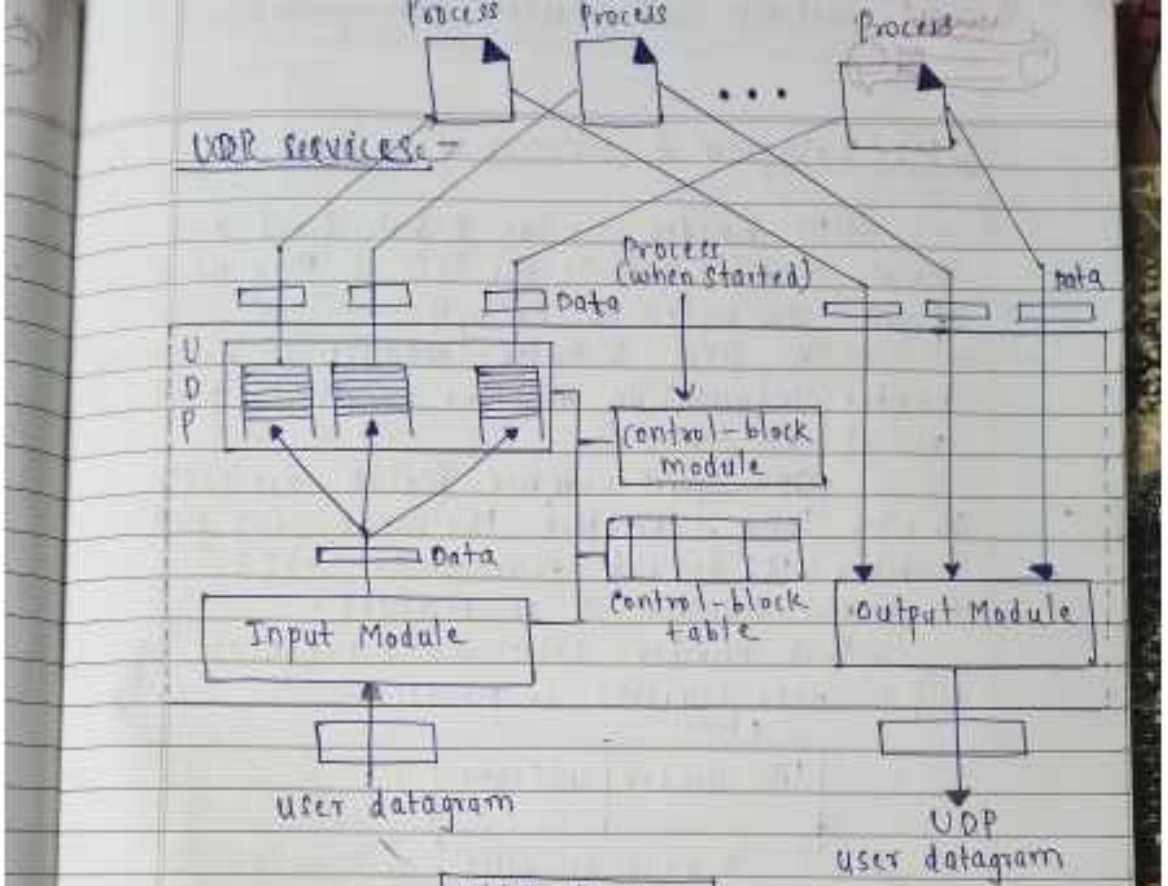
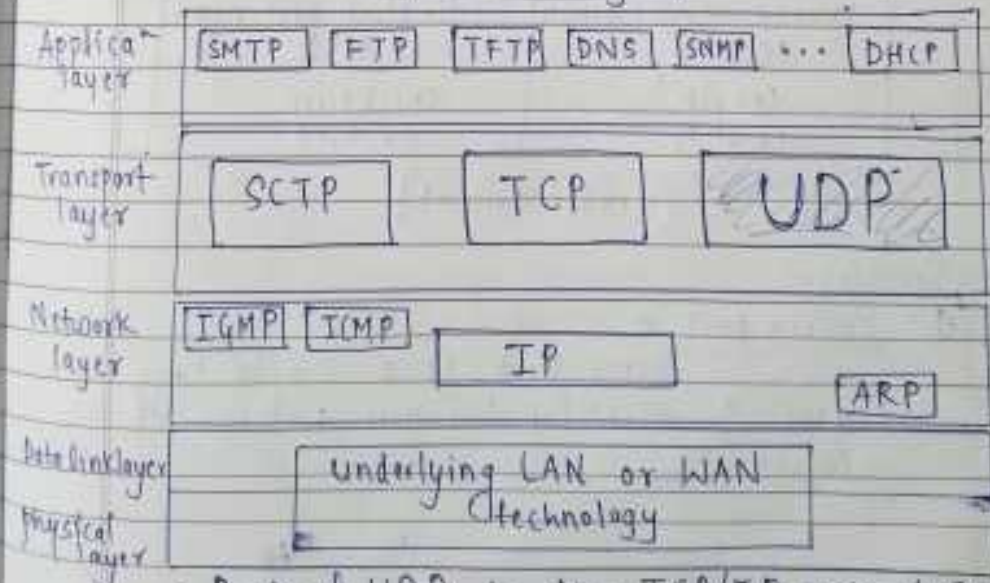
Transport Layer -

• UDP - datagram, services, applications -

- User Datagram protocol (UDP) is a transport layer protocol.
- UDP is a part of the Internet protocol suite, referred to as UDP/IP suite.
- unlike, TCP it is an unreliable and connectionless protocol.
- So there is no need to establish a connection prior to data transfer.
- The UDP helps to establish low-latency and loss-tolerating connections establish over the network.
- The UDP enables process to process communication.
- UDP comes into picture for real-time services like computer gaming, voice or video communication, live conferences; we need UDP.
- Since high performance is needed UDP permits packets to be droped instead of processing delayed packets.
- It saves Bandwidth
- It is more efficient in terms of latency and bandwidth.



• UDP Design •

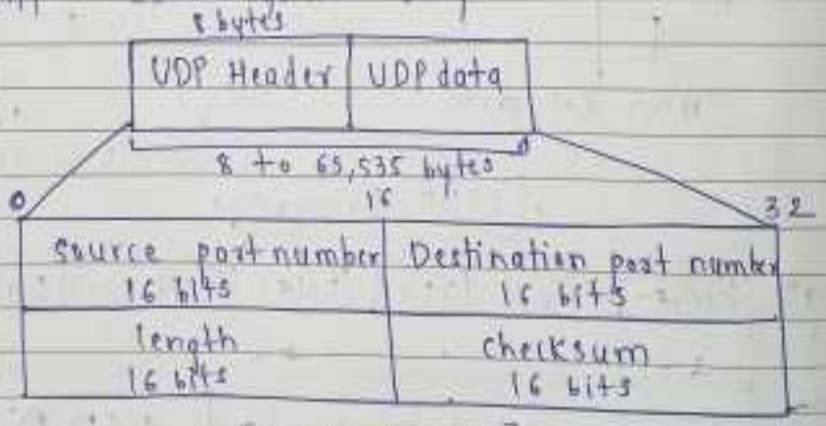


• Pos^n of UDP in the TCP/IP protocol Suite •

" UDP packets, called user Datagrams

User Datagram & services -

- UDP header is an 8-bytes fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes.
- The first 8 Bytes, contains all neccessary header information and the remaining part consist of data.
- UDP port number fields are each 16 bits long, therefore the range for port number is defined from 0 to 65535
- port number 0 is reserved.
- Port number helps to distinguish diff. user requests or processes.



[Header format]

Services-

1. Source port -

Source port is a 2 byte long field used to identify the port number of the source.

2. Destination port -
It is a 2 byte long field, used to identify the port of the destined packet

3. length -
length is the length of UDP including the header and the data.
It is 16-bit field.

4. Checksum -
- checksum is 2 bytes long field.
- It is 16-bit one's complement of the one's complement sum of the UDP header.

Applications of UDP -

① Following implementations uses UDP as a transport layer protocol:
- NTP (Network time protocol)
- DNS (Domain Name service)
- BOOTP, DHCP
- NNP (Network News protocol)
- Quote of the day protocol
- TFTP, RTSP, RIP.

② Task done through UDP by applica^n layer
- Trace route
- Record route
- Timestamp

3. Used for Simple request - response communication when the size of data is less and hence there is lesser concern about flow and error control.

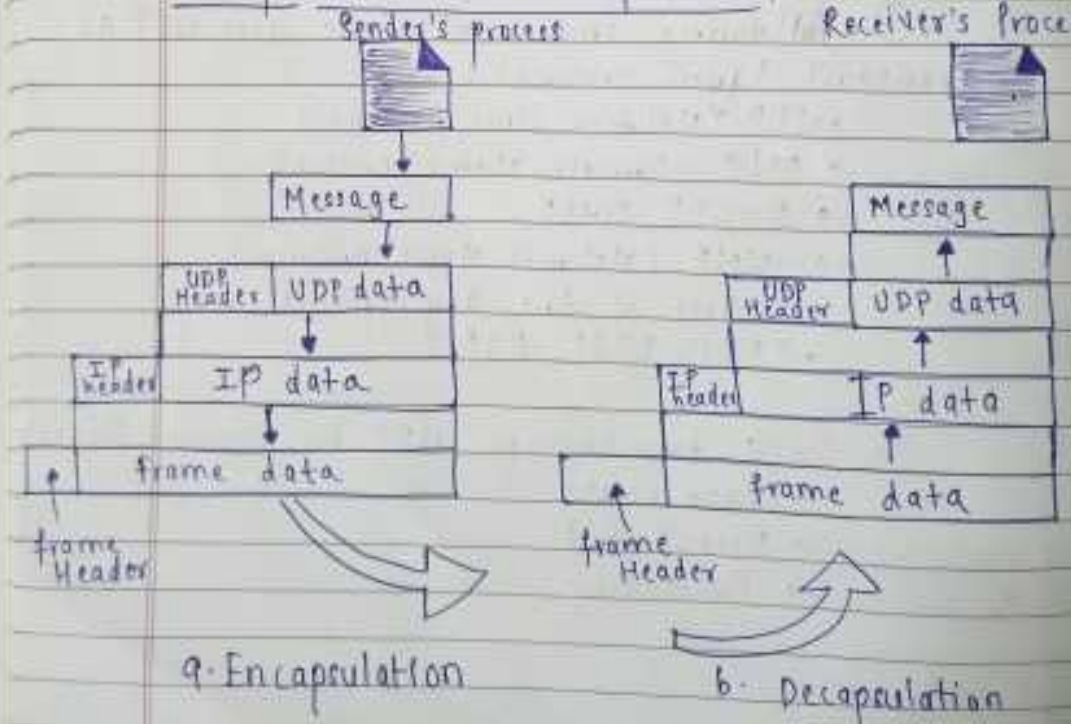
4. UDP is used for some routing update protocols like RIP (Routing information protocol)

5. UDP is suitable protocol for multicasting as it supports packet switching.

6. UDP takes a datagram from Network layer, attaches its header, and sends it to the user.

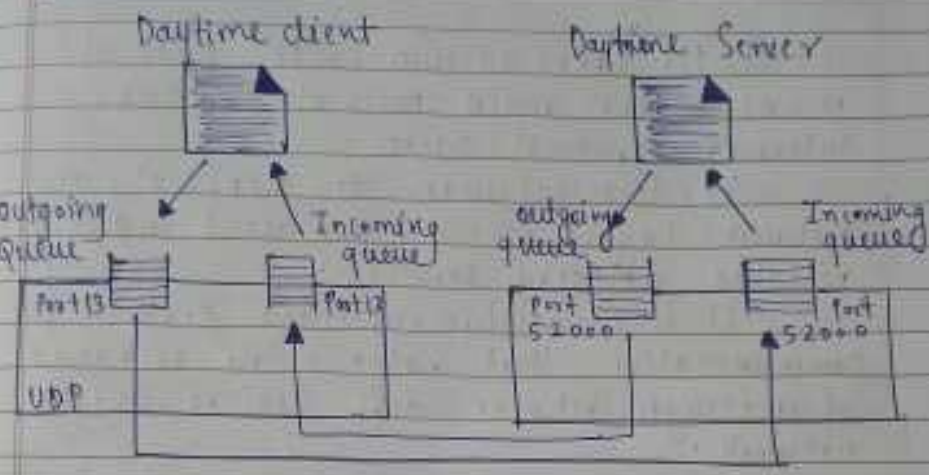
7. UDP is null protocol if you remove the checksum field.

- Encapsulation and Decapsulation -

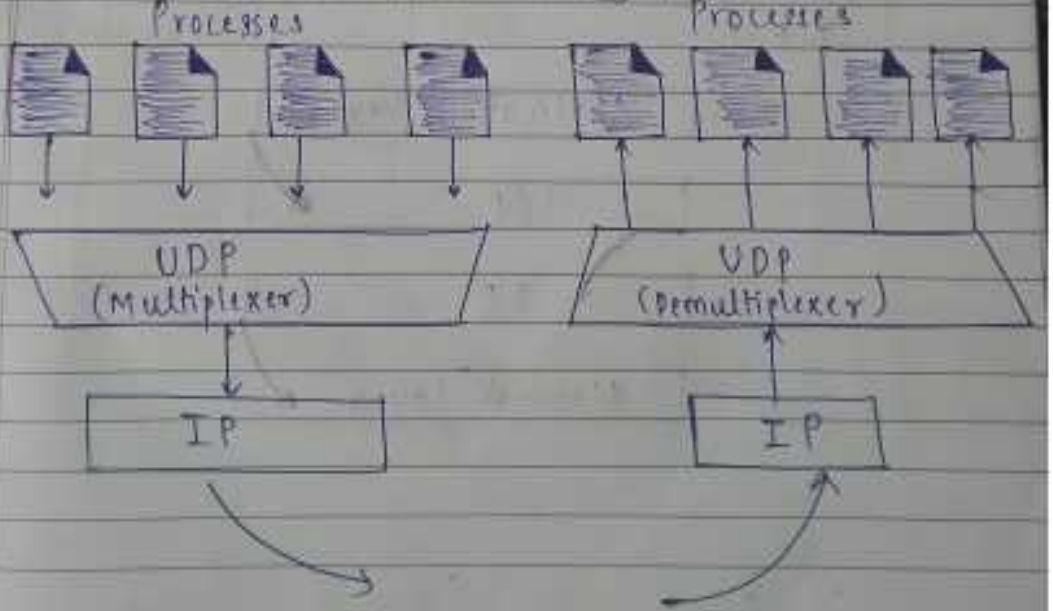


TCP- services, segment, connection, state transition diagram, flow control, congestion control, errors control, timers.

- Queues in UDP

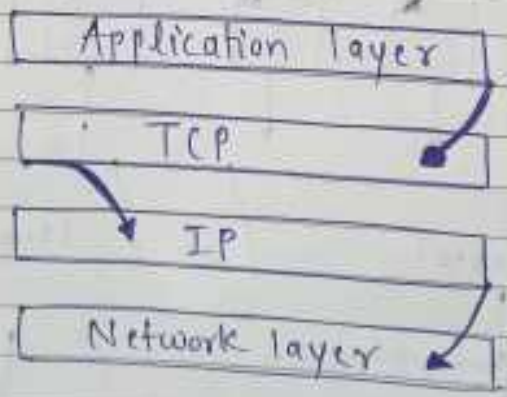


- Multiplexing and Demultiplexing -



TCP -

- TCP (Transmission control protocol) is one of the main protocols of the internet protocol suite.
- It lies between the Applicaⁿ and Network layers which are used in providing reliable delivery services.
- It is connection-oriented protocol for communication that helps in the exchange of message between diff. devices over a network.
- The Internet protocol (IP), which establishes the technique for sending data packets between computers, works with TCP.

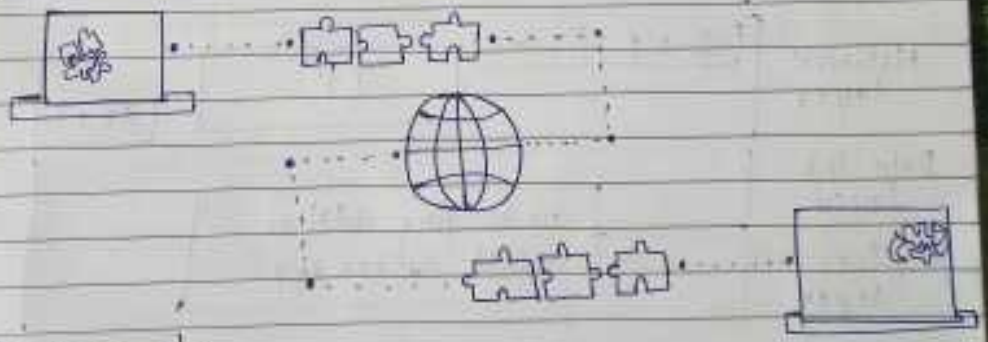


Working of TCP :

- To make sure that each message reaches it's target location intact, the TCP/IP model breaks down the data into small bundles and afterward reassembles the bundles into the original message on the opposite end.

- Sending the info. in little bundles of info. makes it simpler to maintain efficiency as opposed to sending everything in one go.

- After a particular message is broken down into bundles, these bundles may travel along multiple routes if one route is jammed but the destination remains the same



For Example,

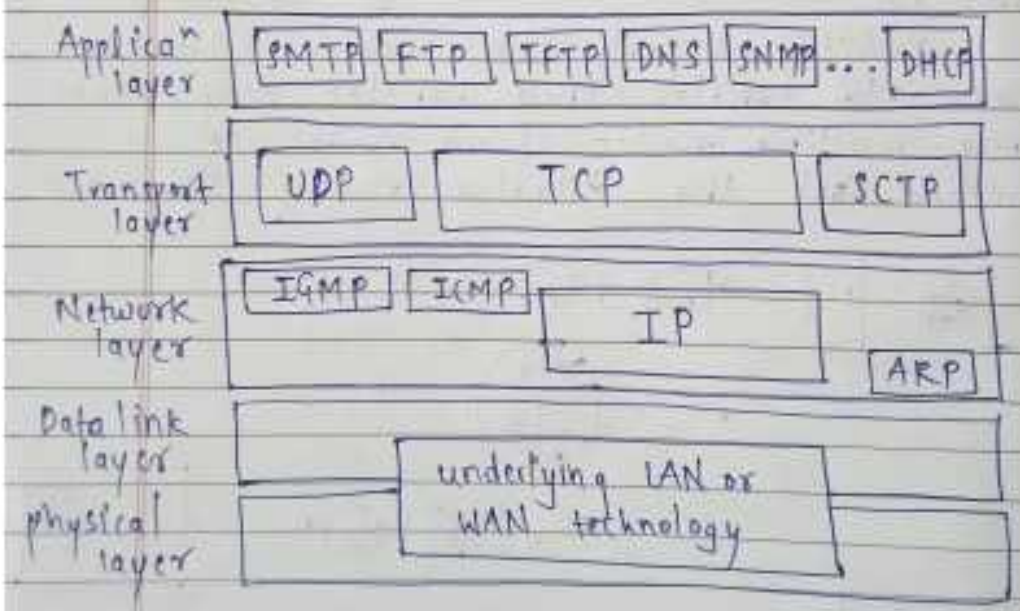
- When a user request a web page on the internet, somewhere in the world, the server processes that request and sends back an HTML page to that user.
- The server makes use of a protocol called the HTTP protocol.
- The HTTP then request the TCP layer to set the required connection and sent HTML file.

• Now, the TCP breaks the data into small packets and forwards it toward the (IP) layer.

• The packets are then sent to the destination through diff routes

• The TCP layer in the user's system waits for the transmission to get finished and acknowledges once all packets have been received.

TCP/IP Features



TCP features-

① Numbering system-

- TCP keeps the track of segments being transmitted or received.
- There is no field for a segment number value, in the segment header.
- There are two fields called the sequence number and the acknowledgement number.
- These two refers to byte number not a segment number.

② Byte Number-

TCP number all data bytes (octets) that are transmitted in a connection.

TCP chooses an arbitary number between 0 and 2^{32}.

③ Sequence Number-

After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent.

The sequence number for each segment is the number of the first byte of data carried in that segment.

① Acknowledgement Number -

• The sequence number in each direⁿ shows the number of first byte carried by the segment.

• Each party also uses acknowledgement number ~~defines the number of the next byte~~ to confirm the bytes it has received.

• However, acknowledgement number defines the no. of next byte that the party expects to receive.

> The value of the ack. field in a segment defines the number of the next byte a party expects to receive.
> The acknowledgement number is cummulative.

② Error control :

To provide reliable service, TCP implements an error control mechanism.

Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented. ~~as lot of~~

③ Flow control -

- TCP, unlike UDP, provides flow control.
- The sending TCP controls how much data can be accepted from the sending process; the receiving TCP controls how much data can to be sent by sending TCP.

④ Congestion control -

TCP, unlike UDP takes into account congestion in the network.

The amount of data sent by a sender is not only controlled by the receiver, but is also determined by the level of congestion if any in the network.

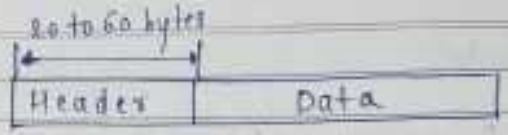
(*) SEGMENT -

A packet in TCP is called segment.

Format :

The format of a segment is as follows:-

1. The segment consist of a header of 20 to 60 bytes, followed by data from the applica^ program.
2. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.



a. Segment

b. Header

1. Source port Address-
This is 16-bit field number that defines the port no. of the applica^ program in the host that is sending the segment
Serves same purpose as the source port address in the UDP header.

2. Destination port Address-
This is 16-bit field that contains defines the port number of the applica^ program in the host that is receiving the segment.
Serves same purpose as the destina^ port address in the UDP header.

3. Sequence Number

The 32 bit field defines the number assigned to the first byte of data contained in this segment.

- The seq. no. tells the destinaⁿ which byte in this sequence is the first byte in this segment.
- During connection establishment each party uses a random number generator to create an initial sequence number (ISN), which is usually diff. in each direcⁿ.

4. Acknowledgement Number -

This 32-bit field defines the byte number that the receiver segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number 'x' from the other party it returns 'x+1' as the acknowledgement number.

Acknowledgement and data can be piggybacked together.

5. Header length -

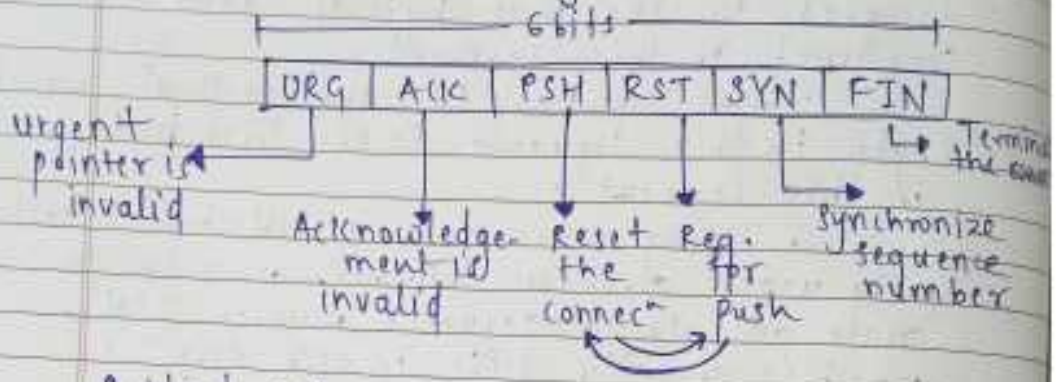
This 4bit indicates the number of 4-byte words in the TCP Header.

length - 20 to 60 bytes

6. Reserved -

6-bits reserved for future use.

7. control -
This field defines 6 diff. control bits or flags.



8. Window size -

- The field defines the window size of the sending TCP in bytes.
- Max. size - 65,535 bytes.

9. checksum -

- This is 16-bit field contains the checksum.
- Use of checksum in UDP is optional
- TCP is mandatory

10. urgent Pointer -

- 16-bit field, which is valid only when urgent flag is set is used when the segment contains urgent data.

11. options -

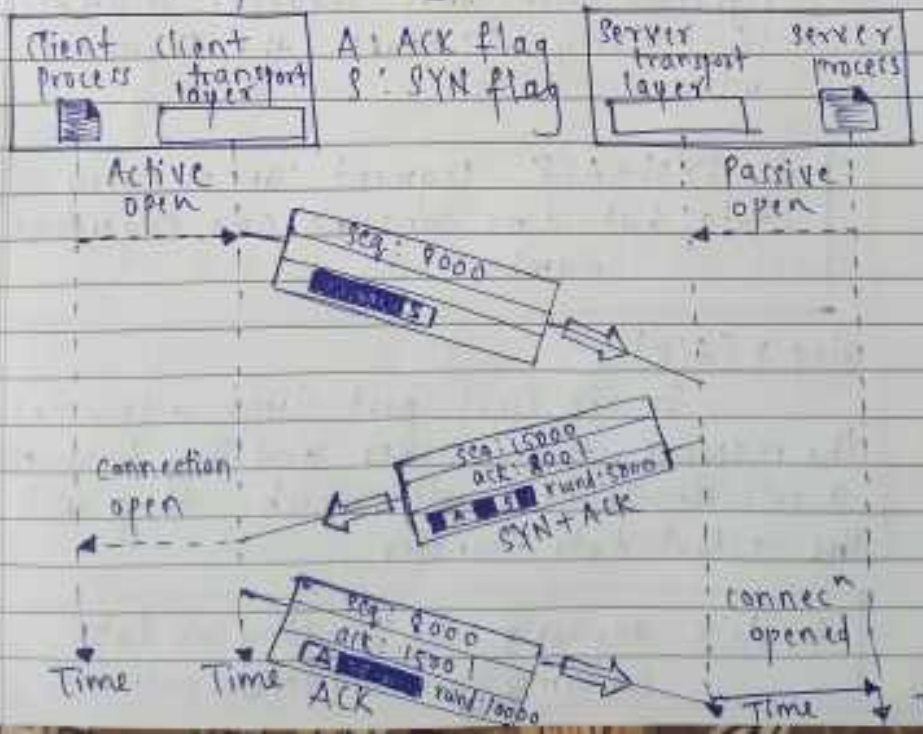
- There can be 40 bytes of optional info. in the TCP header.

A TCP CONNECTION -

1. connection Establishment
2. Data Transfer
3. Connection Termina^n
4. connection Reset

1. connection Establishment -

- TCP transmits data in ful-duplex mode.
- When two TCPs in two machines are connected they are able to send segments to each other simultaneously.
- This implies that each party must initialize communica^n and get approval from the other party before any data are transferred.



Three way Hand-shaking -

step 1 (SYN) :

In the first step, the client wants to establish a connection with a server. So it sends a segment with SYN (Synchronize sequence Number) which informs the server that the client is likely to start communication and with what sequence number it starts segment with.

> A SYN segment cannot carry data, but it consumes one seq. number

Step 2 (SYN + ACK) :

Server responds to the client request with SYN-ACK signal bits set. Ack. (ACK) signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with.

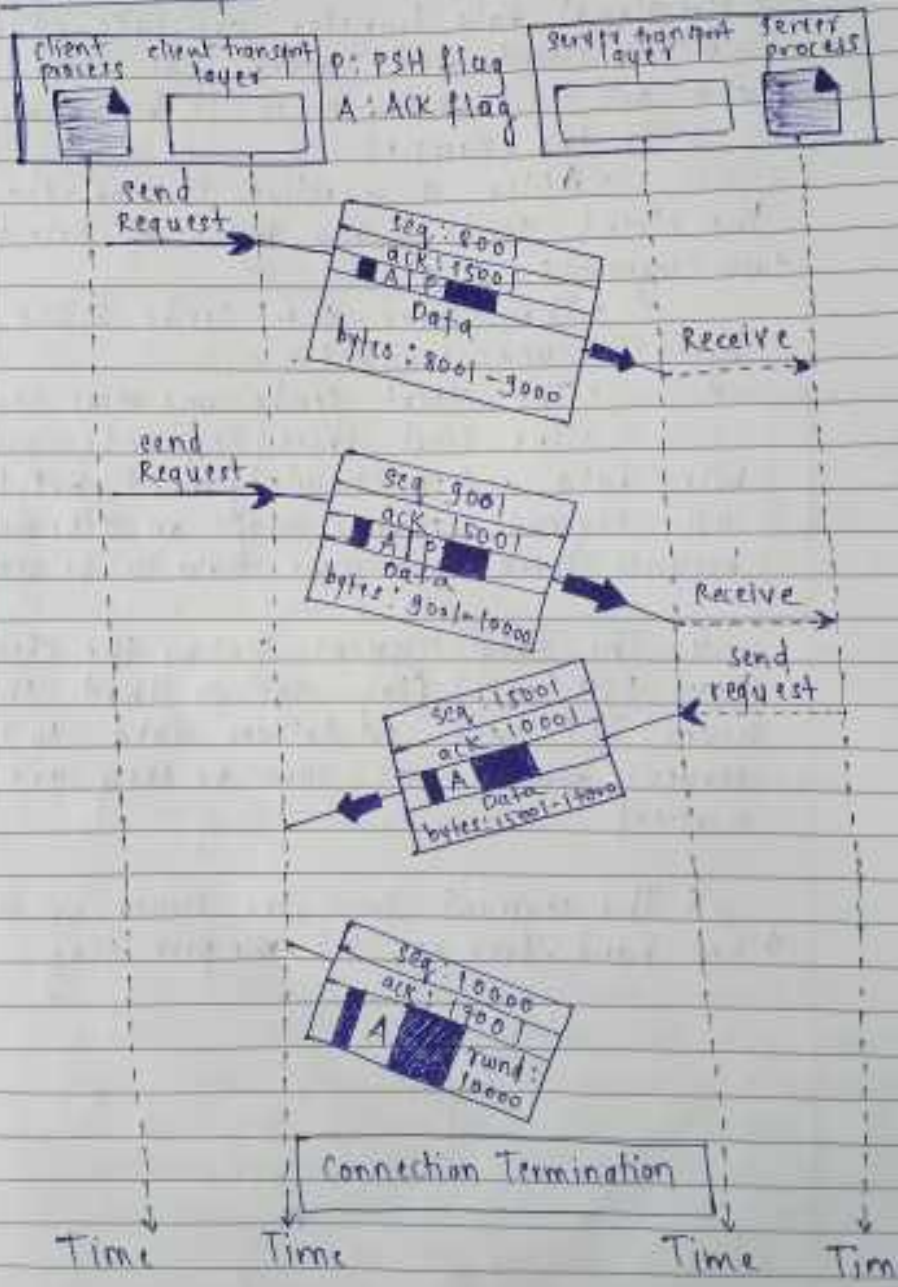
> A SYN+ACK segment cannot carry data, but does consume one sequence number

step 3 (ACK) :

In the final part client acknowledges the response of the server and they both establish a reliable connection with which they will start the actual data transfer.

> An ACK segment, if carrying no data, consumes no sequence number

2. Data transfer :



1. After connection is established, bidirectional data transfer can take place
2. The client and server can send data and acknowledgements in both directions
3. In the example,
- After a connection is established the client sends 2,000 bytes of data in two segments.
- The server then sends 2,000 bytes in one segment.
- The client sends one more segment
- The first three segments carry both data and acknowledgement, but the last segment carries only an acknowledgement because there is no more data to be sent

4. The data segment sent by the client have PSH (push) flag set so that the server TCP tries to deliver data to the server process as soon as they are received

5. The segment from the server, on the other hand, does not set the push flag.

3. Connection Termination -

- Any two parties involved in exchanging data can close the connection, although it is usually initiated by the client.

- can be done by using -

• Three-way handshaking:-

Step 1 :-

1. In a common situation, the client TCP, after receiving a close command from the client process sends the first segment, a FIN segment in which FIN flag is set.
2. A FIN segment can include the last chunk of data sent by the client or it can be just a control segment.
3. If it is only a control segment, it consumes only one sequence number.

* The FIN segment consume one seq. number if it does not carry data

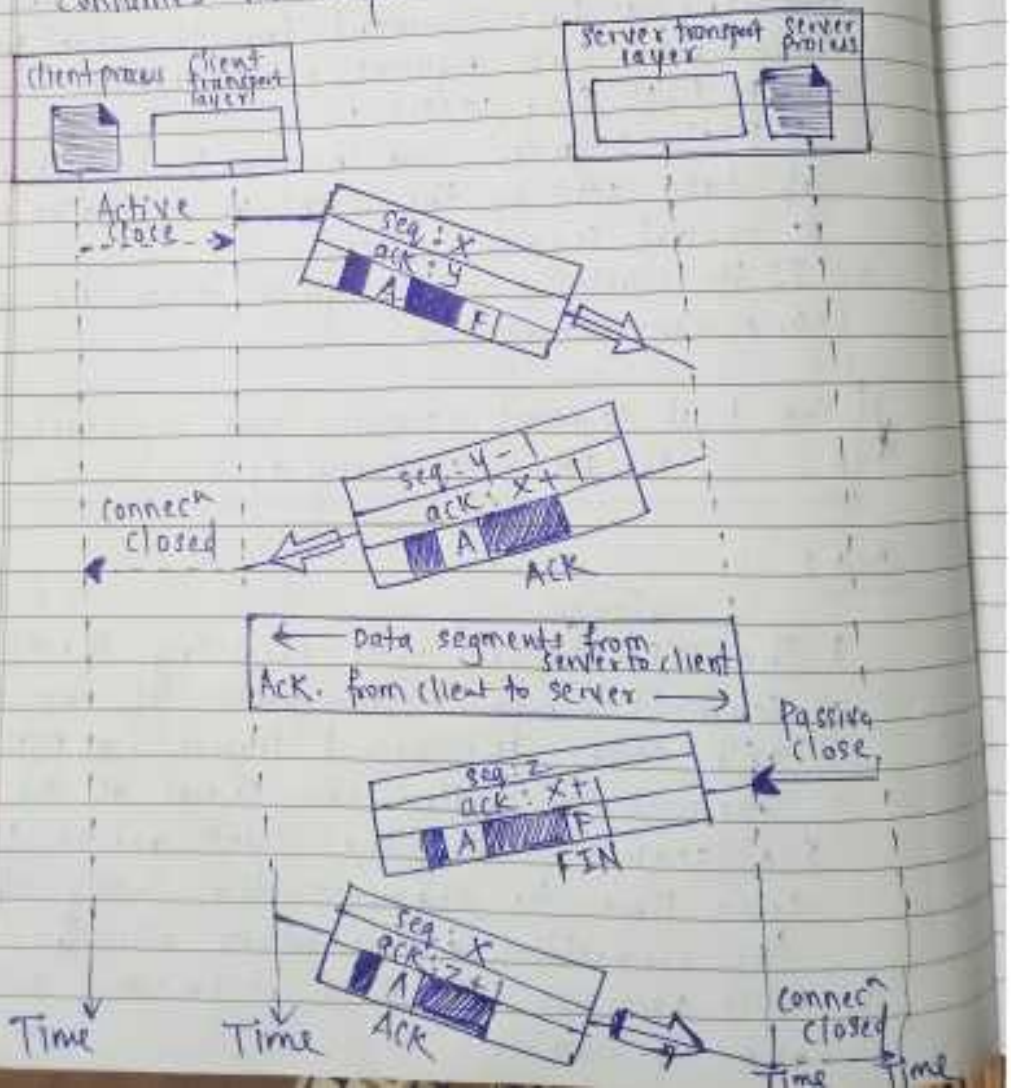
Step 2 :

1- The TCP server, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN+ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction.
2. If it does not carry data, it consumes only one seq. number.

Step 3:

The client TCP sends the last segment an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgement number, which is one plus the sequence number received in the FIN segment from the server.

This segment cannot carry data and consumes no sequence numbers.



4. Connection Reset –

- TCP at one end may deny a connec^n request, may abort an existing connec^n, or may terminate an idle connec^n.
- Done by RST (reset) flag.

① Denying a Connection –

~~Suppose the TCP on one side has requested, may abort an existing connec^n, or may terminate an idle connection.~~

- Suppose the TCP on one side has requested a connection to a nonexistent port
- The TCP on the other side may send a segment with its RST bit set to deny the request

② Aborting a connection –

- one TCP may want to abort an existing connec^n due to an abnormal situation.
- It can send an RST segment to close the connection.

③ Terminating an Idle Connection –

- The TCP on one side may discover that the TCP on the other side has been idle for a long time.
- It may send an RST segment to end the connec^n.
- The process is the same as aborting connec^n.

(*) State transition Diagram-

The fig. shows -

1. The two FSMs used by the TCP client and server combined in one diagram
2. The ovals represent the states
3. The transi^n from one state to another is shown using directed lines.
4. Each line has two strings operated by a slash.

5. The first string is the input, what TCP receives.

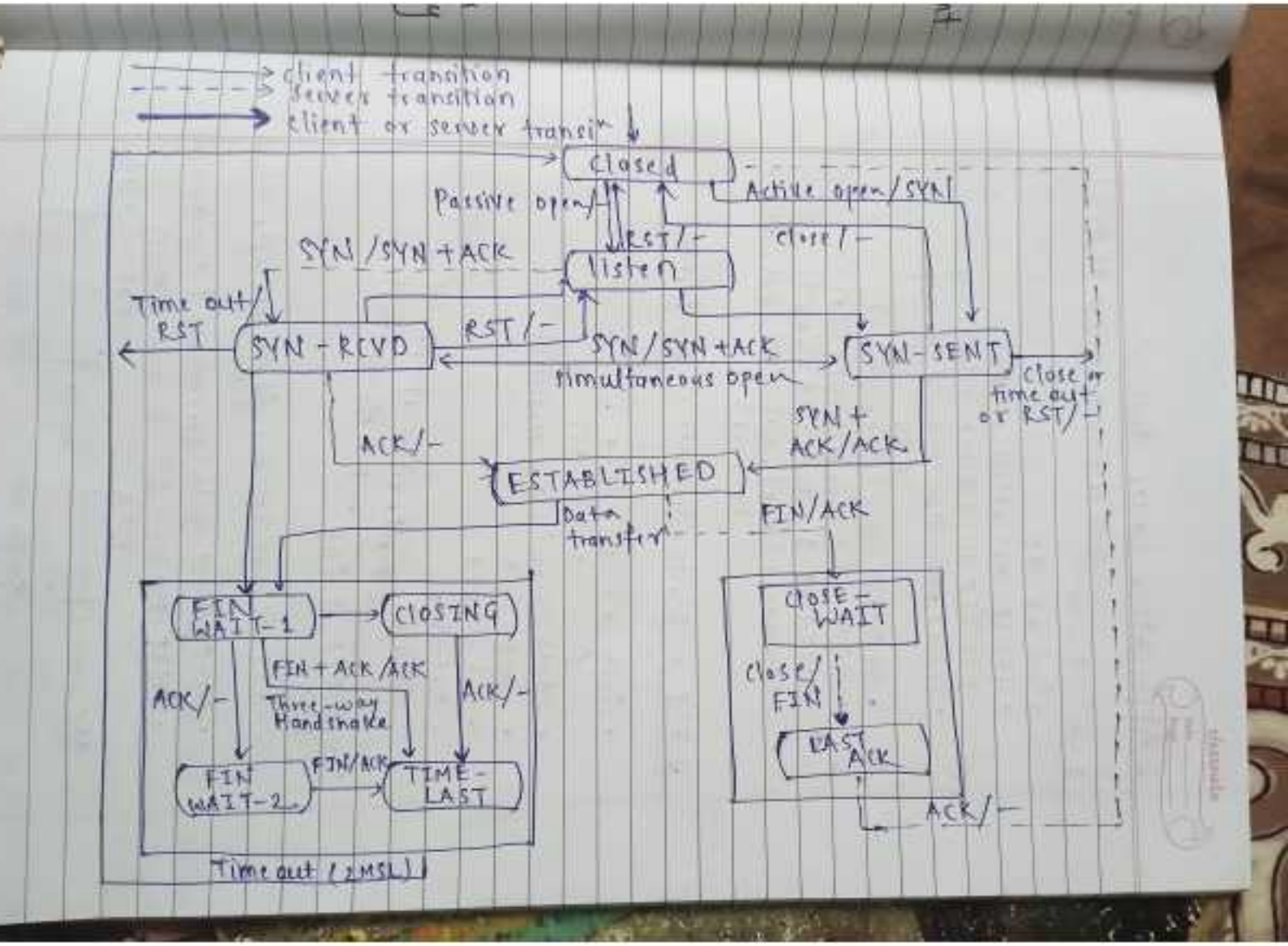
6. The second is the output, what TCP sends.

7. The dotted black lines in the fig. represent the transi^n that a server normally goes through; the solid black lines show the transitions that a client normally goes through.

8. However, in some situation, a server transition through a solid line or a client transitions through a dotted line

The state marked as ESTABLISHED in the FSM is in fact two diff. sets of states that the client and server undergo to transfer data.

client transition


client transition
Server transition
client or server transition
CLOSED
Passive open
Active open/SYN
SYN/SYN+ACK
RST/-
close/-
LISTEN
Time out/
RST
SYN-RCVD
RST/-
SYN/SYN+ACK
SYN-SENT
simultaneous open
close or time out or RST/-
SYN+ ACK/ACK
ACK/-
ESTABLISHED
Data transfer
FIN/ACK
FIN WAIT-1
CLOSING
FIN+ACK/ACK
ACK/-
Three-way Handshake
ACK/-
FIN WAIT-2
FIN/ACK
TIME-LAST
Time out (2MSL)
CLOSE-WAIT
close/ FIN
LAST ACK
ACK/-