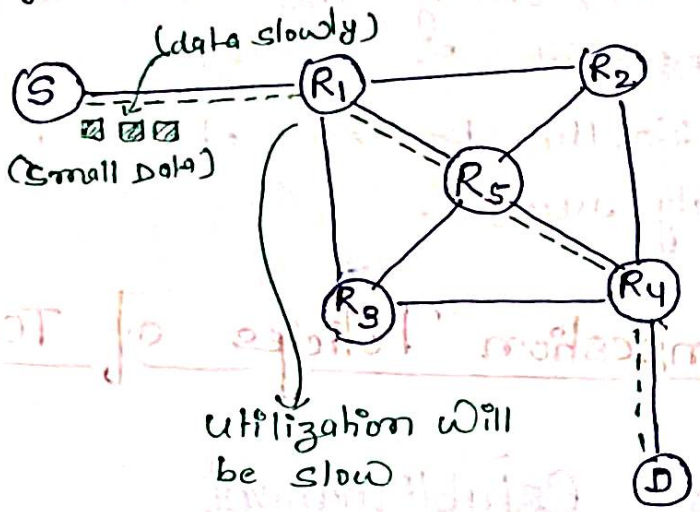
TCP
- → Maxm Utilization
- → Maxm throughput
- → All types of Applications

Slow application generating data slowly



Utilization will be slow

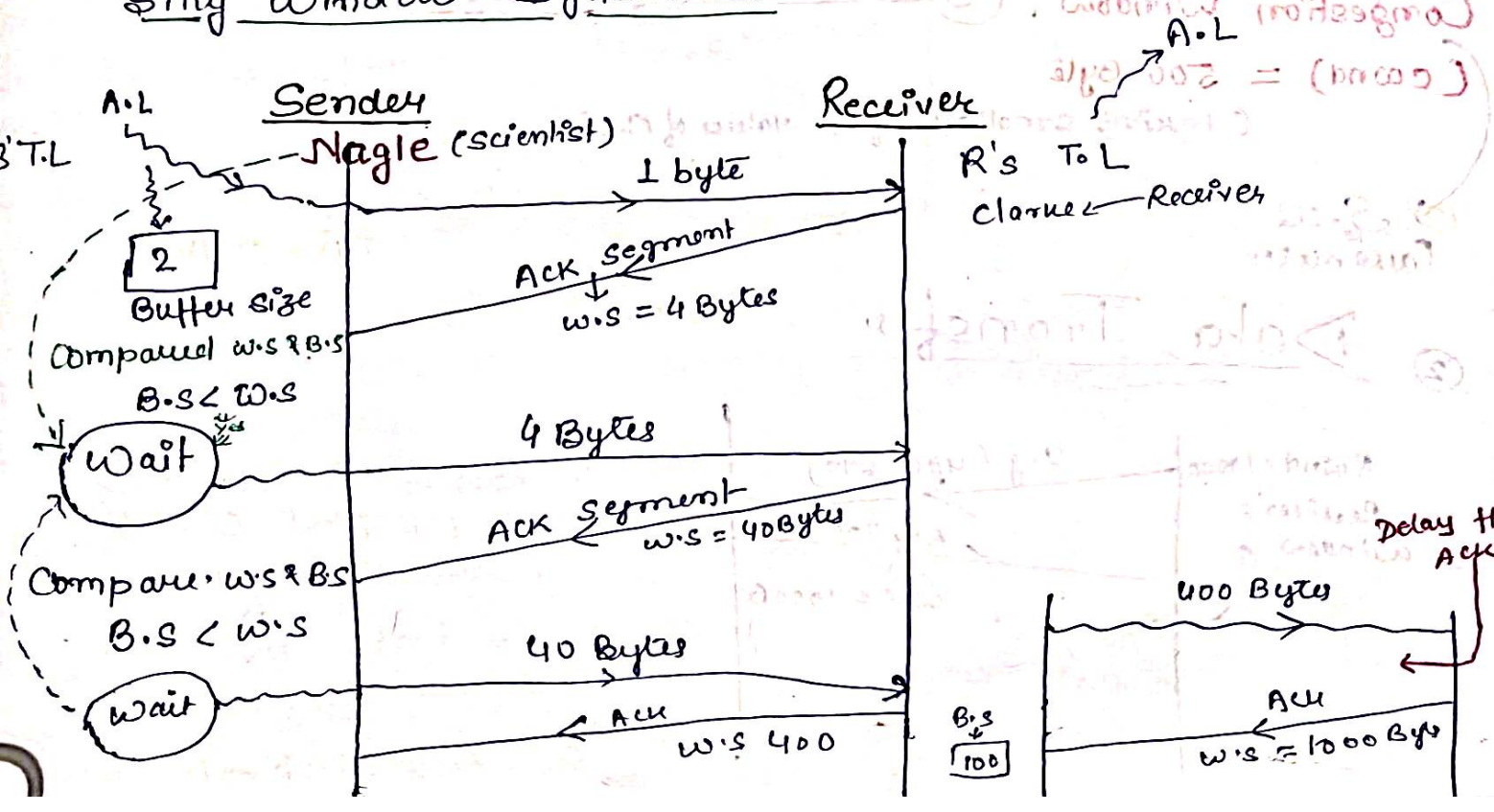
⇓

Silly Window Syndrome

↑ Small — ↑ Problem

[Bcz, in TCP ⇓ Connection-Oriented ⇓ Large Data Transfer ⇓ More utilization Should be there always]

→ If the application is generating data slowly then the utilization will be less, this problem is known as Silly Window Syndrome.



→ Whenever the Ack segment Reaches to client, Nagle suggested that compared the Buffer size with the Window size.

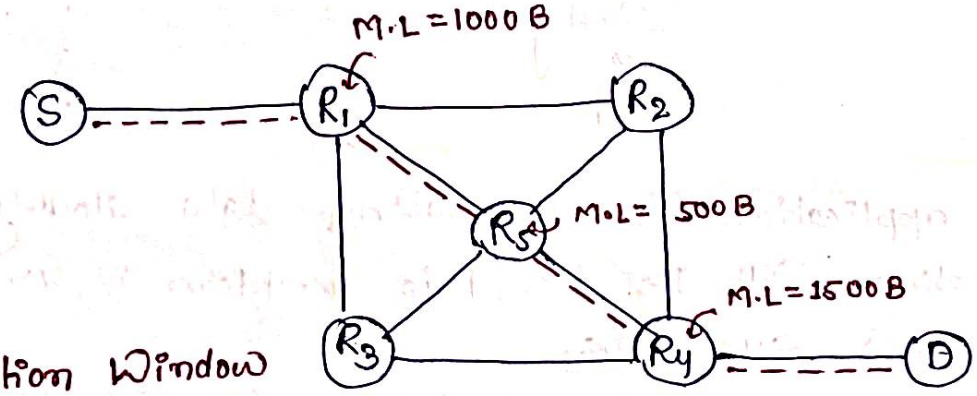
→ If the B.S is less than W.S, Sender should wait until B.S = W.S, then the data is transmitted.

→ Clarke suggested, delay the ack. So that I/y W.S will increases along with B.S.

So, the problem of Silly Window syndrome will be fastely resolve.

Congestion Policies of TCP

① Connection Establishment

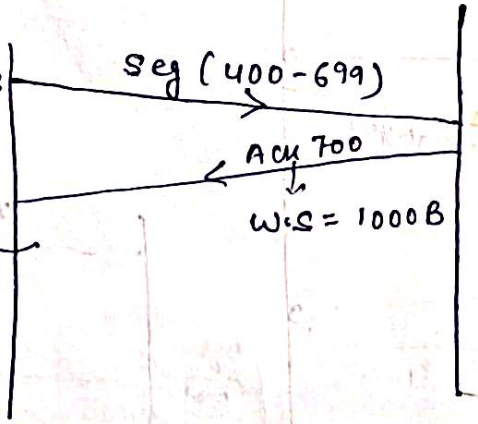


Congestion Window (cwnd) = 500 Byte
(taking smallest of all values of M.L)

→ S.W Parameter

② Data Transfer

S.W
Rwnd = 1000B
Receiver's Window

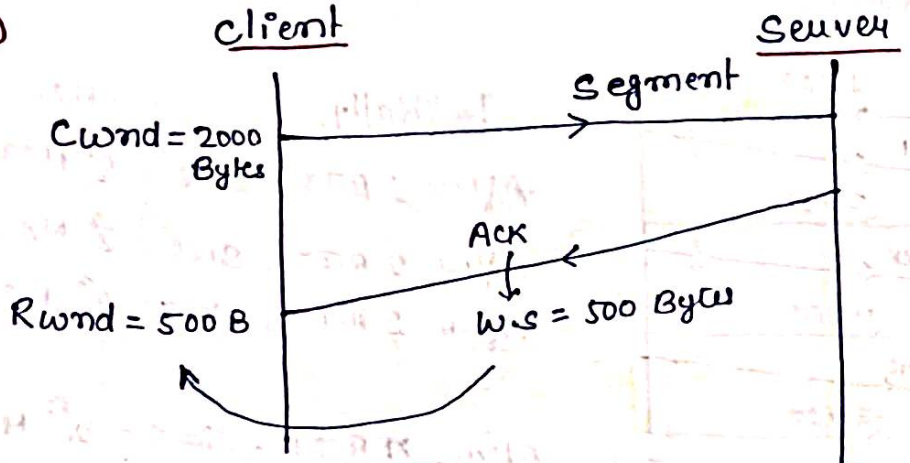


SW (Rwnd, Cwnd)

➡ Congestion Window will be known to sender during Connection establishment.

➡ Receivers Window will be known to sender during Data Transfer phase.

(i)

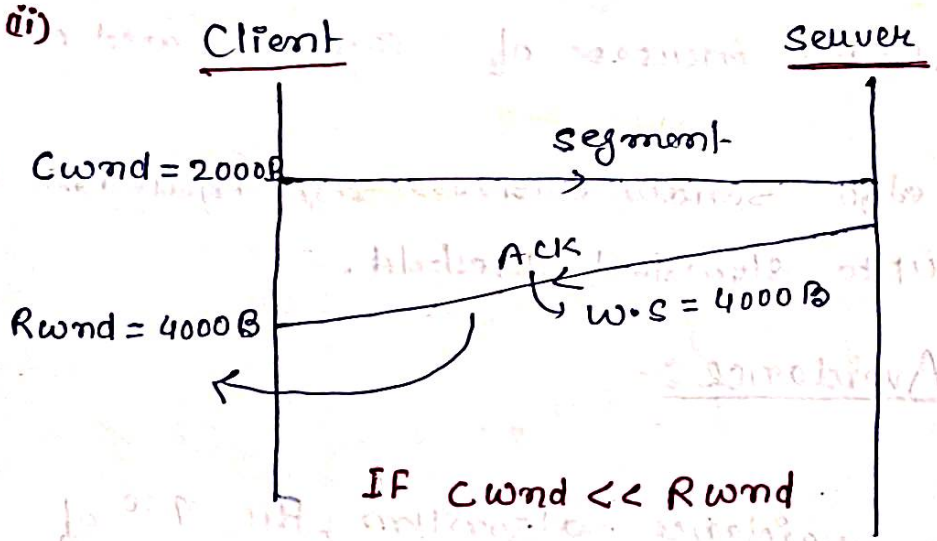


If Rwnd << Cwnd

S.w = Rwnd

⇒ flow control Policies of TCP

(ii)



IF Cwnd << Rwnd

S.w = Cwnd

⇒ Congestion Policies of TCP

Congestion Policies of TCP

(Cwnd << Rwnd)

(i) Slow Start Algorithm

(ii) Congestion avoidance

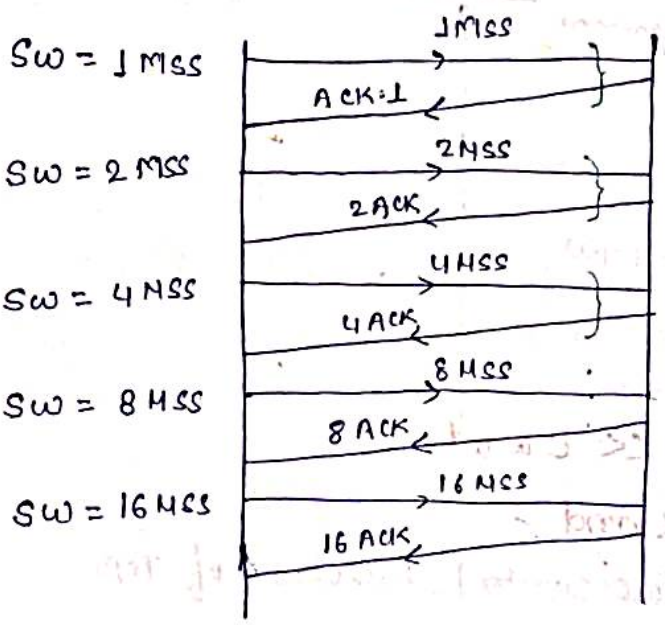
(iii) Congestion detection

(i) Slow Start Algorithm :-

MSS → Max^m Segment Size = 100 Bytes

Cwnd => 3200 Bytes

SSthreshold => 1600 Bytes



Initially, SWS = 1MSS

After 1 R.TT, SWS = 2^1 MSS

After 2 R.TT, SWS = 2^2 MSS

After 3 R.T.T, SWS = 2^3 MSS

⋮ ⋮

After N R.T.T, SWS = 2^n MSS

→ In slow start algo increase of S.W is based on number of ack.

→ In slow start algo sender window size increases exponentially upto slowstart threshold.

(ii) Congestion Avoidance :-

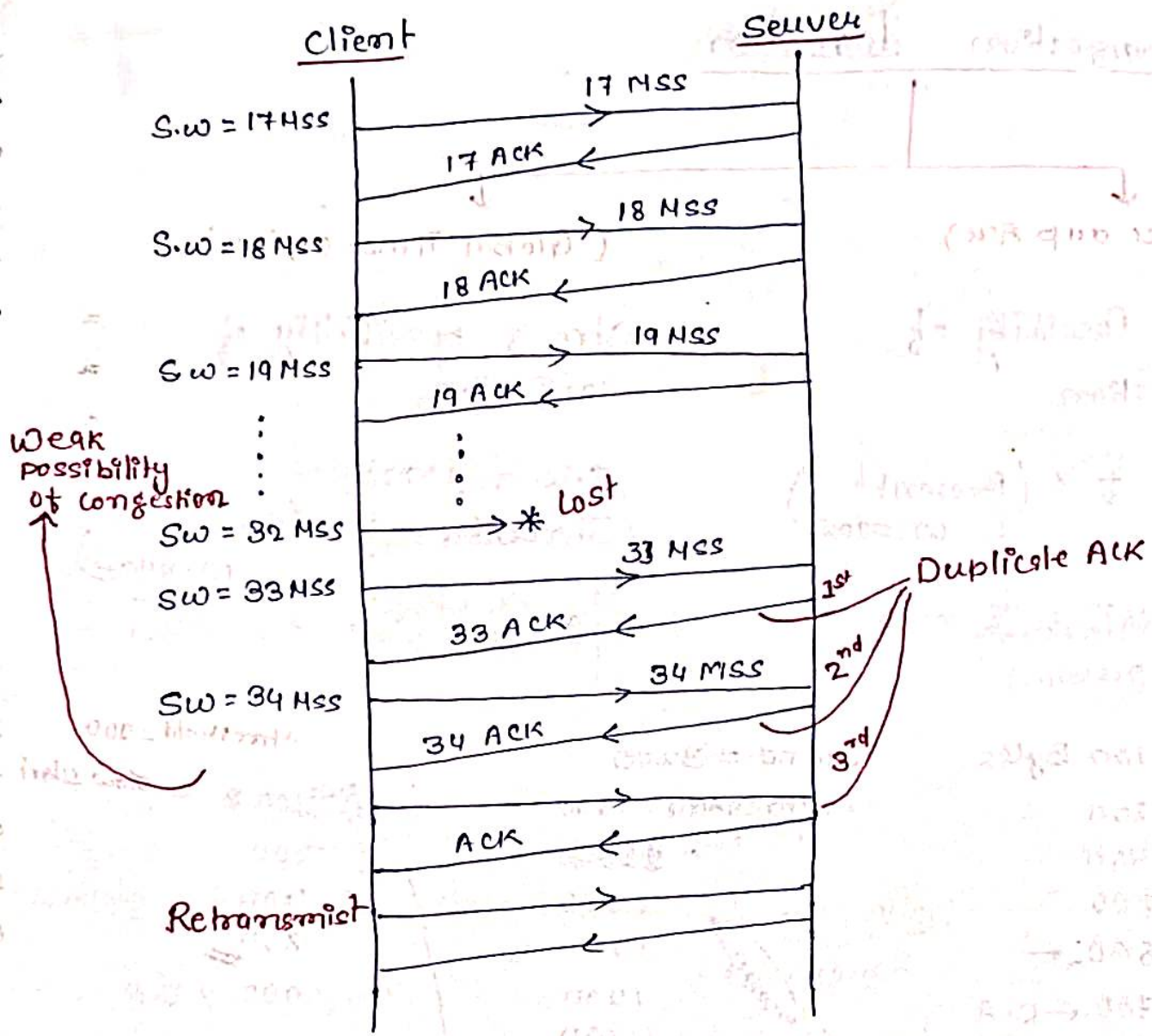
Client-

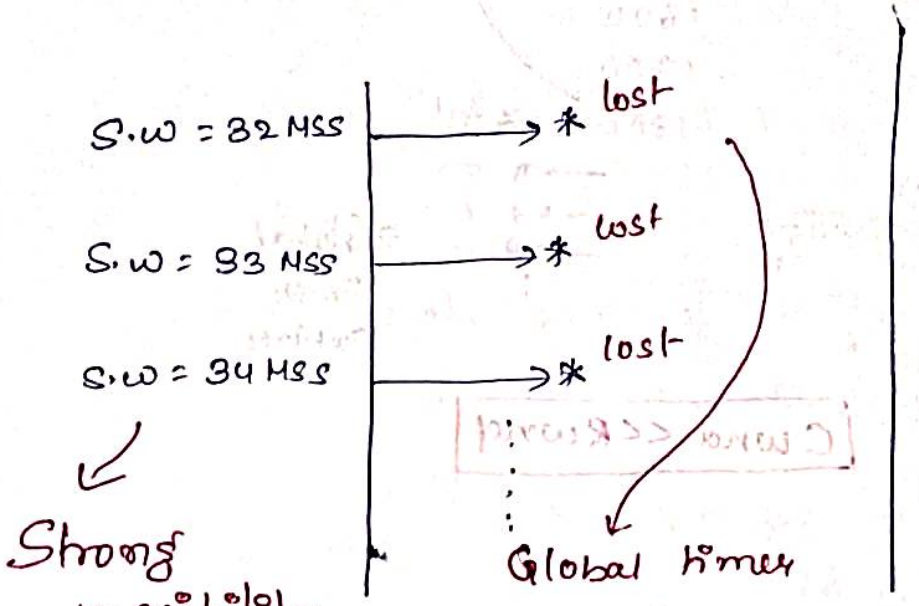
→ In congestion avoidance algorithm, the ↑se of Sender Window is based on RTT.

→ Once the data is lost & after 3 duplicate ack the data is accepted.

So, it is the Weak Possibilities of Congestion.

→ If the data is lost continuously until the glob timer expires, then it is known as a Strong possibilities of congestion.





Strong possibility of congestion

→ In Congestion detection

(three dup Ack)

→ Weak Possibility of Congestion

→ S.w = $\frac{1}{2}$ * (Present window)

Congestion Avoidance

(Global Timer Expires)

Strong possibility of congestion

S.w = 1 MSS

Threshold = $\frac{1}{2}$ * (Present window)

Slow start Algorithm

Ex

S.w = 100 Bytes

cwnd = 3200

SSthreshold = 1600

200
400
800
1600 ←
1700 ← C.A
1800
1900
2000
2100
2200 → * lost → Weak Possibility

(Linearly from Now)

sw = $\frac{1}{2}$ Present Window

1100
1200
1300
1400
1500
1600
1700
1800 → * lost
→ * lost
→ * lost
→ * lost ⇒ Global timer expires

threshold = 900

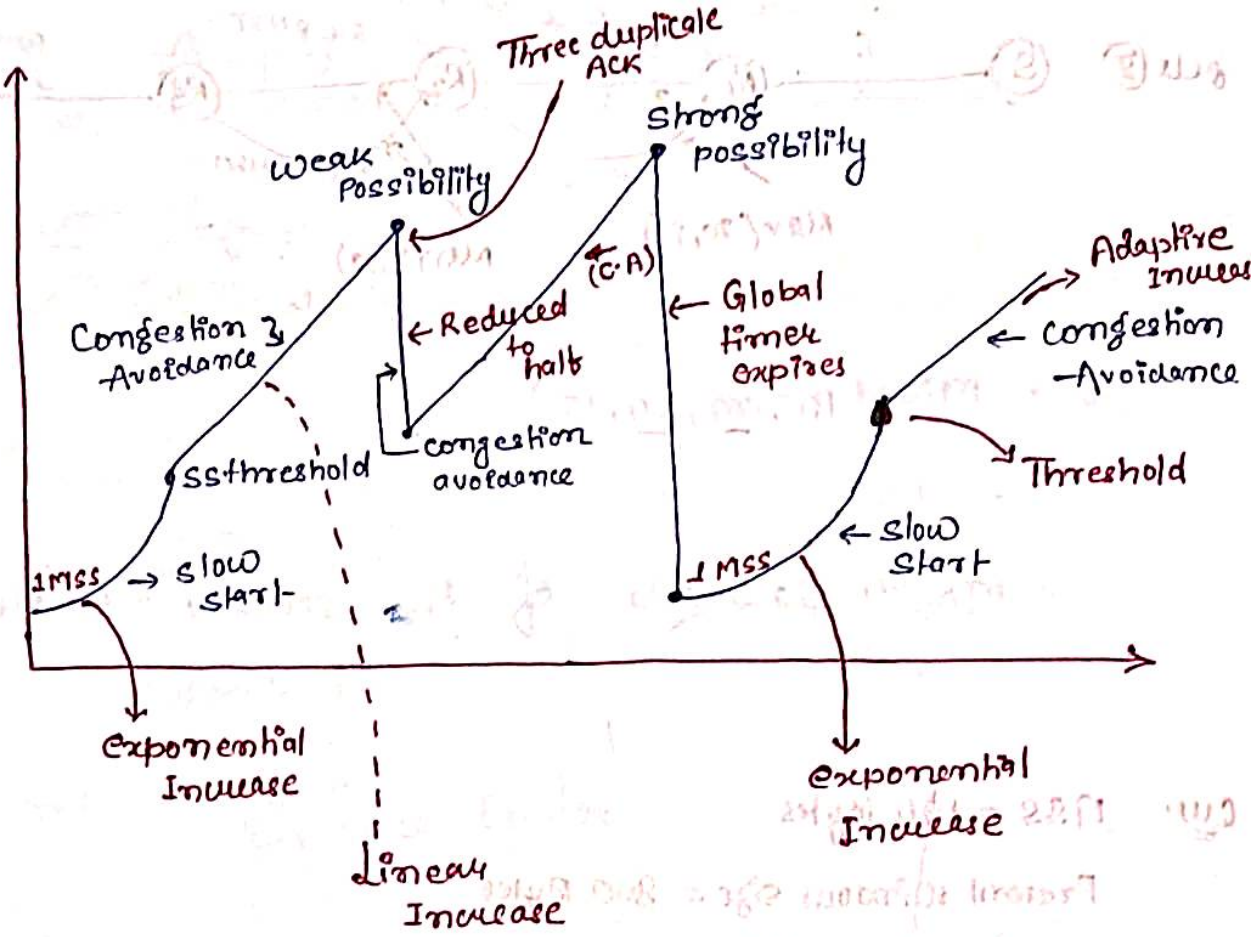
sw = 100 B ← slow start
200
400
800
900 → C.A

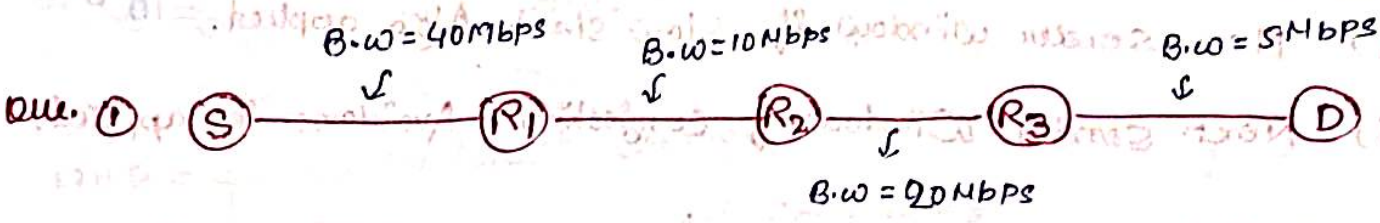
= ← three duplicate (Ack) data

→ data sent

Cwnd << Rwnd

***





Calculate Max data rate of sender to transmit to dest = ?

Sol^n → Min (40, 10, 20, 5)

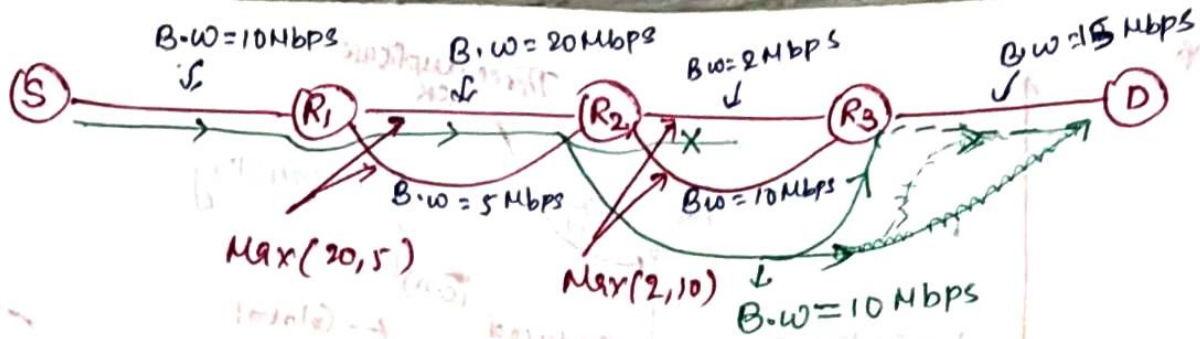
= 5

∴ Max^m data rate of sender to transmit to destination = Min^m B.W among all that

= 5 Mbps

Ans

Que (2)



Sol^n: Min(10, 20, 10, 15)

= 10

∴ Max^m data rate of the sender = 10 Mbps Ans

Que. MSS = 100 Bytes

Present Window size = 800 Bytes

What would be the

(i) Next sender window if Slow start Algo applied. = 16 MSS

(or) (ii) Next sender Window if Congestion Avoidance is applied.

= 9 MSS