5. Transport Layer.

Page No.
Date

- Transport layer is second layer in TCP/IP model & fourth layer in OSI model.
- It is an end to end layer used to deliver msg to host.
- it provide point-to-point connection rather than hop-to-hop. betⁿ source & destination to deliever service reliably.
- Unit of data encapsulation in Transport layer is a segment.

working:
- takes services from Network layer & provides services to Application layer.

At the sender's side:
- receives data(msg) from application layer & perform segmentation, divides actual msg into segments, add source & destination port no. into header of segment & Transfer msg to NW layer.

At receiver side.
- reassembles the segmented data, reads its header, identify port no. & forwarded the msg to appropriate port in application layer.

Functions of Transport layer:
① process to process delivery.
- Transport layer requires port no. to correctly deliever the segments of data to correct process amongst multiple processes running on a particular host.

Date

A port no. is 16 bit address used to identify any client-server program uniquely

End to end connection:
- create end-to-end connection b/w hosts for which mainly uses TCP and UDP.
- TCP, connection-oriented protocol that uses a handshake protocol to establish robust connection.
- UDP is a stateless & unreliable protocol that ensures best-effort delivery

Multiplexing & Demultiplexing: acquired
- multiplexing is when data is ~~acquired~~ from no of processes from sender & merged into one packet along with header & sent as single packet.
- processes are differentiated by their port no.s
- Demultiplexing required at receiver side when msg is distributed into diff processes

congestion control:
- congestion is situation in which too many sources over n/w attempt to send data & router start overflowing due to loss of packet occurs.
- In this situation TL cont provides congestion control in diff ways.
- It uses open loop congestion control to prevent the congestion & closed-loop congestion control to remove congestion in a n/w once it occured

- Tcp provides AIMD: Additive increase multiplicative decrease, leaky bucket technique for congestion control

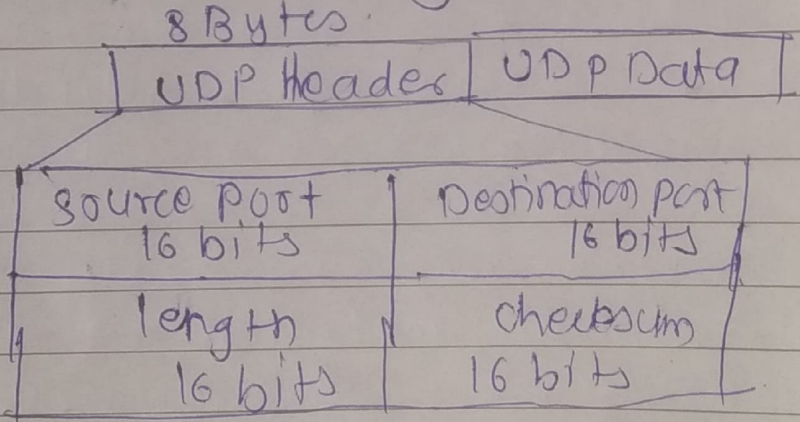
error correction
— TL checks error in msg from application layer by using error detection codes, computing checksum. it checks whether received data is not corrupted & uses the Ack & NACK service to inform the sender if data has arrived or not.

flow control:
— TL provide flow control mechanism blw adjacent layer of TCP/IP model.
- Tcp also prevent data loss due to fast send & slow receiver some flow control tech → uses method of sliding window protocol which is accomplished by receiver by sending a window back to sender informing size of data it can receive.

Date

UDP: (User Datagram Protocol)
- it is an unreliable & connectionless protocol, so there is no need to establish a connection prior to data transfer.
- UDP enables process to process communication.
- For real time services like computer gaming etc we need UDP.
- UDP permits packets to be dropped, instead of processing delayed packets.
- no error checking in UDP. so it also saves bandwidth.

UDP header: is an 8 byte fixed & simple header.
- UDP port no. fields are each 16 bits long, so range for port no's is defined from 0 to 65535. port no. 0 is reserved port no. help to distinguish ~~to~~ diff user requests or processes
- first 8 byte contains all necessary header info & remaining consist of data



1. sourceport: is a 2 Byte long field used to identify port no. of source

Page No.
Date

Destination port : 2 Byte long field, used to identify port of destined packet

Length : Length of UDP including header & the data. 16 bits field.

Checksum : 2 Bytes long field complement 16 bit - one's complement of one's ~~complement~~ sum of UDP header, the Psuedo header & of information from IP header.

Application of UDP:
- Used for simple request-response communication when size of data is less & hence there is lesser concern about flow & error control
- it is a suitable protocol for multicasting as UDP supports for packet switching
- used for some routing update protocol like RIP.
- UDP is used for real time application which cannot tolerate
- UDP as a transport layer protocol uses following implementations
  1. NTP (Network time protocol)
  2. DNS
  3. DHCP.
- Application layer can do some task via UDP -Trace route, -Record Route, -Timestamp
- UDP takes datagram from Network layer, attaches its header & sends it to user so, it works fast.

TCP: Transmission control Protocol

- most common transport layer protocol
- It works together with IP & provides a reliable transport service between processes using the network layer service provided by IP protocol.

- services provided by TCP to application layer.

1) Process to Process communication:
- Tcp provides process to process communication i.e. transfer of data takes place between individual process executing & systems.
- This done using port no.

2) stream oriented:
- data sent & received as a stream of bytes
- Tcp groups a no. of bytes together into a segment & adds a header to each of these segment & delieves these to n/w layer.
- At n/w layer, each of these segment is encapsulated in IP packet for transmission

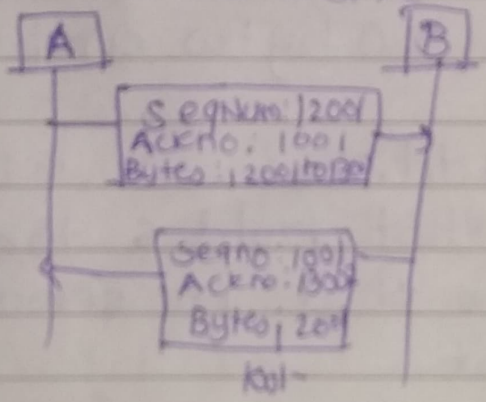
3) Full-duplex service:
- communication can take place in both direction at same time

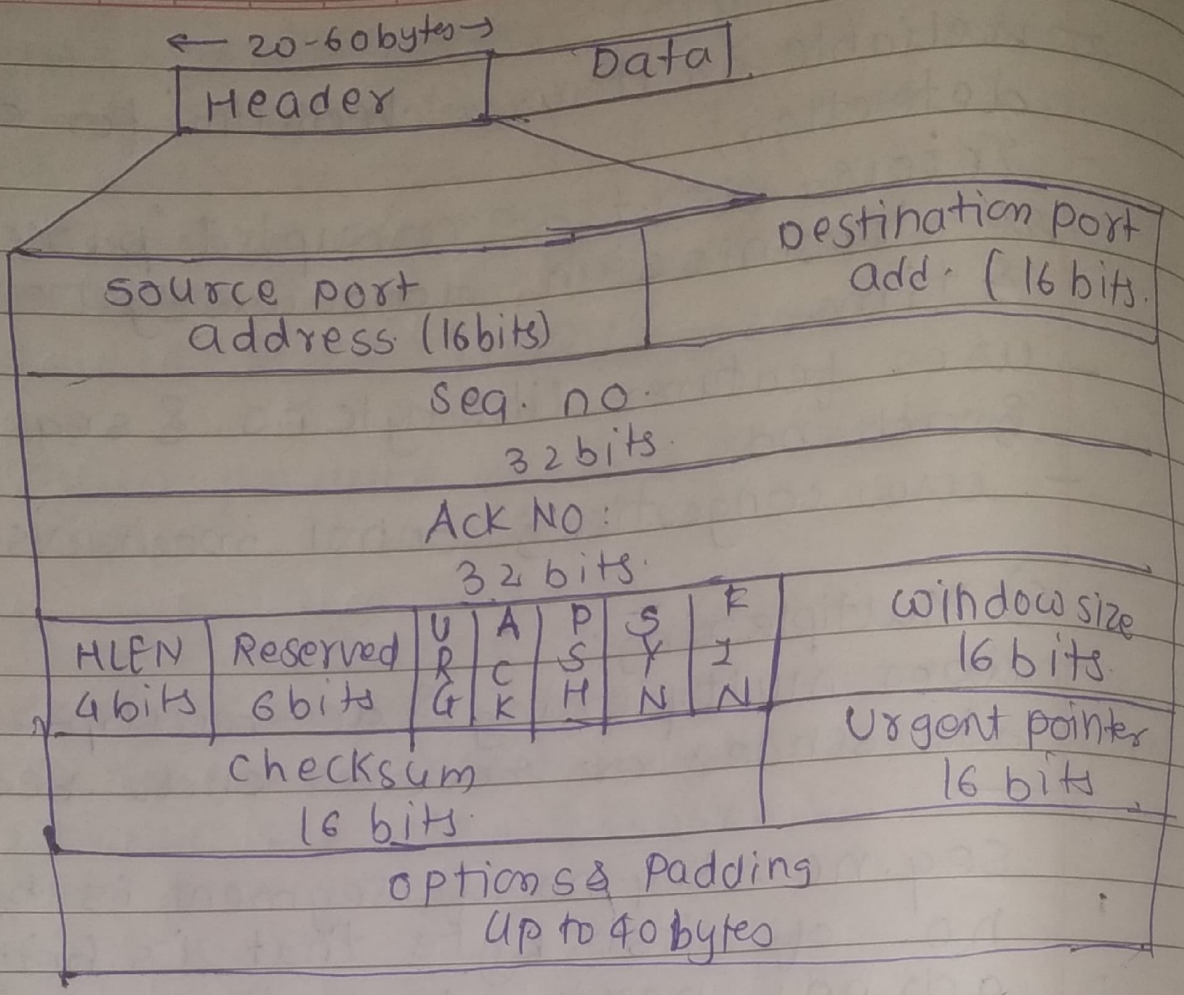
4) connection-oriented service
- 3 diff. phases
1) connection establishment
2) Data transfer 3) connection termination

5) Reliability:
- reliable as it uses checksum for error detection,
- recover lost or corrupted packets by re-transmission, acknowledgement policy & timers
- uses features like byte no. & seq. no. & ack no.
- uses congestion & control mechanism

6) Multiplexing:
- does multiplexing & de-multiplexing at sender or receive ends resp.

Seq. no: seq. no. of segment is byte no. of first byte that is being sent
ack no: next byte no that receive expect to receive





header of Tcp segment can range from 20-60 bytes. 40 bytes are for options

- source port address:
  16 bit field that holds the port add of application that is sending the data segment

- Destination port address.
  16 bit field that holds port add. of application in host that is receiving data segment

- Seq. NO:
  A 32-bit field that holds Seq. no.
  i.e byte no. of first byte that is sent in

that particular segment. used to reassemble the msg at reciving end of segment that are received out of order

- ACK NO:
- 32 bit. i.e.
- the byte no. that receiver expects to receive next.

- Header length (HLEN):
- This is 4 bit indicates length of TCP header by no. of 4 byte words in header.
- min length = 20 bytes this field will hold 5 & max length = 60 bytes. this field will hold 15.
- the value of this field always bet" 5 & 15.

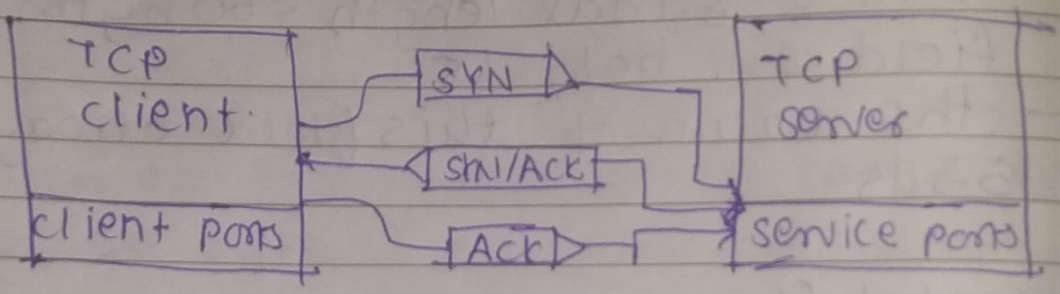
- Control flags:
there are 6 1-bit control bits that control connection establishment.
  - URG: Urgent pointer is valid
  - ACK: ack. no. is valid
  - PSH: Request for push
  - RST: Reset the connection
  - SYN: Synchronize seq. no.
  - FIN: Terminate the connection

Window size:
- window size sending TCP in bytes

checksum:
- that for error control:

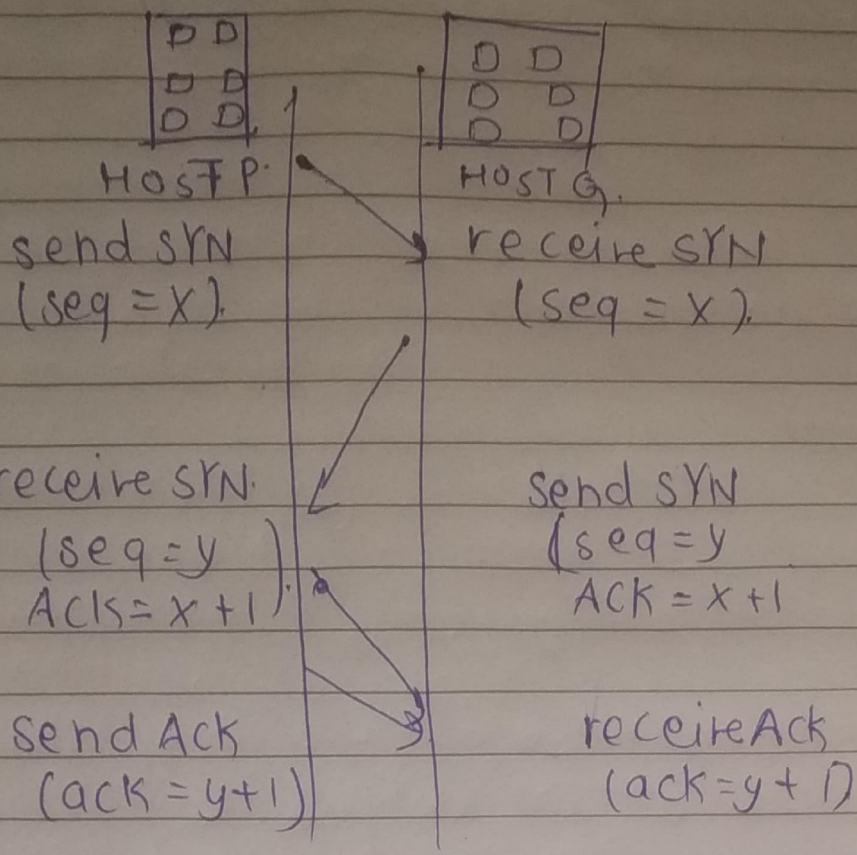
Urgent pointer:
- used to point to data i.e. urgently required that needs to reach receiving process at earliest
- value of this field is added to seq. no. to get byte no. of last urgent byte

TCP is connection-oriented. A TCP connection established by a 3-way hand shake



- Tcp provide reliable communication with something called Positive acknowledgement with re-transmission.
- Protocol Data Unit of transport layer is called a segment
- device using PAR resend the data unit until it receives an acknowledgement. if data unit received at receiver's end is damaged, receiver discards the segment.
- So, sender has to resend the data unit for which positive ack is not received
- This mechanism i.e. three segments are exchanged bet^b sender (client) & Receiver (server) for reliable TCP connection to get

established.



Step 1 (SYN):
- client wants to establish a connection with a server, so it sends a segment with SYN which informs that server that client is likely to start comm. with what seq no. it starts segments with.

Step 2 (SYN + ACK):
- server responds to client request with SYN-ACK signal bits set. Ack signifies the response of segment it received & SYN signifies with what seq. no it is likely to starts segments with.

Step 3 (ACK):
- client acknowledges the response of server & they both establish a reliable connection with which they will start the actual data transfe