Date __/__/____

MOD - 3

Transport layer.

A T N D P.

Saathi

Transport layer:

- End to end delivery. [port to port]
- uses TCP & UDP protocol
  - TCP → better
  - UDP → worst

TCP → connection oriented
→ reliable
→ Inorder delivery
→ no loss of data
→ error control, congestion control, flood control. (checksum)

UDP → not reliable
→ datagram entity is sent to other side.
→ loss of data
→ fast.

⇒ • main role of Transport layer is to provide communication services directly to app. process at diff. hosts.

- Provides a logical connection, both the app. process running on different host, they are not physically connected but get connected by logical connection from transport layer.

- The transport layer protocols are implemented in end systems (computers) but not in routers.

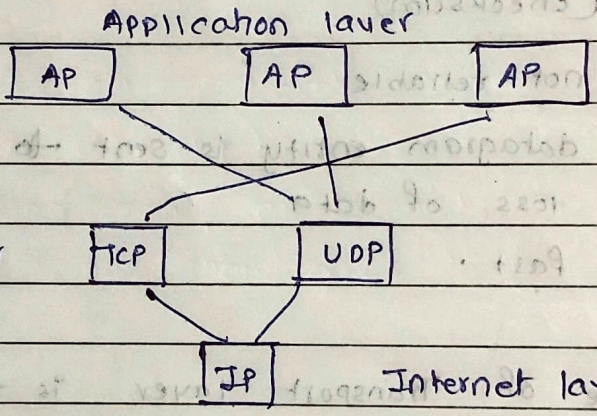
- TCP & UDP — Transport layer protocols.
- All transport layer protocol provide multiplexing and demultiplexing.

Date __/__/____

* • Each of applications in app. layer has ability to send msg by using TCP or UDP.

• App. can communicate using TCP & UDP.
• App. can read & write in transport layer, thus communication is a two way process.

① App layer (App) communicate with TCP / UDP. they can send msg using TCP / UDP.
② TCP / UDP further communicates to Internet layer



* Services Provided by ~~TCP~~ TL:

• Similar to services of datalink layer..

(EARFM) ① End to end delivery.
② Addressing
③ Reliable
③ Flow control, error & congestion control.
④ Multiplexing.

Page No.

* data link layer also provides node to node error control
* error will not get caught if error is given in bet routers.

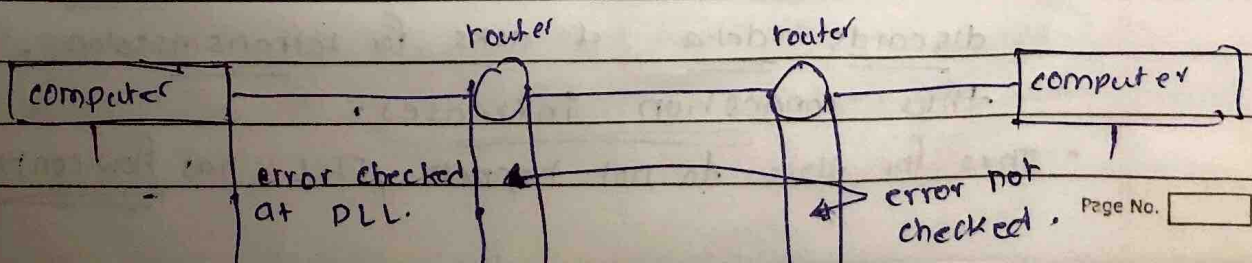
Saathi

Date __/__/____

- End to end delivery:
  - Transmits the entire data to the destination.
    thus ensures end to end delivery.

- Reliable delivery:
  - provides reliability over lost packets by retransmitting
  - we get, • Error control [ESLD]
    • sequence control
    • loss control [DELS]
    • Duplication control

* Error Control:
  - transport layer protocol are designed to provide error free transmission.
  - Data link layer also provides error handling mechanism, but it checks for error only at node-to-node.
    So Data link layer check for error in each network, But if error is introduced in routers, error will not get caught in data link layer.
  - Data link layer checks error only at start & end.
    But Transport layer checks for error, end to end thus, packet is arrived correctly.



* seq. no are used to identify missing & duplicates.

Saathi

Date __/__/____

* Sequence Control :

- Thus on the sending end, transport layer is responsible for packets from upper layer ~~etc~~ can be used by lower layers.

- on receiving end, the segments sent are reassembled properly.

* Loss Control :

- It ensures all segments arrive at the destination properly.
- on sending end, all segments are given sequence numbers, thus allowing us to identify the missing

* Duplication Control :

- no duplicates arrive at receiving end.
- sequence no. are used to identify the duplicates.

Flow Control

- If receiver is overloaded & sender sends in more segments / data, then the receiver discards data & asks for retransmission, thus congestion increases.
- Thus for this to not happen, T.L. has flow control.

Page No.

* sliding window protocol → handles data properly for transmission & handles flow control.
→ byte oriented than frame oriented.

Saathi

Date __/__/____

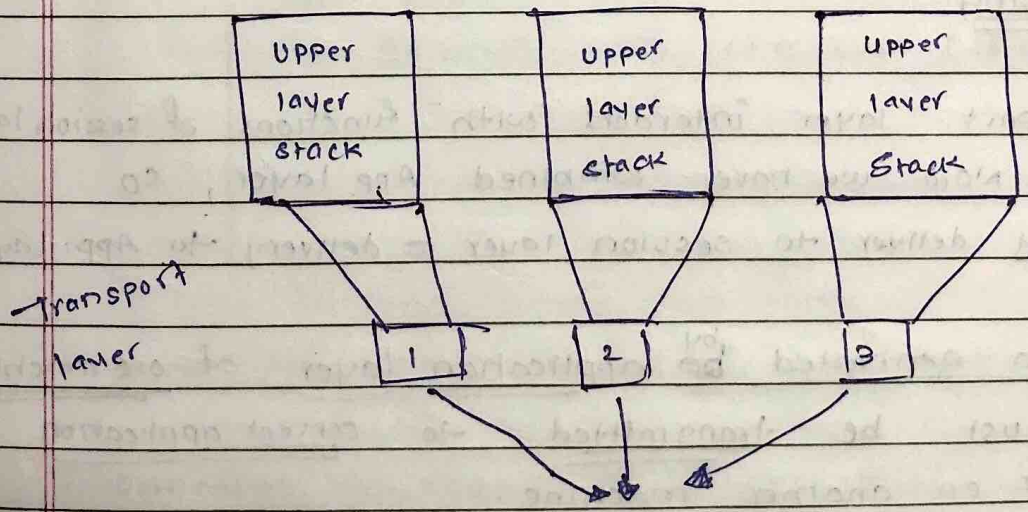
- T.L. uses sliding window protocol, helps in smooth data transfer. & controls the flow of data.

* Multiplexing → T.L. supports multiplexing.

- Multiplexing of 2 types ① upward Multiplexing
② Downward.

① Upward Multiplexing:

- multiple transport layer connection use same one network connection. ≡ multiplexing (many to one)
- to make more cost effluent; T.L. sends several transmission bound for same destination along same path.



upward
* multiplexing ⇒ virtual circuit has 1 path. (upward)
⇒ several transmission on same ~~path~~ for ~~along~~ same path.

Page No.

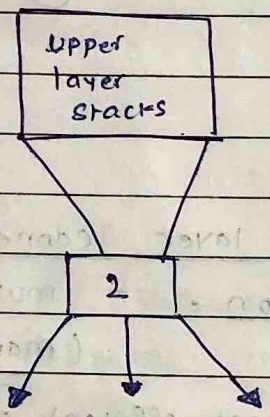
Downward → 1 to many (T.L. connection, net. L. connec.
→ T.L. connec. splits into several parts, increase accuracy
upward → many to one (T.L. connectn, net. L. connectn)
→ many T.L. connectn to same destn are sent through same path → cost efficiency.

Saathi

Date __/__/__

② Downward Multiplexing:

- One transport layer connection uses multiple network layer connections.
- allows transport layer to split a connection among several paths to improve accuracy.



← one T.L. connectn.

Transport layer.

one virtual circuit.

Downward

Addressing

① Transport layer interacts with functions of session layer.
② Now we have combined App layer, so any deliver to session layer = delivery to App. Layer.

* * So data generated by application layer of one machine must be transmitted to correct application on another machine.
here addressing is used.

* T.L. provides the port. Port represent a particular transport service (TS) user of a point called TSAP, transport service access point.

* T.L. needs to know, which upper layer protocols are communicating.

Page No.

* Ip protocol in network layer delivers datagram from source host to destination host
* Port = 16 bits

Date __/__/____

Saathi

Protocols of T.L ① UDP ② TCP.

① UDP → user datagram Protocol.
* msg oriented

- It is a simple protocol & provides no sequential transport of data
- connectionless protocol
- used when reliability & security is less important.
- end-to-end T.L. protocol, includes transport level address, Checksum error control, & information of data from upper layer. ^ total length.
- Packet produced by UDP = datagram.

* user datagram Format (16 byte header).

Source port address (16 bit)	destination port address (16 bit)
Total length (16 bits)	checksum (16 bits
Data.	

- Source Port add. = address of app. process that delivered msg.
- Destination port add. = add. of app. process that will receive msg
- Total length = total length of user datagram (in bytes)
- Checksum = used in error detection.

Page No.

UDP → error detection = checksum

* TCP itself segments the data

Date __/__/____

* App. of UDP
① Fast ② Gaming, voice calls.

Saathi

• • Disadvantage of UDP : Adv = fast & sent in form of entities.

• does not provide sequencing, reordering,
• no reliability
• Data gets lost.
* • can discover error, but cannot tell which packet is lost, does not have sequence no.

② TCP → Transmission Control protocol.

• Provides full t.l. services to applications.
• Connection oriented ⇒ at first creates a connection betn two ends, & generates Virtual circuit betn sender & receiver for the duration of transmission.

* Features :
① stream data transfer.
② Reliable
③ Flow control
④ Multiplexing
⑤ logical connection
⑥ Full duplex.

* Stream data transfer : TCP transfers data in form of contagious stream of byte. It groups the byte in the form of segment & passes it to IP for transmission.

① when we send data, receiver send ack. of received data.
② If Ack not received, resend data.
③ when Ack contains, bytes it can handle without overflow,
④ bytes are sent again at max no.

Saathi

Date __/__/____

* Reliability: TCP assigns sequence to each byte transmitted & expects an +ve acknowledgement of bytes received from receiver TCP.

** • If ACK is not received in interval timeout, the TCP, resends the data.

* Flow control: so, when receiver sends Ack back to sender, of how many bytes it can receive without congestion.

• The no. of bytes sent in Ack in the form of highest sequence number that it can receive without any problem.

⇒ Window mechanism.

* Multiplexing: process of accepting & forwarding data.

forwarding data to correct app. = demultiplexing.

* TCP sends data through ports

* logical connection: ~~see~~ combination of sockets, seq. no., window size, = logical connection.

Full Duplex = data flows in both directions at same time.

IP address = unique 32 bit string of numbers that identifies each computer using internet protocol to communicate over network.

UDP ⇒

source	destination
total length	checksum
data	

Date ___/___/_____

UDP → 8 bytes

TCP segment → 20 bytes

Source port address (16 bits)								Destination port address (16 bits)
Sequence no (32 bits)								
Acknowledgement no (32 bits)								
HLEN (4 bits)	Reserved (6 bits)	U R G	A C K	P S H	R S T	S Y N	F I N	window size (16 bit)
Check sum (16 bits)								Urgent pointer (16 bits)
X								X
options & padding.								

* Source port add = add. of app. prog. of source computer.
* sequence no = stream of data is divided in to segments, seq. no. represents ~~data~~ position of data in OG data stream.
* ACK no. = acc. data from other communicating devices.
  If ACK = 1, receiver is expecting to receive.
* Reserved = reserved for future.
* Window size = defines size of window in flow control.
* Checksum = error detection.
* Urgent ptr = if URG = 1, then urgent ptr is offset from seq. no. indicating that it is last urgent data byte.
* opt & padding = convey additional info. to receiver.
* HLen → size of tcp header. in 32 bit words
  min size = 5 words max = 15 words.

full duplex ⇒ data transmission on both sides

TCP = client server.

Date __/__/____

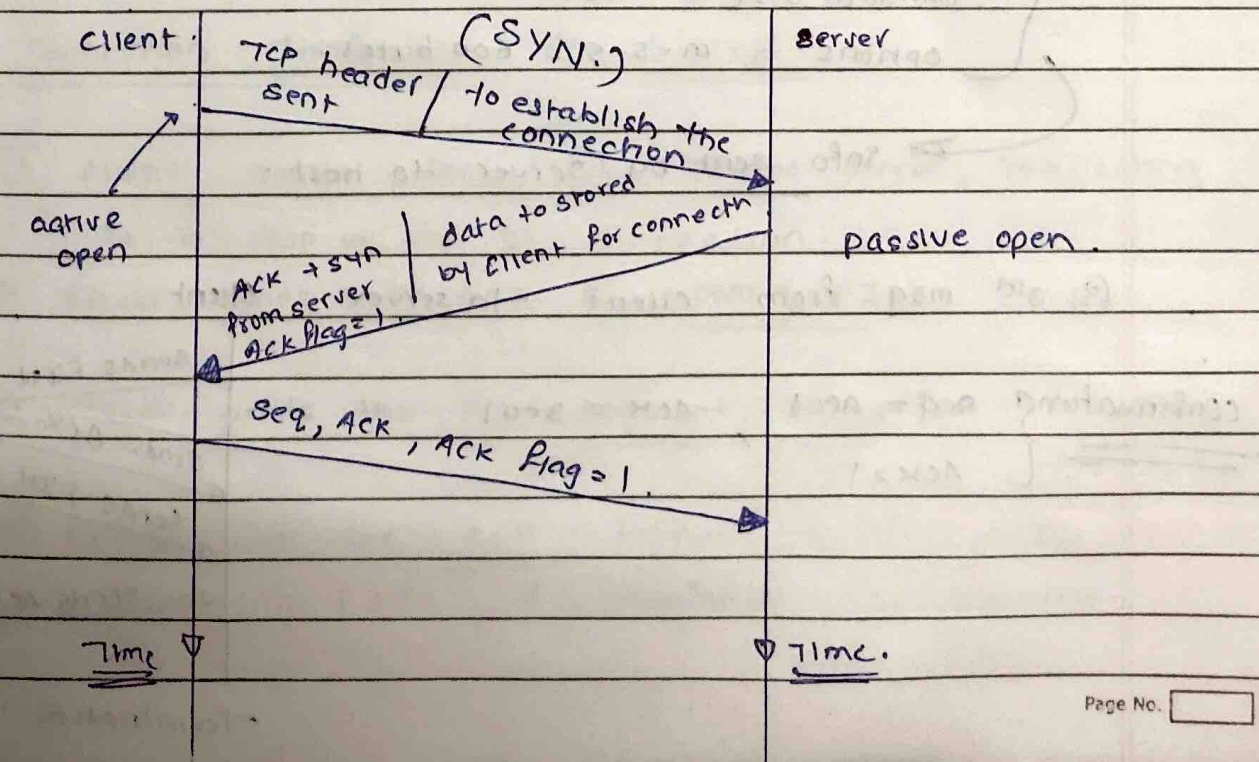
* Control bits (URG, SYN, ACK, RST, PSH, FIN).

- URG = data in segment is urgent
- ACK = if it is set, the it validates the ack no.
- PSH = informs that higher throughput is needed.
- RST = resets TCP connection if confusion in seq. no.
- SYN = synchronleses the msg. to 3 parts, request, confirmation & acknowledgement.
- FIN = sender has finished sending data.

TCP connection establishment

- connection establishment is performed by using 3 way handshaking. This syncs both ends of the network. & helps agree on og. seq. no.

- mechanism also provides both the sides to transmit data. & know if other side is able to communicate.



Page No.

seq. no = $0 - (2^{32} - 1)$

ACK = seq + 1

Date __/__/____

Saathi

① 1st msg sent by A to establish the connection.

, data is not transmitted.

port = A. , seq = 9000.

SYN = 1, to synchronies with server.

send the window size. = 12000.

options = max segment size = 1200 byte.

we give this data in 1st segment sent, to establish connection.

* now server stores this info.

② 2nd msg from server to client, server will give its info. along with syn + ACK (your packet received).

port no = 80, dest. port = A

seq = 3000, ACK 9001

ACK = 1, SYN = 1.

window size = 18000.

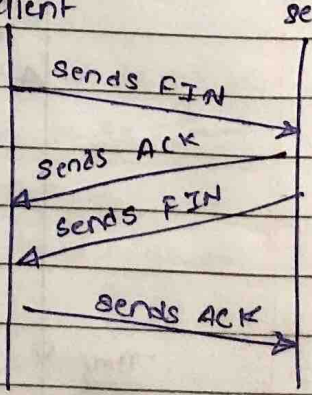
options = m.s.s = 600 bytes.

info sent by server to host.

③ 3rd msg from client to server.

confirmations { seq = 9001, ACK = 3001

ACK = 1



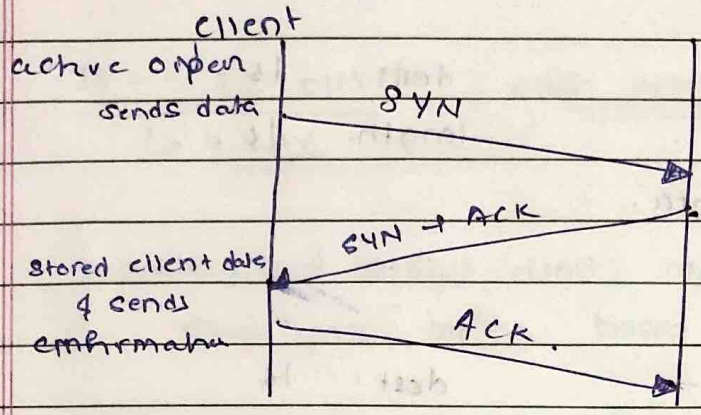
Terminated.

size of TCP header = 20 bytes

Flags ⇒ ACK, URG, RST, PSH, FIN, SYN

UDP header size = 8 bytes.

Date __/__/____



Data transfer:

- once connection is established, bidirectional data transfer is done
- msg is divided in 4 segments.

→ first three sends both data & acknowledgement & last sends only acknowledgement

- when receiver gets msg / packets, it increases the ack by (length) of received data.

Connection Termination → 4 steps.

① client sends the FIN to the server, requesting it to stop or end the connection.

② server sends the ACK of receiving FIN.

③ server sends the FIN segment, ready for termination.

④ client sends the ACK segment to server, for receiving FIN, & terminates the connection.

Page No.