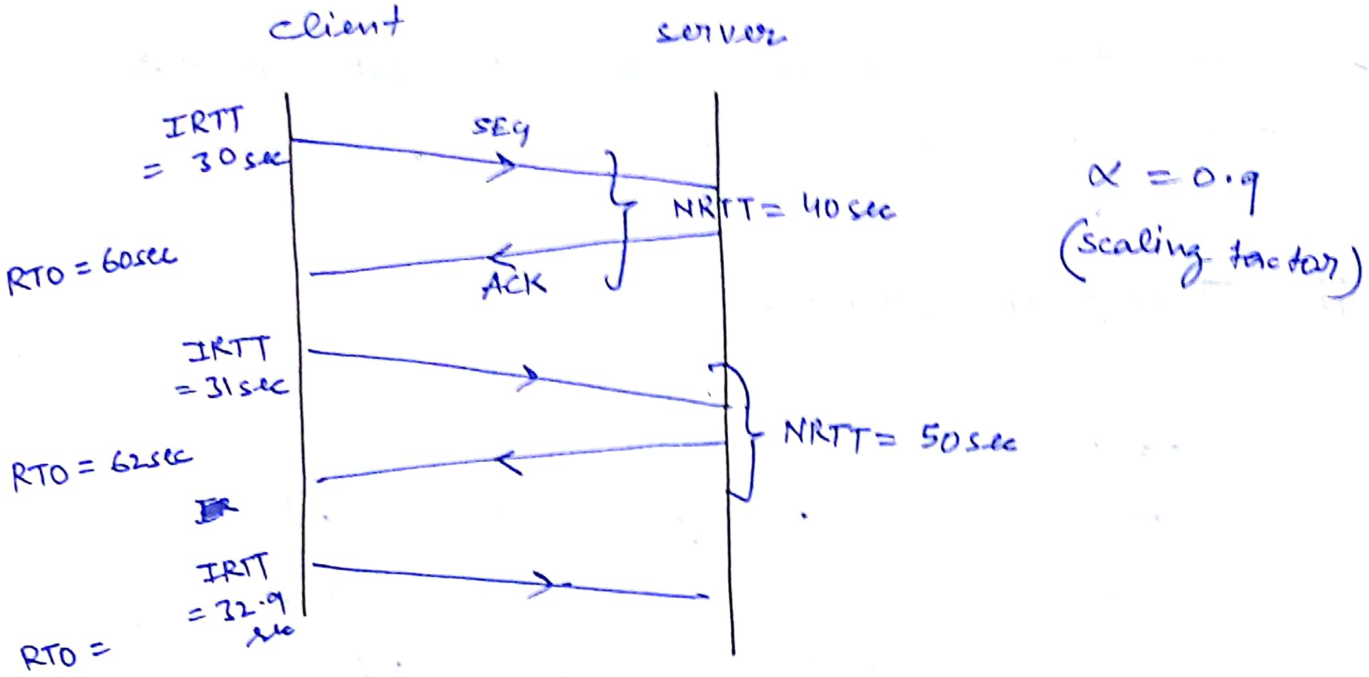


so, $ERTT_1 = \alpha\, IRTT + (1-\alpha)\, NRTT$

$= (0.9)(30) + (0.1)(40)$

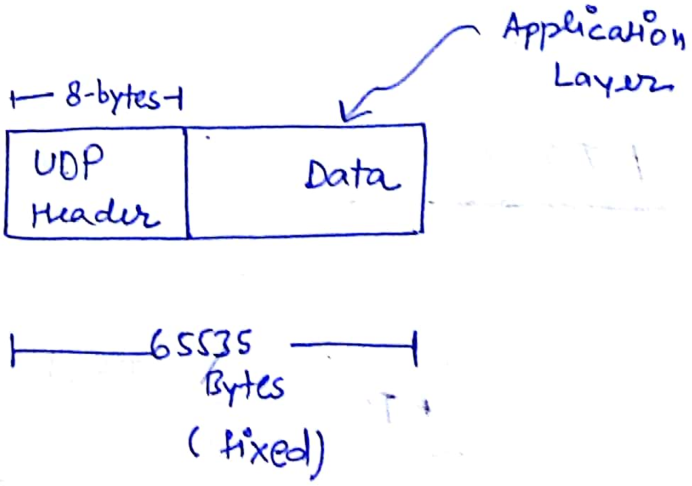
$= 31$ sec

$ERTT_2 = (0.9)(31) + (0.1)(50)$

$= 32.9$

UDP protocol:-

Datagram =



UDP header

	16-bits	16-bits
Row1	Source port	Destination port
Row2	Checksum	Total length

* Since Header length is fixed i.e 8-bytes.

Q. Total Length bits = 0000 0000 1111 1111, then calculate size of the datagram, and payload.

Size of datagram = $2^8 - 1$ = 255 Bytes

Payload size = 255 − 8 = 247 Bytes.

Q. UDP header is given as $(FFF000\ 50\ FFFE\ FFFE)_{16}$.

Calculate :

(i) Source port

(ii) Destination port

(iii) Size of Datagram

(iv) Size of Data or Payload

$$\underbrace{FFF0}_{16\text{ bits}}\overbrace{00\ 50}^{16\text{-bits}}\ \underbrace{FFFE}_{\text{checksum}}\ \overbrace{FFFE}^{\text{Total length}}$$

Source Port = $(FFF0)_{16}$ = $\underline{65520}$ (dynamic port)

Destination Port = $(0050)_{16}$ = $\underline{80}$ (http port)

Size

Total length = FFFE = 65534 ~~bytes~~

so, size of datagram = $\underline{65534 \text{ bytes}}$

Size of data = 65534 − 8

= $\underline{65526 \text{ bytes}}$

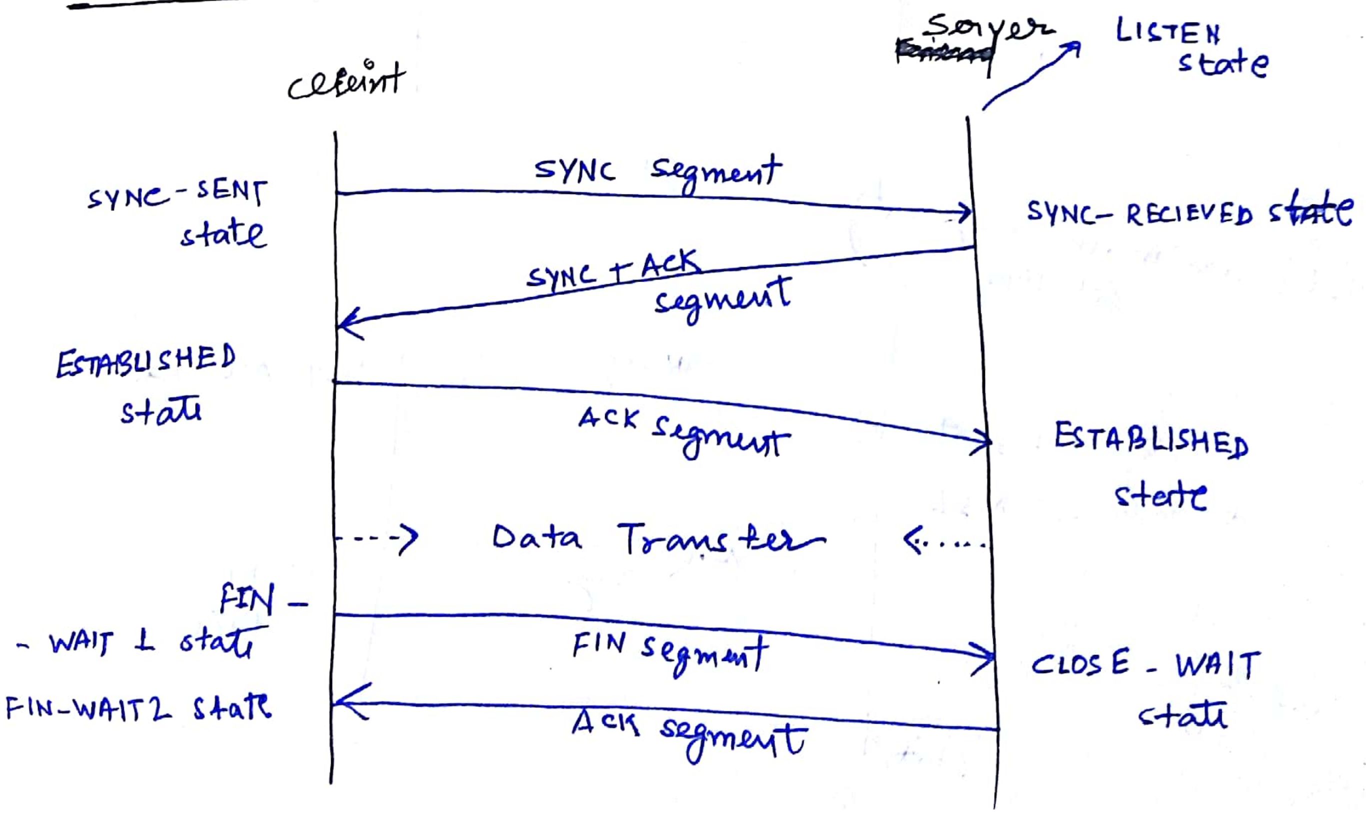
* This is client to server process.

TCP vs UDP :-

TCP	UDP
(i) Dynamic header (20–60) Bytes	(i) fixed header 8-bytes
(ii) It is connection oriented with the help of sequence number.	(ii) It is connectionless as no sequence number

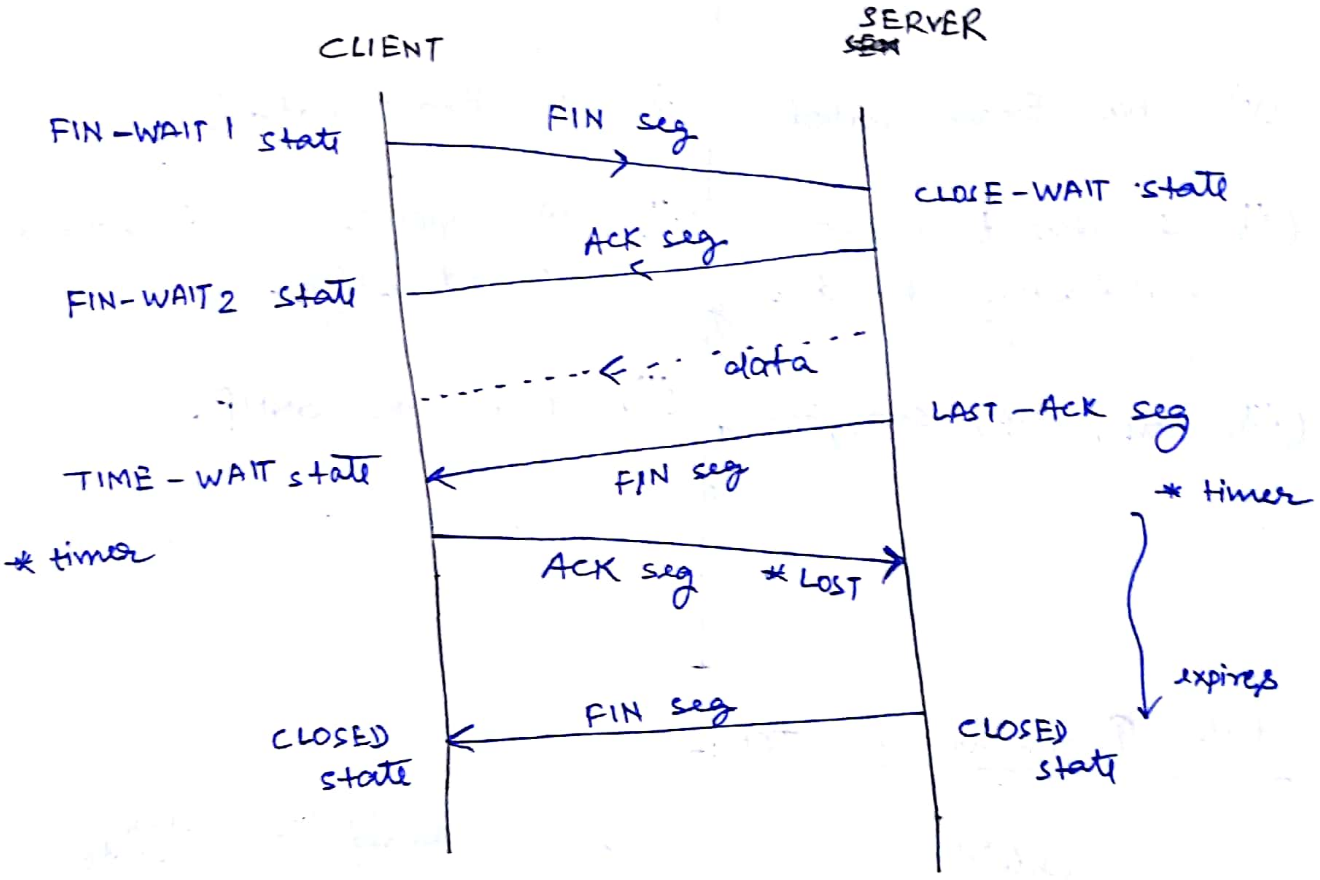
(iii) Flow control	(iii) no flow control (Each datagram is independent)
(iv) Slow	(iv) Fast
(v) Checksum is mandatory.	(v) Checksum is optional
(vi) has Error control	(vi) No Error control
(vii) It doesn't support multicasting & broadcasting	(vii) It supports multicasting & broadcasting.
(viii) http, ftp, SMTP, Telnet	(viii) TFTP, DNS, SNMP.

state Transition of TCP :-

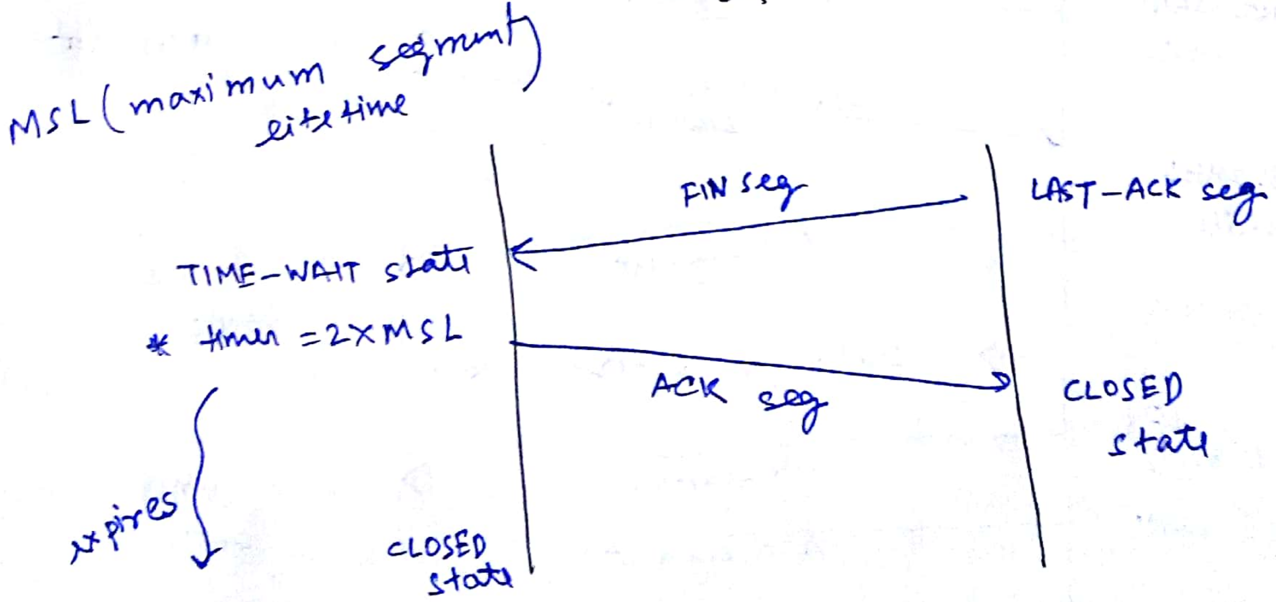


- client will move from SYNC-SENT state to ESTABLISHED state when it gets SYNC + ACK segment.

- Server will move from SYNC - RECIEVED state to ~~Estbsd~~ ESTABLISHED state when it gets ACK segment.



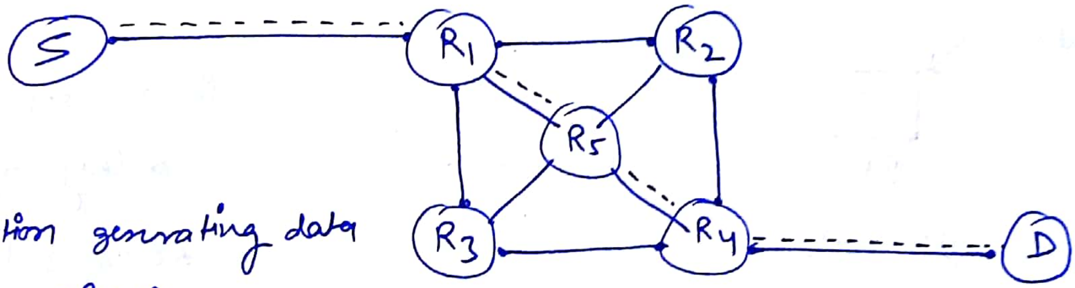
OR



- In both state FIN-WAIT 1 & FIN-WAIT 2 client will not send any data to server but it recieves data From server.

- Client will move from FIN-WAIT 1 to TIME-WAIT state when it gets FIN + ACK from server.

↳ when the application is generating data slowly and it is using TCP then the window size is small, this problem is known as silly window syndrome.



Application generating data slowly

↓

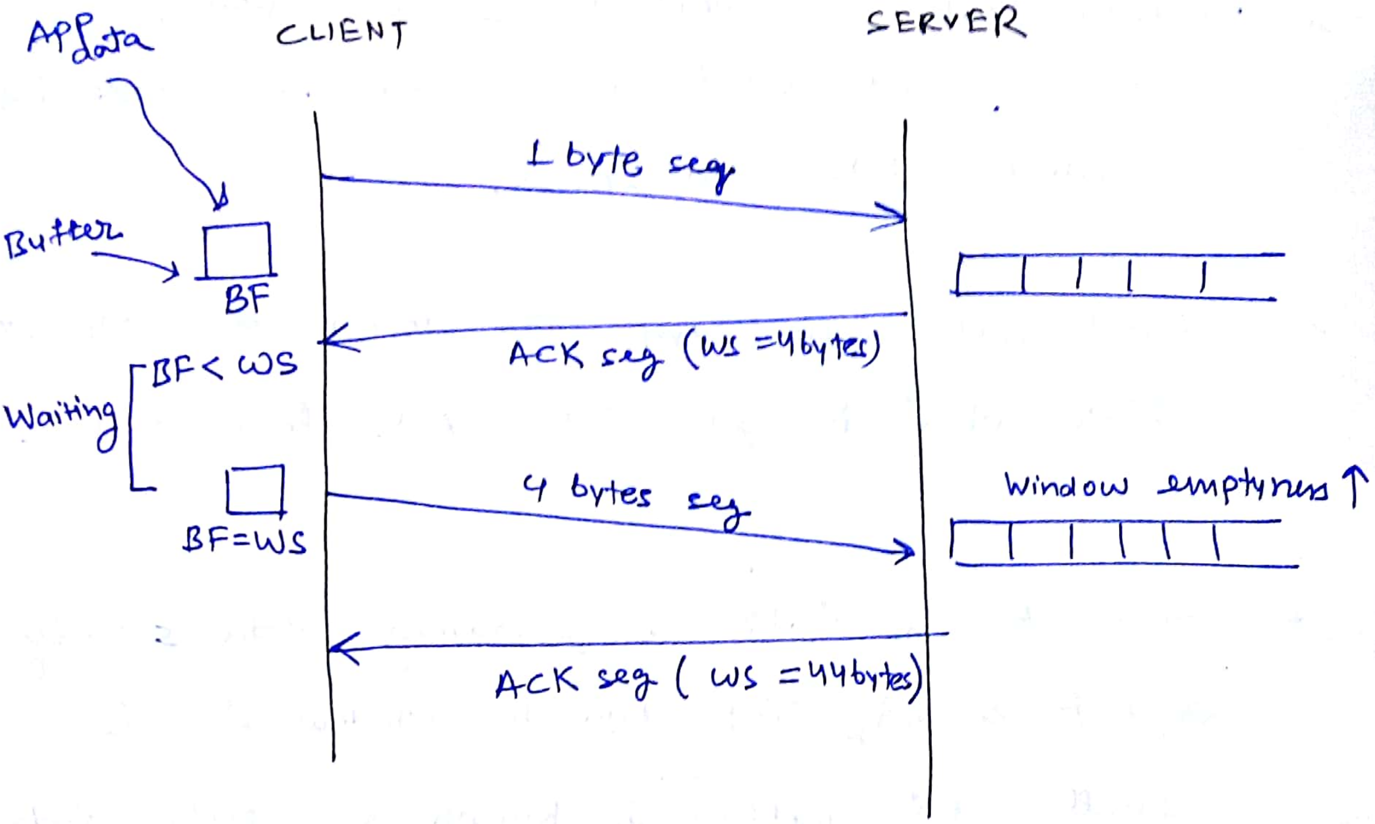
low channel utilization

↓

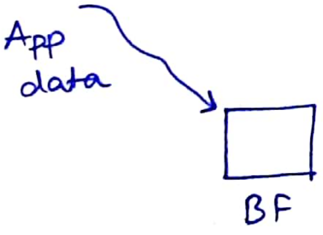
window size is small

↓

silly window syndrome



Naiglle Theory —extends→ Clarke Theory



BF < WS

when Buffer size is small than window size then delay the sending of data.

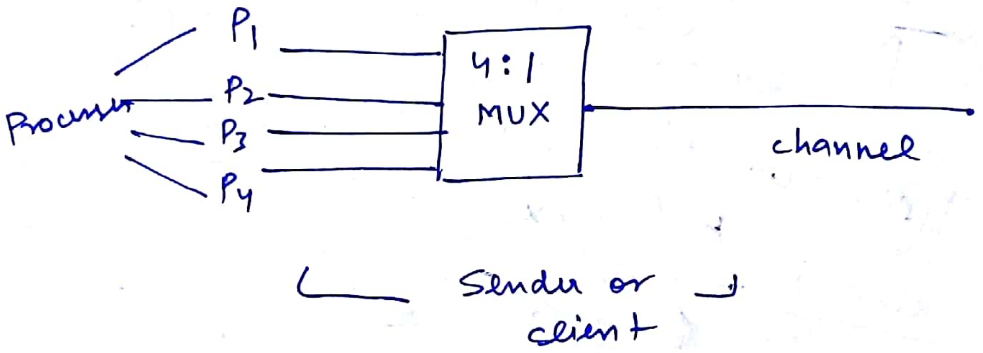
when (BF = WS)
{ send data }

insert some delay at server side for ACK ~~Sed~~. sending.

Thus both emptyness of window and Buffer size will increase

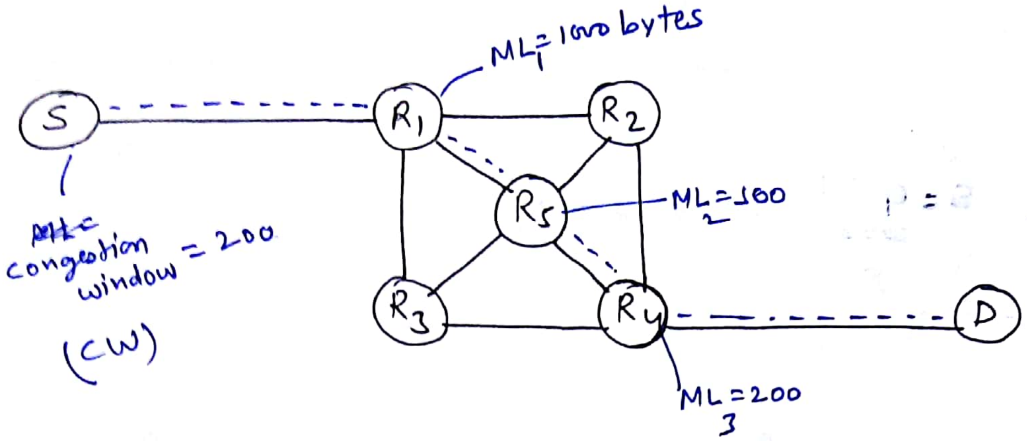
- Nagle suggested that whenever the ACK reached to the client ; compare the buffer size with the window size. When the buffer size is less than window size sender has to wait until the buffer size is equal to window size.

- During the waiting time the chance is given to other process to transmit the data.



- Clark suggested delay the ACK so that parallely buffer size and WS increases so that the problem of silly window syndrome will be fastely resolved.

Congestion policies of TCP :-



- Congestion window will be known to sender during connection establishment phase.

- Reciever window will be known to sender during data - transfer phase.

$$SW = (RW, cw)$$

If $RW << CW$

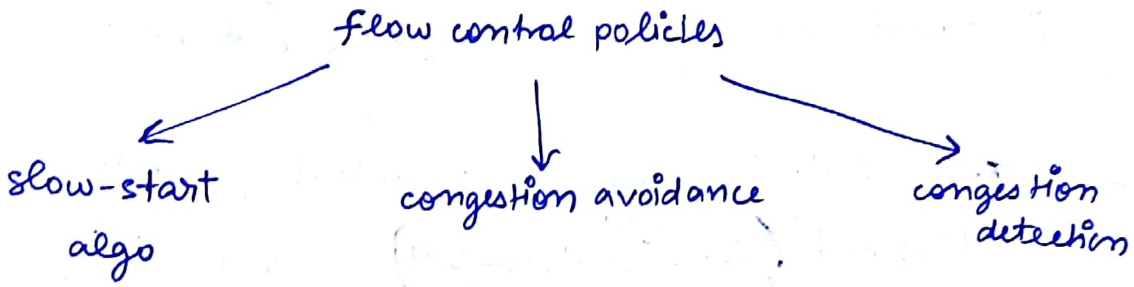
$(SW = RW)$

Else if $CW << RW$

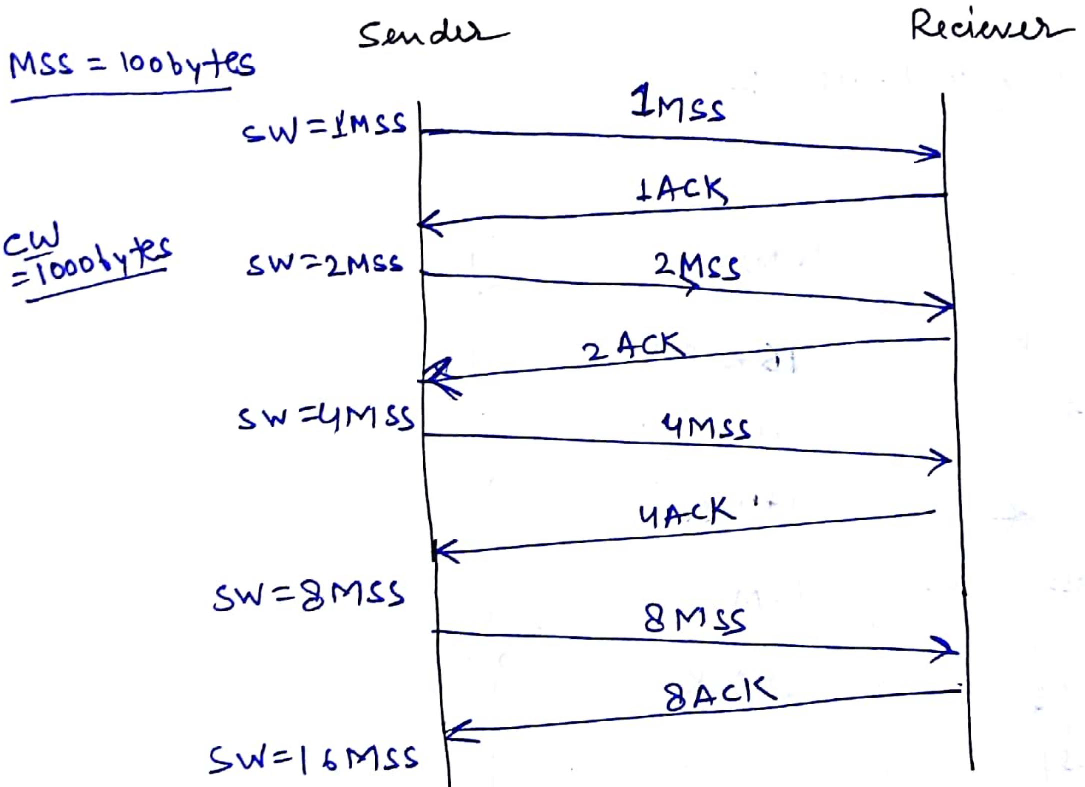
then $(SW = CW)$

so, $\boxed{SW = \min(RW, CW)}$

Case-1 : CW << RW



(i) slow - start algorithm :-



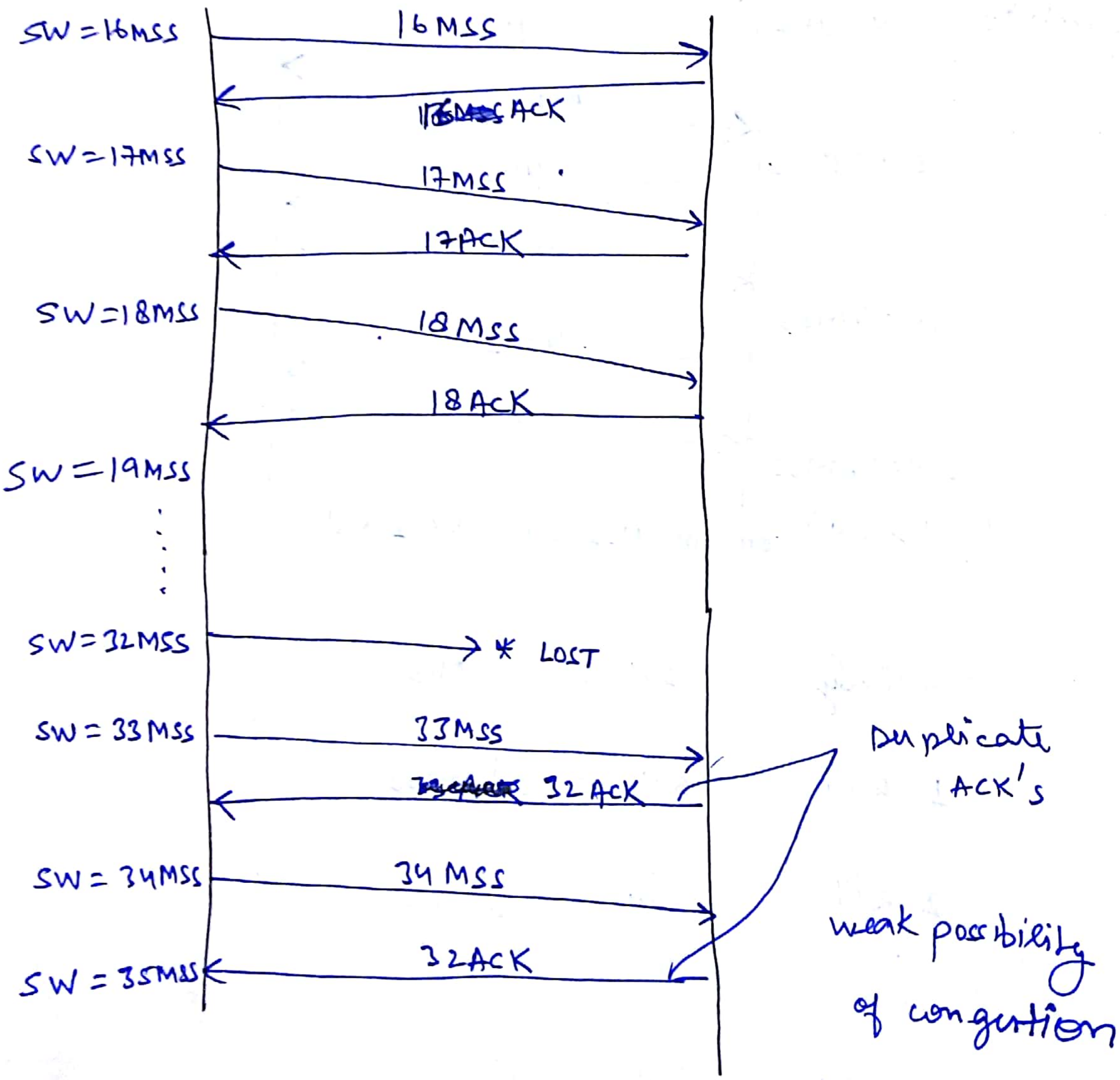
(SW > CW) so no MSS will be send

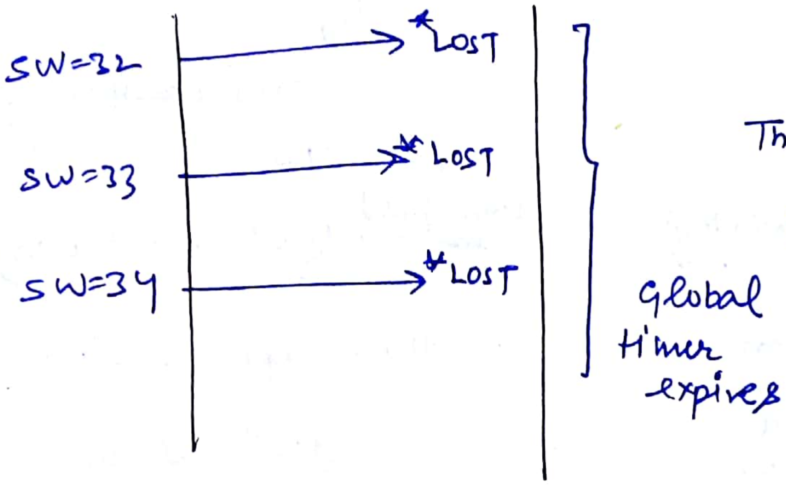
Initially	$SW = 2^0$ MSS
1 RTT	$SW = 2^1$ MSS
2 RTT	$SW = 2^2$ MSS
⋮	
n RTT	$SW = 2^n$ MSS

- In slow start algorithm, the increase of SW is based on number of previous ACK. ~~of previ~~

- ~~I.~~ SW increase exponentially upto slow-start threshold (~~the SW > CW~~ CW). After this TCP flow control policy shifts to collision avoidance

(ii) Collision Avoidance :-





This condition show more probability of congestion.

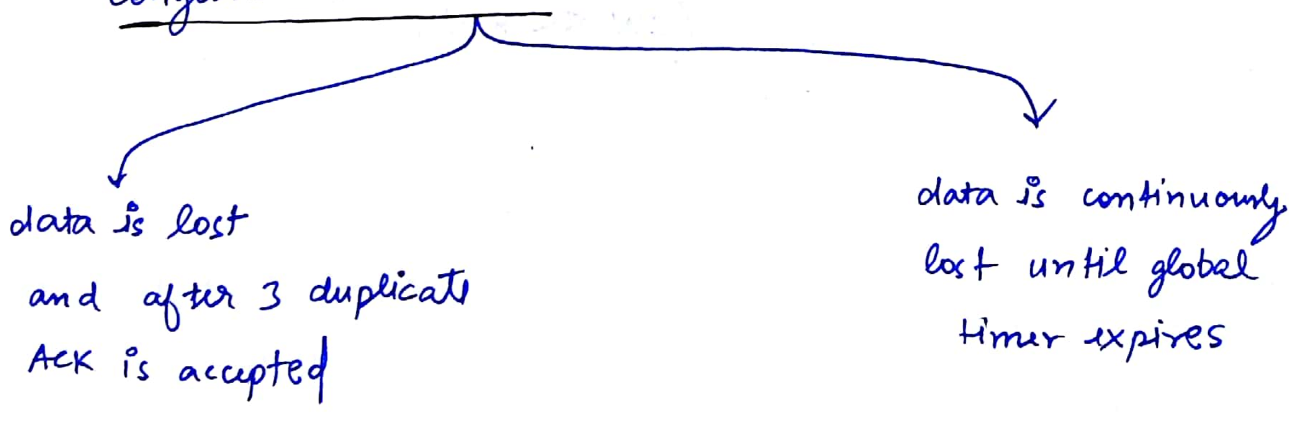
- In congestion avoidance algo , the increase of sender's window size is based upon RTT.

- SW increases linearly.

- Once the data is lost and if it is accepted after 3 duplicate ACK , it is treated as the weak possibility of congestion.

- If the data is lost continuously until the global timer expires then it is treated as the strong possibility of congestion.

(iii) congestion detection


data is lost and after 3 duplicate ACK is accepted
data is continuously lost until global timer expires

[Weak possibility]

SW = $\frac{1}{2}$ x (present window)

then apply congestion avoidance algo.

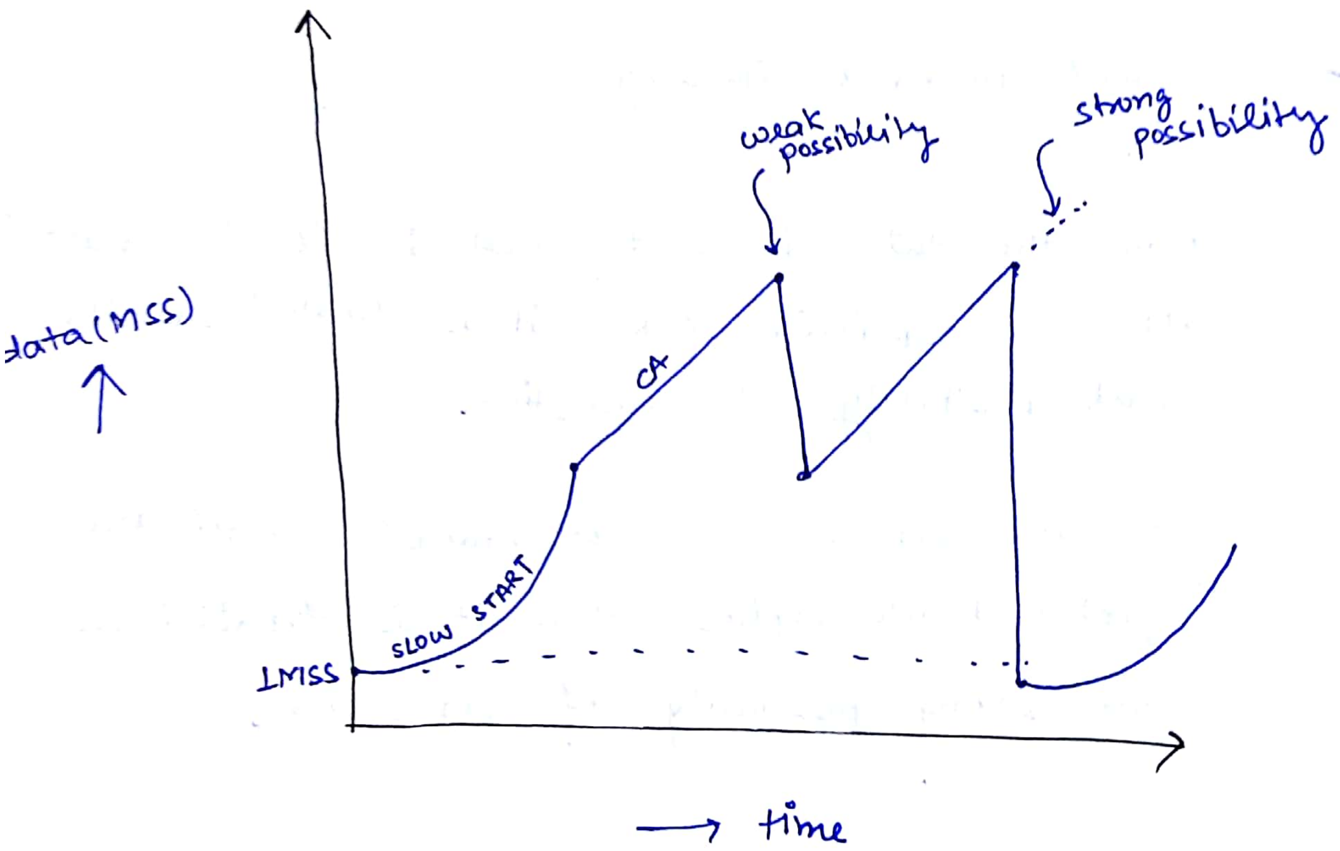
[strong possibility]

SW = 1MSS

threshold ~~SW~~ = $\frac{1}{2}$ x (present win)

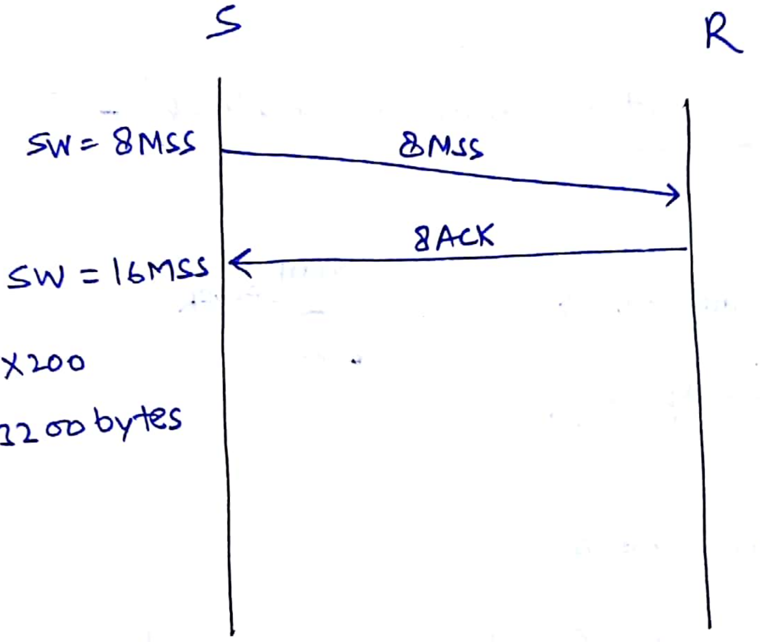
then apply slow start algo.

Example:-



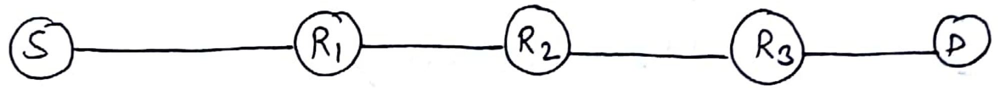
(CW << RW)

Q. Present sender window = 1600 bytes , MSS = 200 Bytes
Next sender window if slow start algo is applied.



= 16X200
= 3200 bytes

Q. A hacker is snooping at router R_2 , then what Information he can get



(i) IP address of source ✓

(ii) MAC address of source ✗

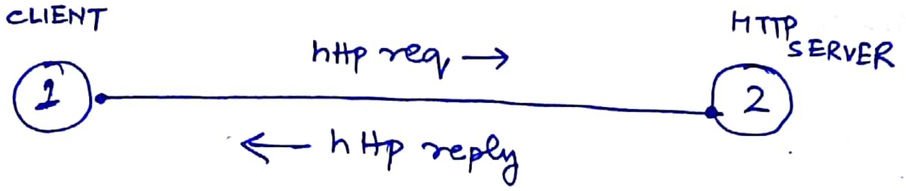
(iii) Port address of source ✓

- As data-link layer Header is eliminated at entry of router so hacker can't access MAC address of Source, but it can access MAC of R_2.

Application Layer :-

Protocols -

(i) **HTtp protocol (hypertext transfer protocol) -**



- client - server protocol

- synchronous protocol breause the clbck of the client is synchronized with the clock of the server.

- Default port : 80

- http connections

↙ persistent

↘ Non-persistent

persistent

- connection or session will not terminate & we can acess n no. of objects from server

- User-Friendly services

Non-persistent

- Connection or session will terminate as soon we access the content.

- Security services

- Http application programing Interfacu (API):

↳ http methods

- get()
- post()
- put()
- head()
- connect()
- trace()
- option()

(i) get() is used to retrive the document.

(ii) put() is used to modify the existing document.

(iii) post() is used to place the modify document back to directory in the server.

(iv) head() is used to get metadata of document.

(v) when connect() is used data will go via a secure channel that to be in encrypted form.

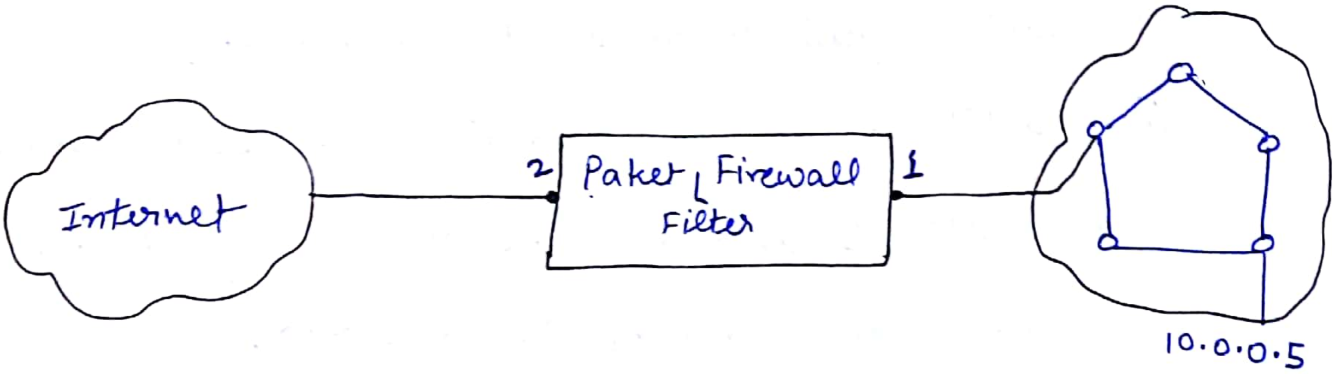
(vi) trace() is used to record route option

- http is stateless protocol because it doesn't store any information about the server in the client system, or vice versa.

- Cookie is piece of data which is saved on client machine for use in future.

- Advantage of cookies is
  - → Authorization
  - → Faster response.

Packet Filter Firewall :-



Interface	SIP	S.port	DIP	D.port
2	144.19.0.0	*	*	*
2	*	*	10.0.0.5	*
2	*	*	*	23
1	*	*	*	80

- It is a Firewall which blocks or forwards the data by observing transport & network layer headers of the content.

* There is no concept like Idle ffirewall, Every Firewall will work according to its design.

(i) Packet coming from a particular network i.e 144.19.0.0 are blocked.

(ii) Packets destinated to 10.0.0.5 are blocked i.e this system only used for internal LAN only.

(iii) Packets destinated to port 23 i.e. (Telnet) are blocked or we can say Telnet service is blocked.

(iv) Packets destinated to port 80 are blocked i.e http service is blocked.

Q. Why Antivirus software are needed ~~as~~ as fire-wall is already present.

If a virus is placed in the application data then the packet filter-wall can't detect it, thus anti-virus software is required.