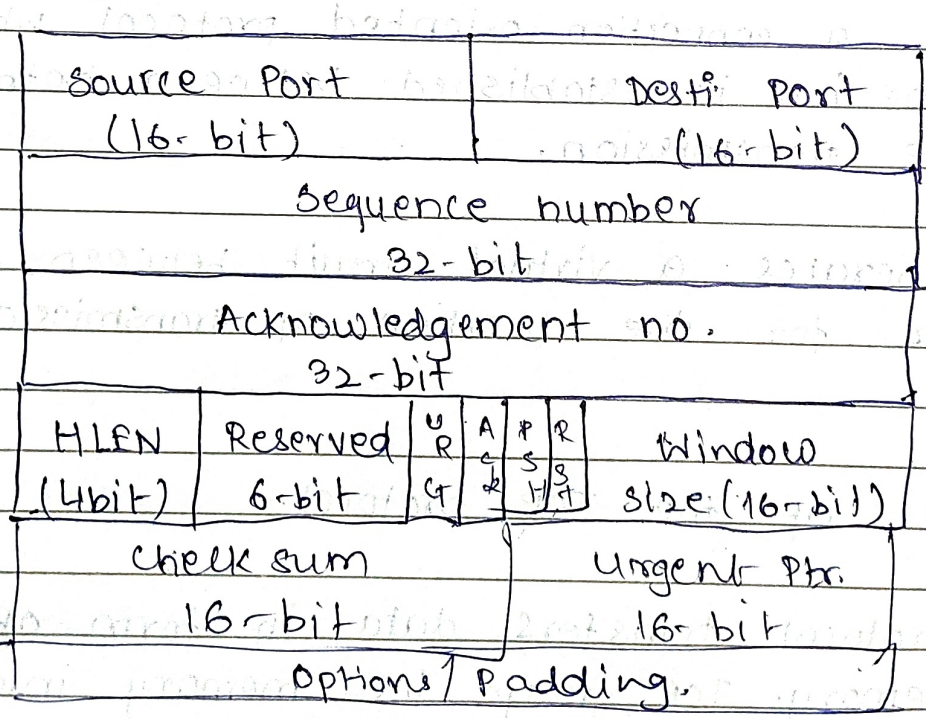
1] TCP protocol transfers data in form of a stream of memory. TCP groups the memory into segments and passes it to the IP layer for transmission to the destination.

2] TCP assigns sequence numbers to each of the segments and upon reaching its destination, expects a positive acknowledgement. If this acknowledgement is not recieved then retransmits the data.

3] Prevents the sender from overflowing the reciever.

4] * Allows multiplexing for various situations.

* TCP segment Format.



Here.

1] Source Port Address:- Address of the application program of source computer.

2] Destination Port Address:- Address of the application program of destination computer.

3] Sequence Number:- Represents the position of a particular segment of data in a data stream.

4] Acknowledgement Number - Acknowledges data from other communicating devices. If ACK field is 1 then it specifies the sequence number that the reciever is expecting to recieve.

5] Reserved - Reserved for future use

6] Header Length (HLEN) - Specifies the size of the TCP header in 32-bit words. The minimum size of the header is 5 words, and max is 15 words.

Control Field :- Defines the use of the segment.

a] URG - Indicates if msg is urgent

b] ACK - Validate acknowledgment number.

c] PSH - Informs that a higher throughput is needed.

d] RST - resets the TCP connection if there is confusion in sequence number.

e] SYN - Synchronizes the msg to 3 parts, request, confirmation and acknowledgement

f] FIN - When sender has finished sending data.

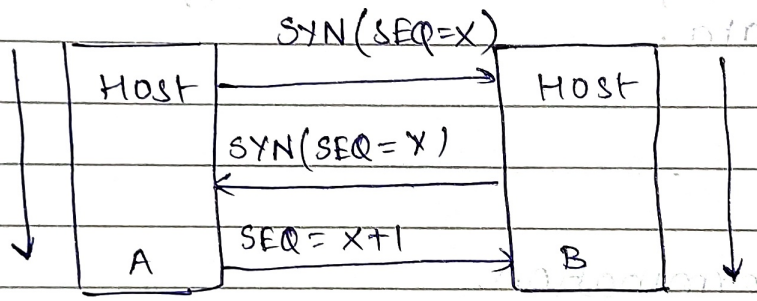
→ Window Size :- Defines size of the window

~~Module 4~~

* TCP connection Establishment

→ Connection Establishment is performed by using a 3-way handshake mechanism. This synchronizes both ends of a network and helps agree on orignal sequence numbers.

→ This mechanism also provides both the sides to transmit data and learn if the other side is able to communicate.



1) The requesting end (A) sends an SYN segment with initial seq. no. 'X'.

2) The server (HOST B) acknowledges his own SYN segment with seq. no. Y and accepts the sender with seq no. X+1

3) An SYN consumes seq. no., then the client acknowledges this SYN from server by accepting SEQ. +1 (seq = X+1, ACK = Y+1)

* Data Transfer.

→ After connection is established, bidirectional data transfer is possible

→ Message is divided to 4 segments, first 3 send both data and acknowledgement while the fourth segment sends only an acknowledgement as there is no more data to be sent.

→ When reciever gets the package it increases the acknowledgement number by the length of recieved data.

* connection Termination.

→ The client TCP, after recieving the close command from the client process, sends a FIN segment which consumes only one sequence number.

→ The server TCP, after recieving the FIN segment, informs the process of the situation and sends the second segment a FIN + ACK segment to confirm the reciept of the FIN segment

→ The client segment then sends a last msg. an ACK segment to confirm ACK segment from server TCP