TCP

↓

Transmissin Control protocol

atlantis

Date

Page

- TCP header:

4 byte ←	source port (16 bit)		Destination port (16 bit)
4B ←	Sequence number (32 bit)		
4B ←	Acknowledgement number (32 bit)		
4B ←	Header length (4 bit)	6 bits: reserved	URG (1bit) ACK (1bit) PSH (1) RST (1) SYN (1) FIN (1)	window size (or) adv window (16 bit)
4B ←	checksum (16 bit)		Urgent pointer (16 bit)
	Options (0 – 40 Byte)		
	Data		

Length of the header = (4+4+4+4+4) + 0 ⇒ 20B (minimum)

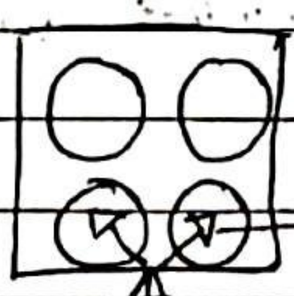
= (4+4+4+4+4) + 40 ⇒ 60B (maximum)

- Source port (SP) ↓ 16 bit — Destination port (DP) ↓ 16 bit

$2^{16} = (0 \text{ to } 2^{16}-1)$

$= (0 \text{ to } 65,535)$ → port number possible.

→ Tcp is end to end protocol.
(because of the port number)



→ Tcp could do multiplexing and Demultiplexing. (Take the packet and distribut properly b/w host proces)

HTTP - port number (80) → (website)

FTP - (21) → (To download a file)

Telnet - (23) → (Remote login)

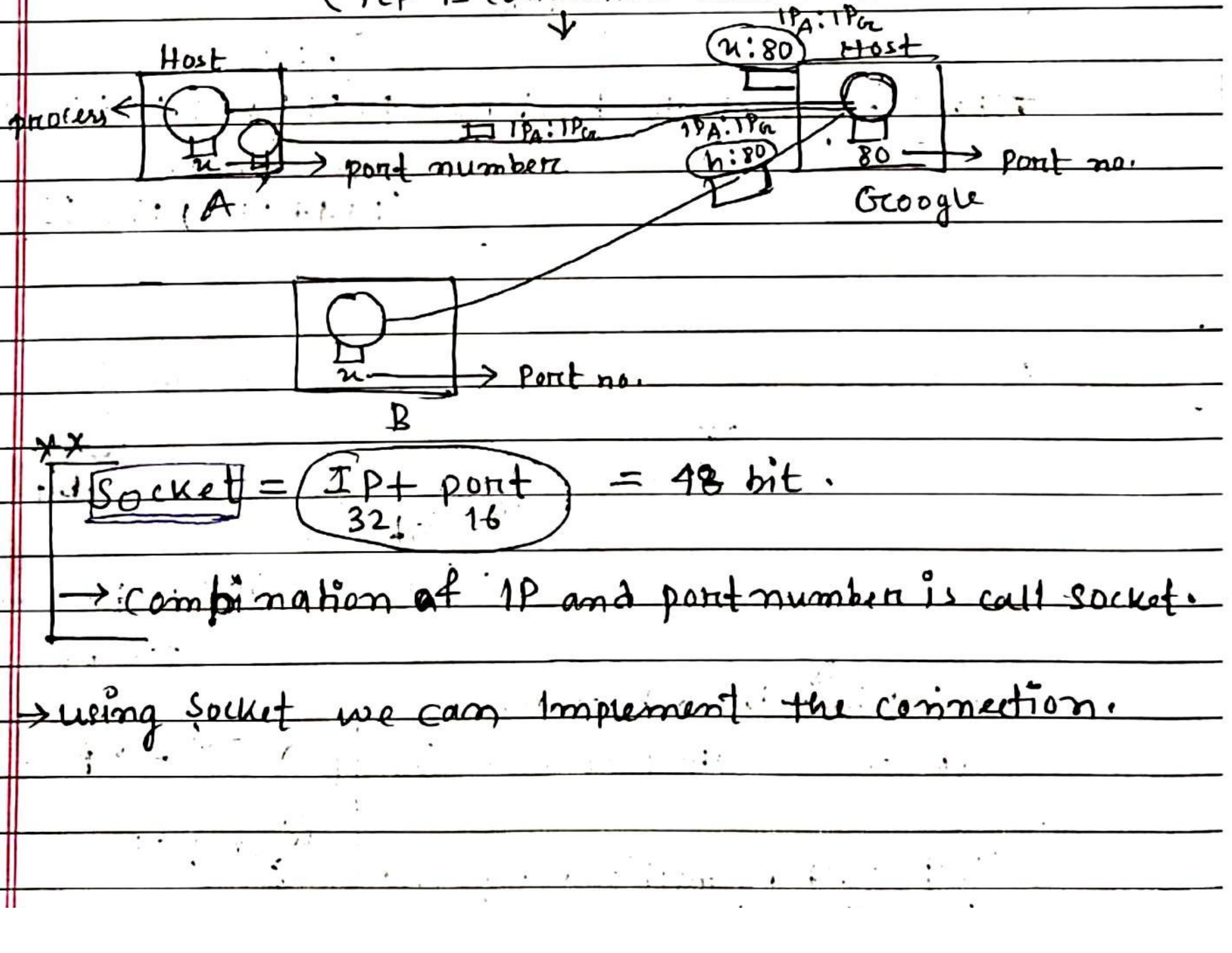
SMTP - (25) → (email transfer)

0 – 1023 } well known — IANA (Internet

1024 – 49,151 } Reserved — IANA (organization) (Internet Assigned Numbers Authority)

49,152 – 65,535 } used by general public.

(TCP is connection oriented)



** Socket = IP + port = 48 bit.
32 16

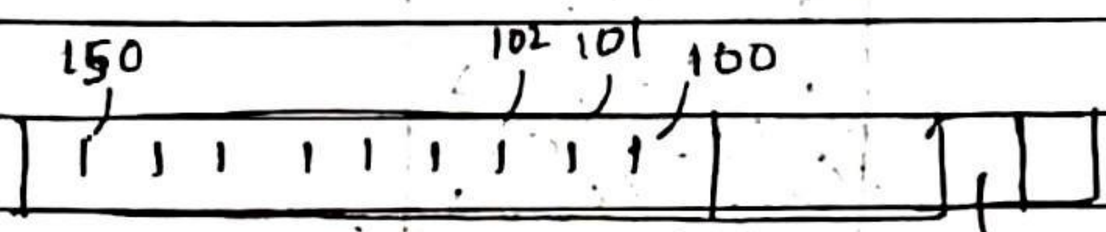
→ Combination of IP and port number is call socket.

→ using socket we can implement the connection.

- **Sequence number :**

→ TCP is Byte stream protocol.
→ IP is packet stream
→ DLL (Data link layer) - HDLC is bit stream.

TCP - Segment (Packet)



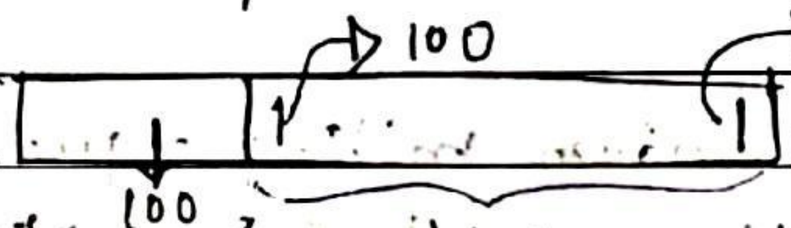
(100-150) all the byte present in this segment

→ 100 first bit sequence no is 100.

- **Acknowledgement number -**

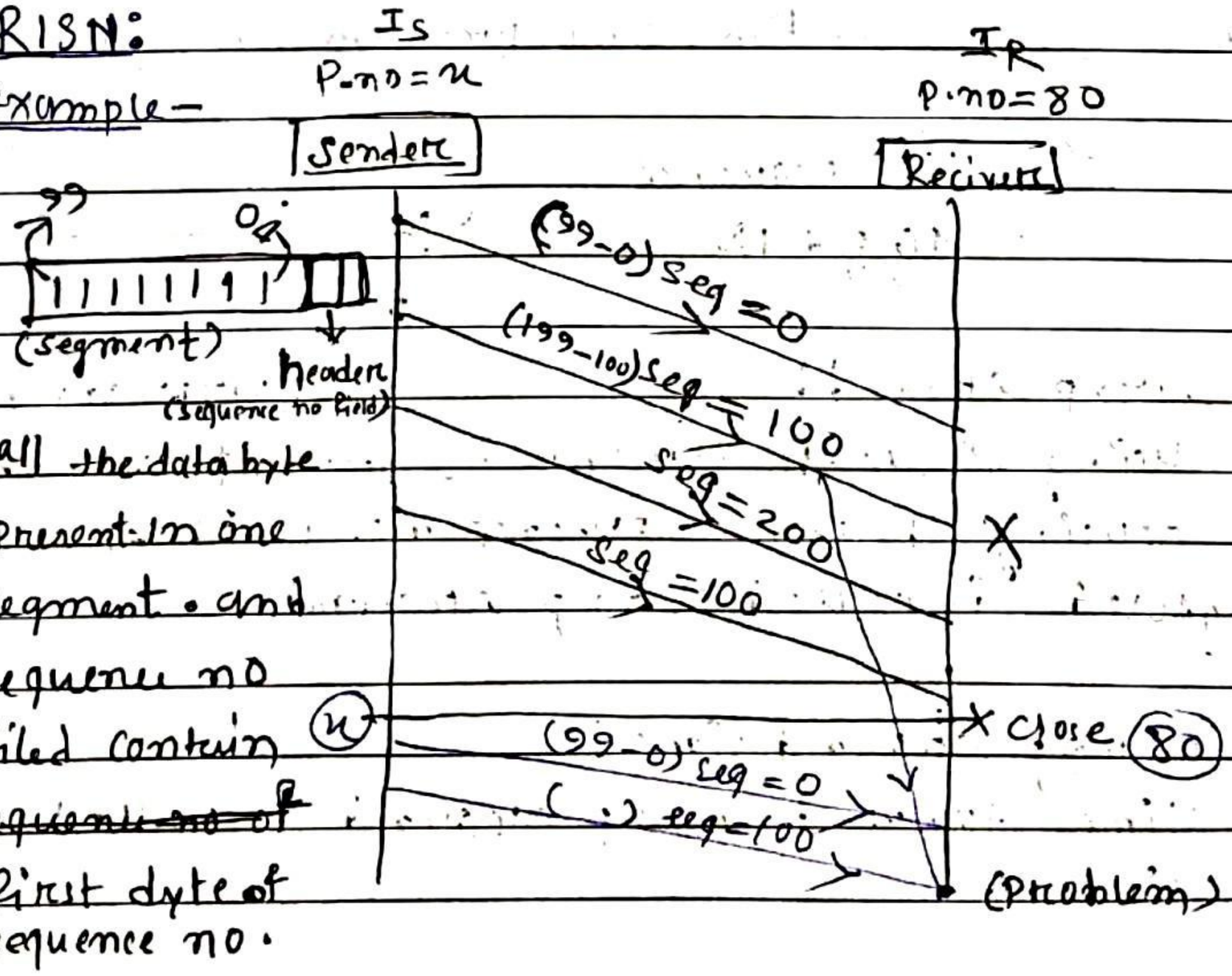
→ last bit sequence number + 1.

Segment -



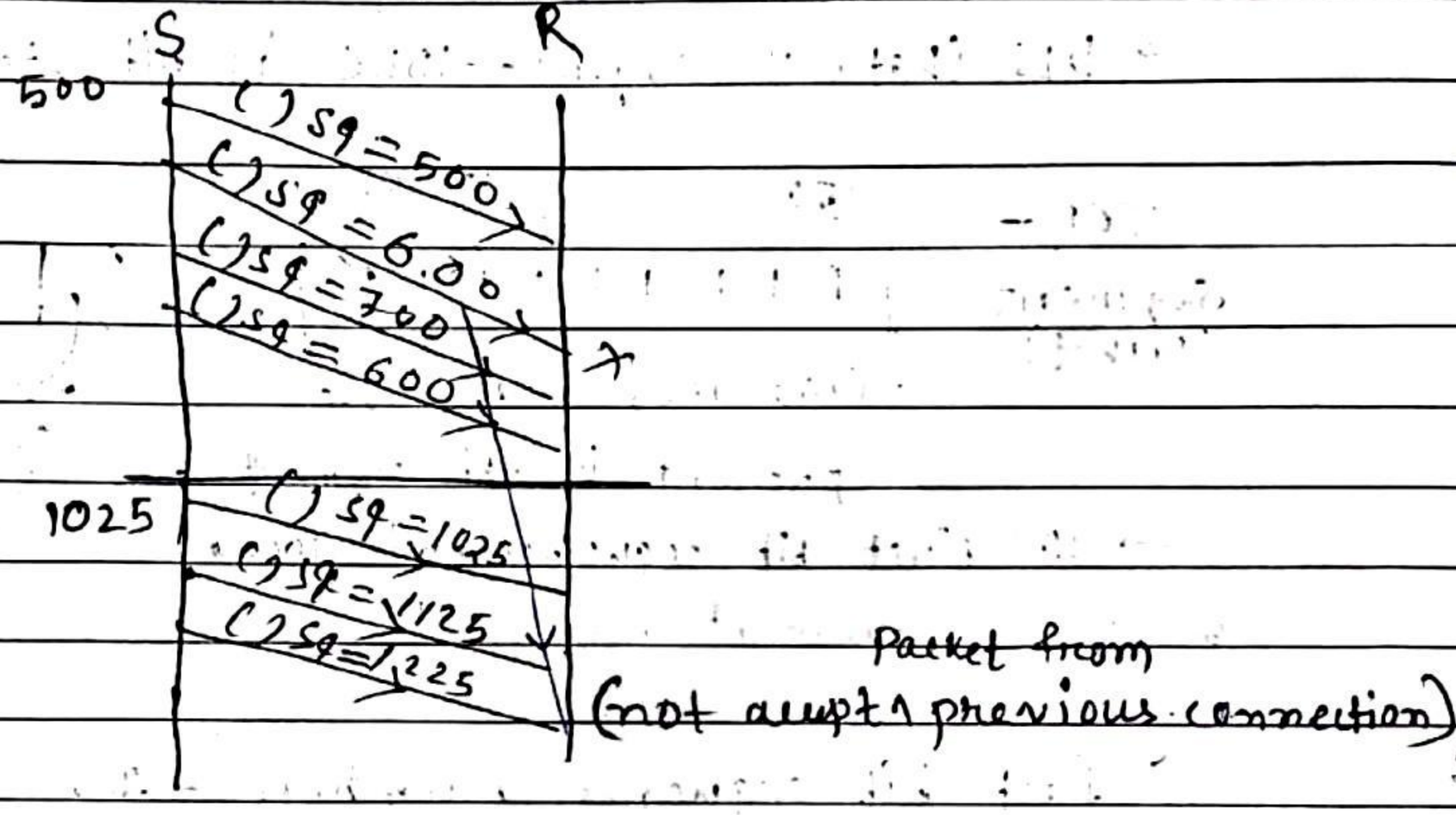
- **RISN:**

Example -



→ all the data byte present in one segment. and sequence no filed contain ~~sequence no of~~ first byte of sequence no.

→ If you choose randomly sequence number then chances are getting ~~not~~ collision ~~is~~ very less. next time is compare to



→ TCP uses Random initial sequence number (RISN). (to stop accepting packet from previous connection in the present connection).

- Wrap around time and problem on wrap around time -

2^{32} = (ACK) sequence.

= 4GB + 1B

(Wrap around time)

Wrap around:

Using of all the sequence no present and then reusing the same sequence no. which have already used, is called Wrap around.

Wrap around time:

Time taken to wrap around is called wrap around time.

Example -

Bandwidth (BW) = 1 MBPS

1 sec = 1 MB

10^6 B = 1 sec.

10^6 seq num = 1 sec

1 seq num = $\frac{1}{10^6}$ sec

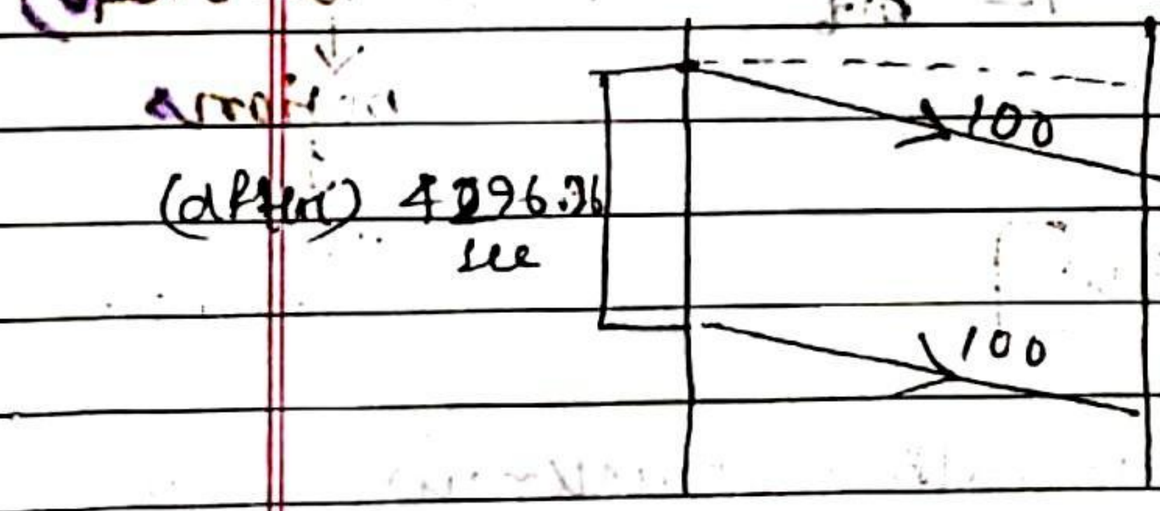
2^{32} seq num = $\frac{2^{32}}{10^6}$ sec

W.A.T = 4296.967296 sec.

1 M = 1000000 B

1 G = 10^9 B

- Life time (LT) = 3 min = 180 sec. (worst sec)
  (time take to reach from one destination to other destination)



→ using same sequence number after the end of first, no problem

Example -

Bandwidth (BW) = 1 GBPS.

10^9 B = 1 sec,

1 sec = 1 GB

10^9 B = 1 sec.

10^9 seq num = 1 sec

1 seq num = $\frac{1}{10^9}$ sec

WAT = 2^{32} seq num = $\frac{2^{32}}{10^9}$ = 4.294 sec.

→ When Bandwidth time increase then W.A.T decreases.

WAT < LT → problem

Example:

BW = 1 GBPS } given calculate what are the
LT = 180 sec } minimum no of sequence no required?

1 sec = 1GB
1 sec = 1 seq Gr seq.
180 sec = 180 X 1 Gr Seq

So, minimum seq num required to avoid WA within LT = 180 X 1Gr seq num.

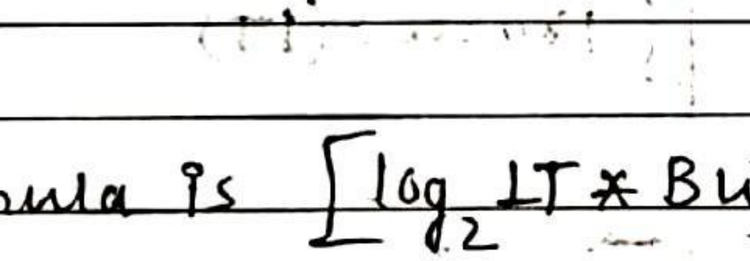
$\lceil \log_2 (180 \times 10^9) \rceil = 42$ bit (42 − 32 = 10 bit req)

↓
options
↓
Time stamp.

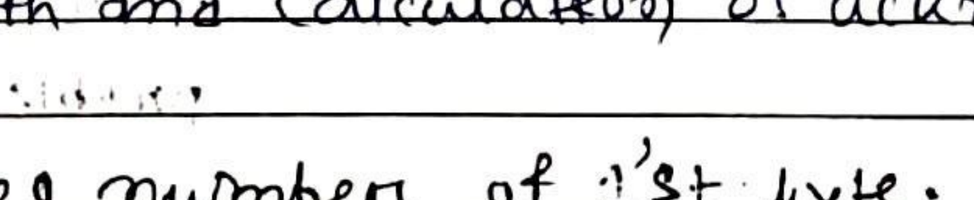
formula is $\lceil \log_2 LT * BW \rceil$

- Header length and calculation of acknowledgement number.

Seq num: seq number of 1'st byte.
Ack num: Seq number of byte expected next.



IP datagram -



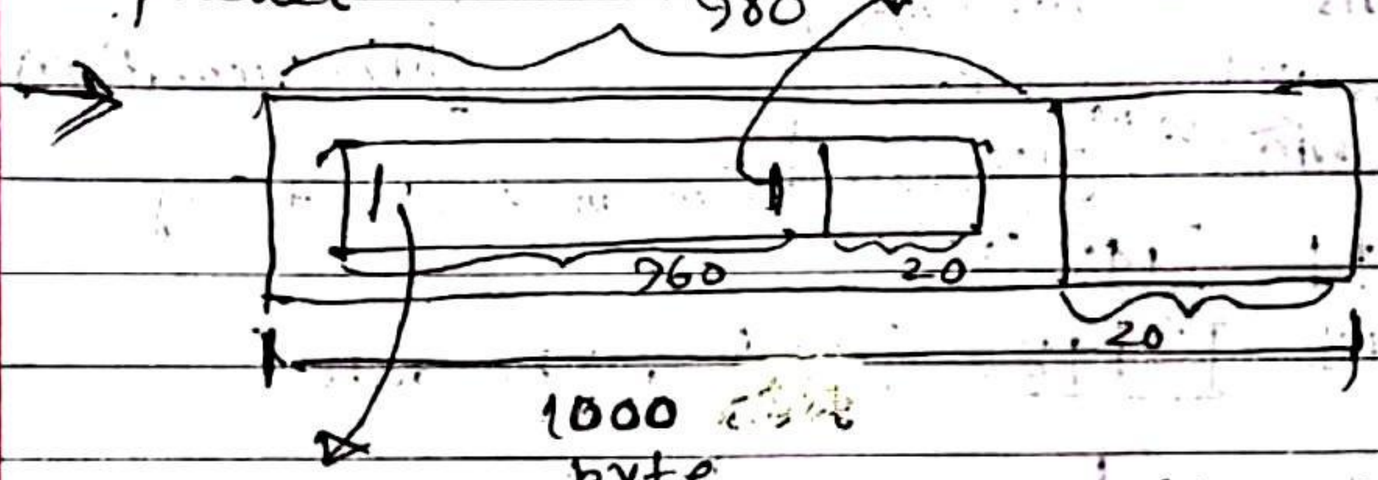
Example-

Total length = 1000 byte } give for ip data gram.
Header length = 5

Inside IP data gram there is a TCP segment and the H2 of TCP segment = 5

Sequence number of TCP segment = 100.

find the acknowledgement number for this packet.



(100+960-1)
↳ (1059)

(header length minimum 20 byte)

here, Total lenght of datagram = 1000 byte
header length of datagram = 5×4 = 20

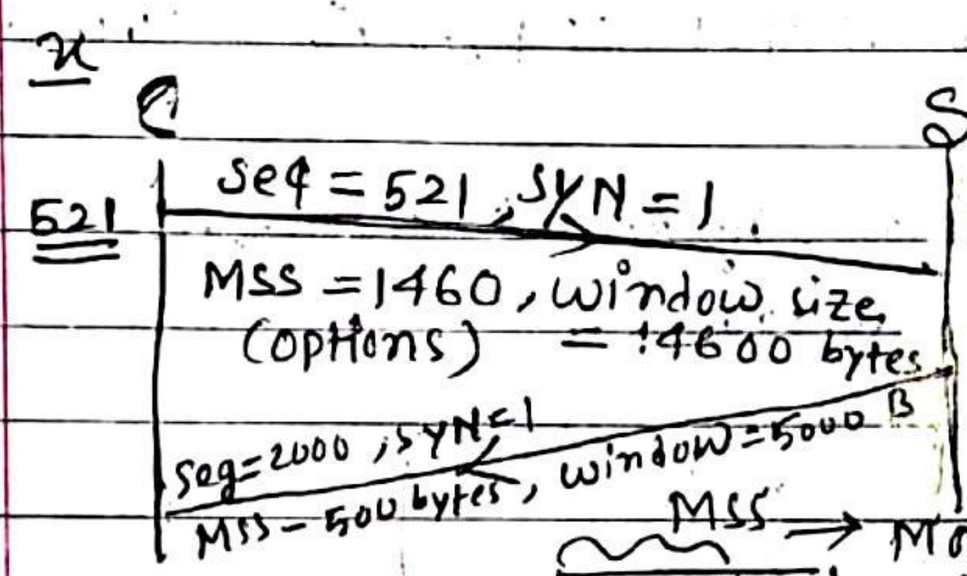
Header length of TCP segment = 5×4 = 20
Total length of TCP segment = body = (1000-20)-20
= 960

In TCP segment 1st bit sequence no = 100
last bit sequence no = (100+960-1)
= 1059

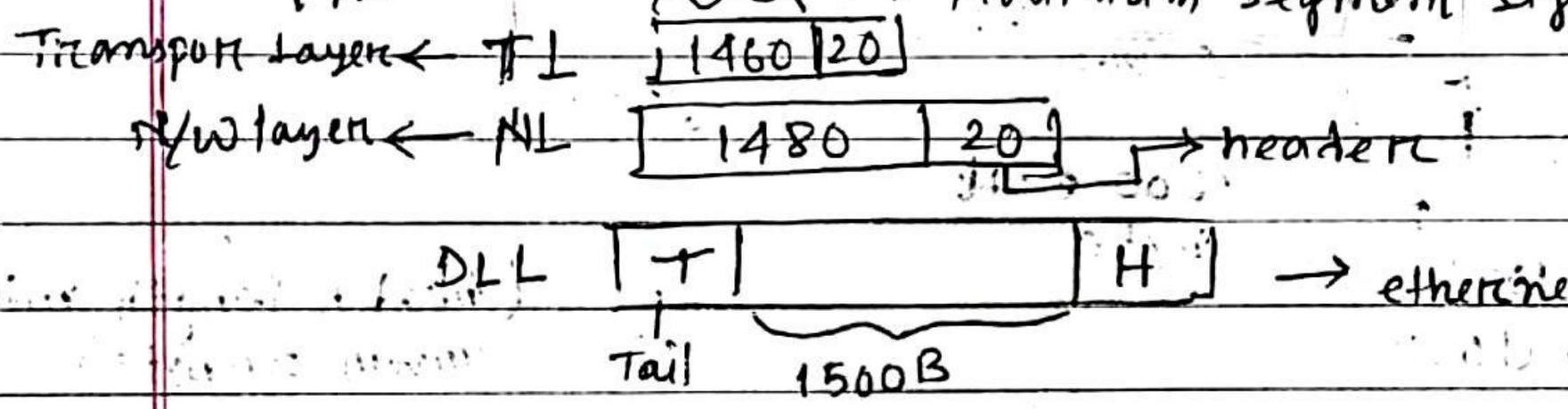
So acknowledgement number for this packet is 1060.

- **TCP connection establishment:**
  - SYN → Synchronization Flag.
  - ACK → Acknowledgement Flag.

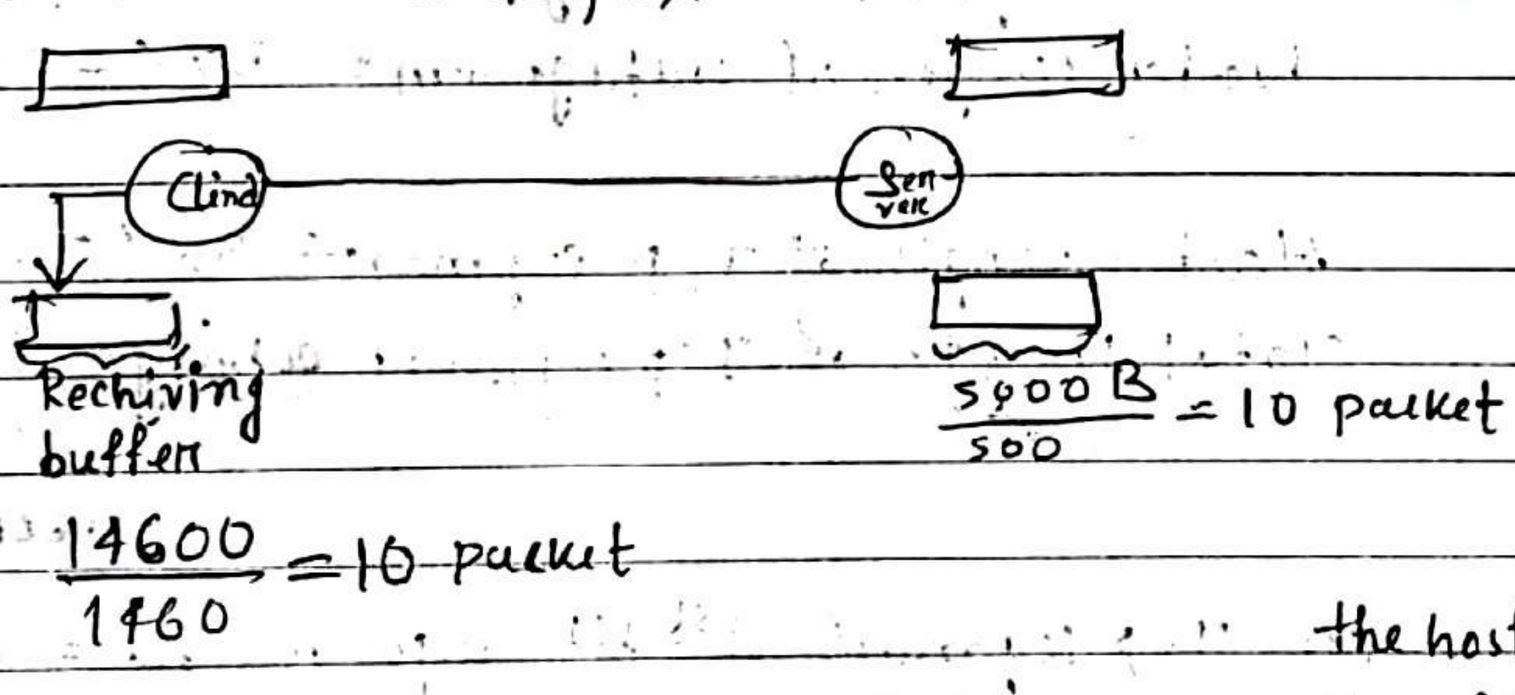
TCP is connection oriented - it has 3 fases
(i) connection establish-ment.
(ii) Data transfer.
(iii) connection termination



MSS → Maximum segment size.



→ TCP is Full duplex.



$\frac{14600}{1460}$ = 10 packet

window size means: what is the capacity of Clind (Rechiver) (window is used to flow control).

SYN = 1 seq num

1 → means whenever you send a packet with "SYN = 1" whatever sequence number present in the packet that is consumed.

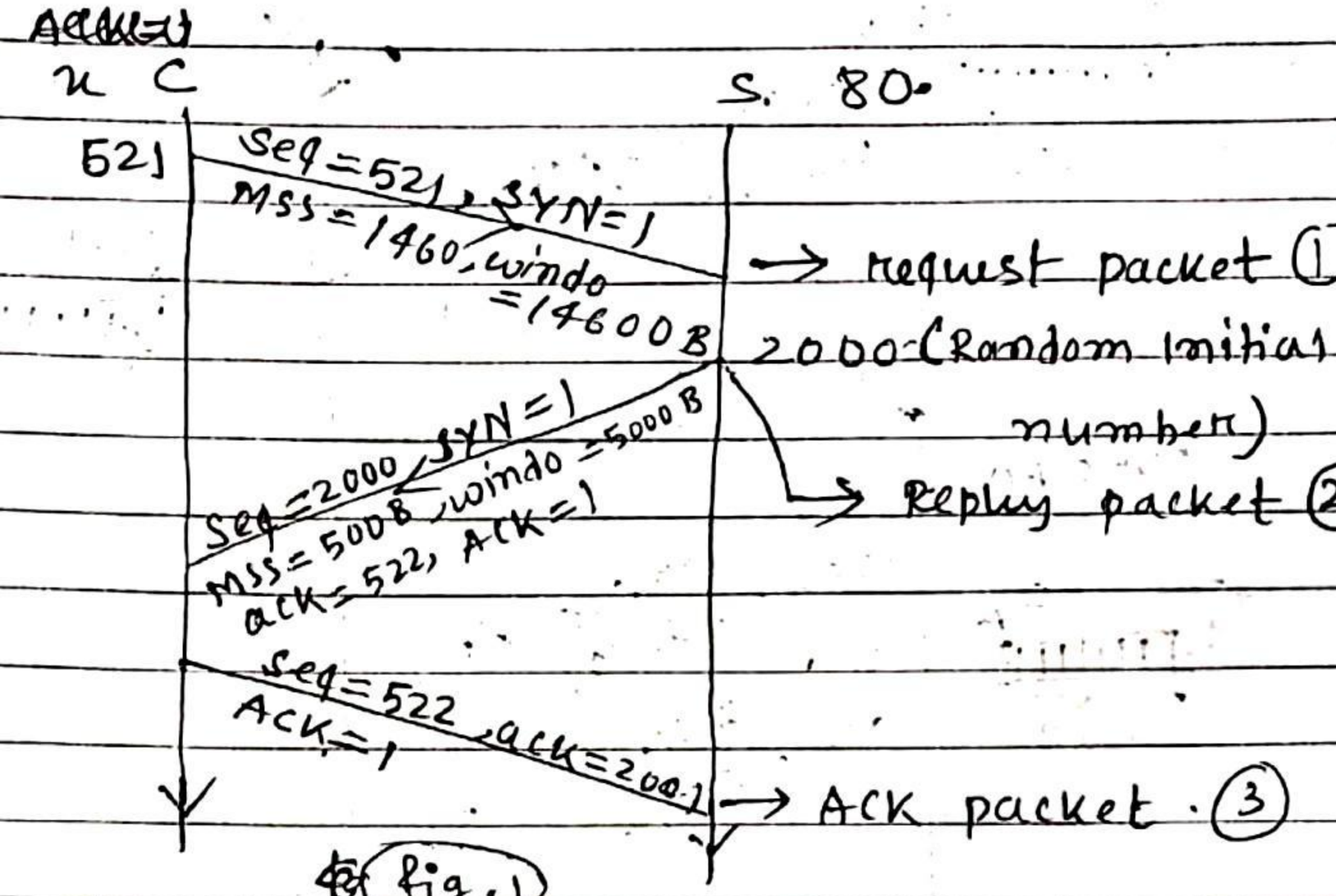


Fig. 1

→

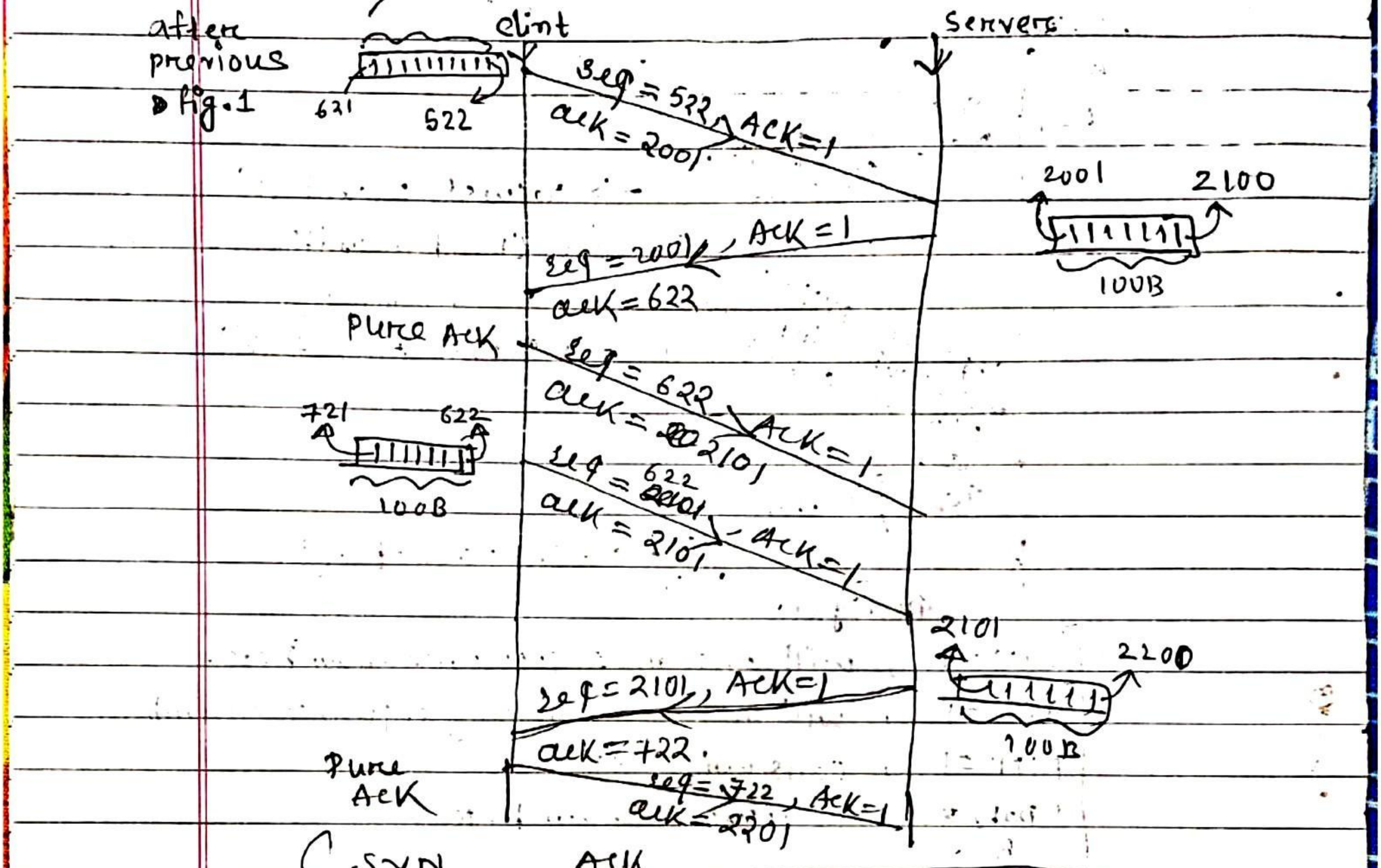
SYN=1	, will consum 1 sequence number.
ACK=1	, will not consum any sequence number.
FIN=1	, consum 1 SN .
1 Data = 1 B	, 1 sequence number

→ After sending this 3 packets connection will be established therefore it is also called as "3-way hand shake".

- Data transfer after connection establishment —

→ next page.

→ every packt
Canny ACK. (ACK=1)



	SYN	ACK	
SYN	1	0	— I, segment / request
↳ seen; flag	1	1	— II, segment / replay
ACK	0	1	— ACK is present in the header.
↳ acknowledge -ment flag.	0	0	— not possible.

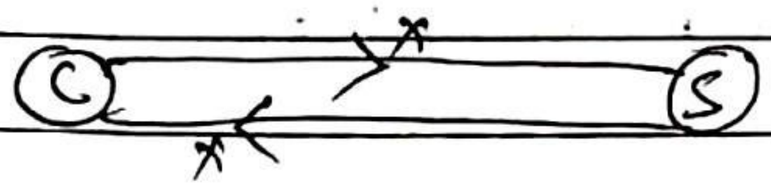
close connection.

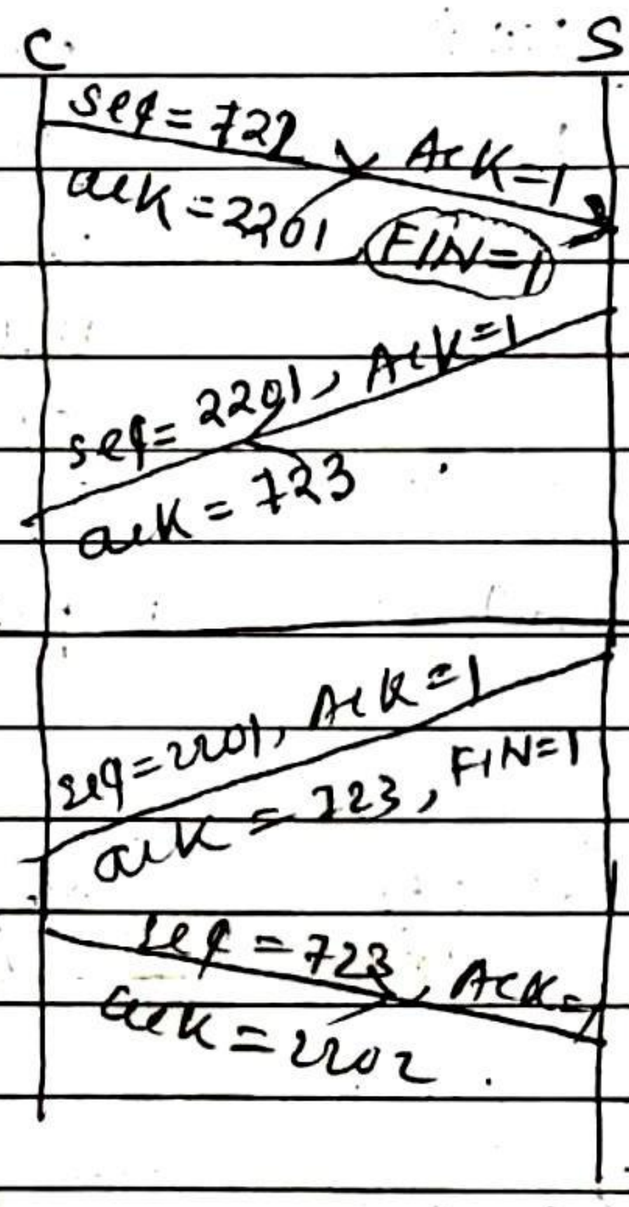
FIN=1

- Connection termination — (close connection) b/w C & S

→ TCP is a full Duplex connection.

FIN=1
↳ finish





(3) → can Data C→S ✓
S→C ✓

→ can Ack C→S ✓(piggy)
S→C (PB) ✓

→ 3 packet for Connection estublishment,
→ 4 " " " termination.

3 phases — Connection Astublishment.
↓
Data transfer.
↓
termination.

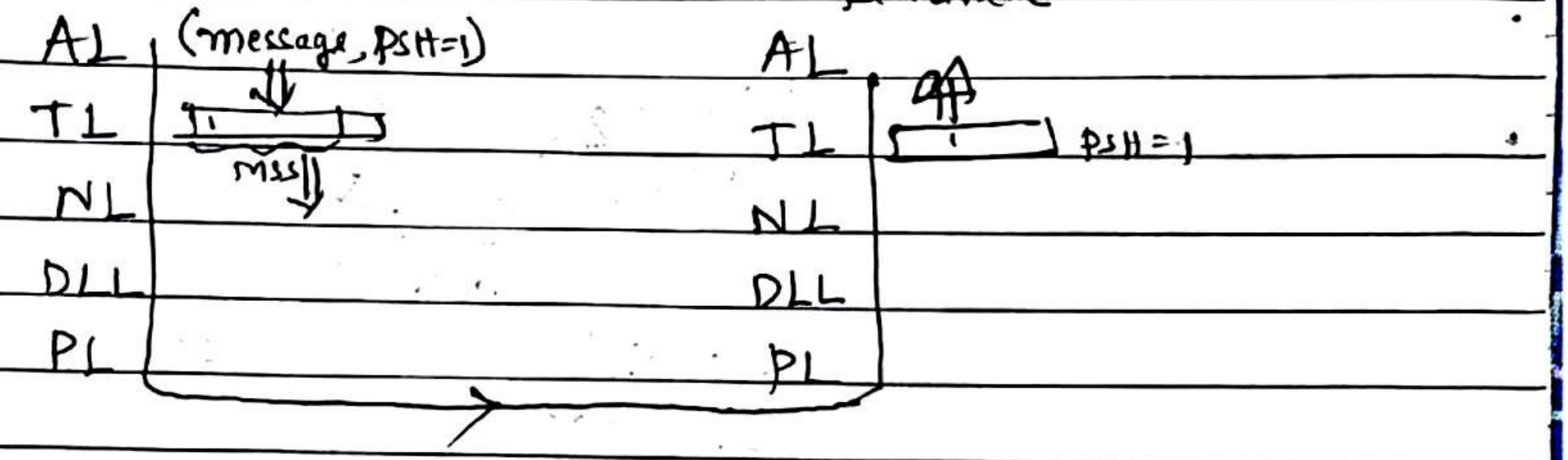
3-Flg used -

1) ~~SEQ~~ SYN : Syn Synchronizing seq number.
2) ACK : Indicate whether Ack num field is valid.
3) FIN : Request for connection termination.

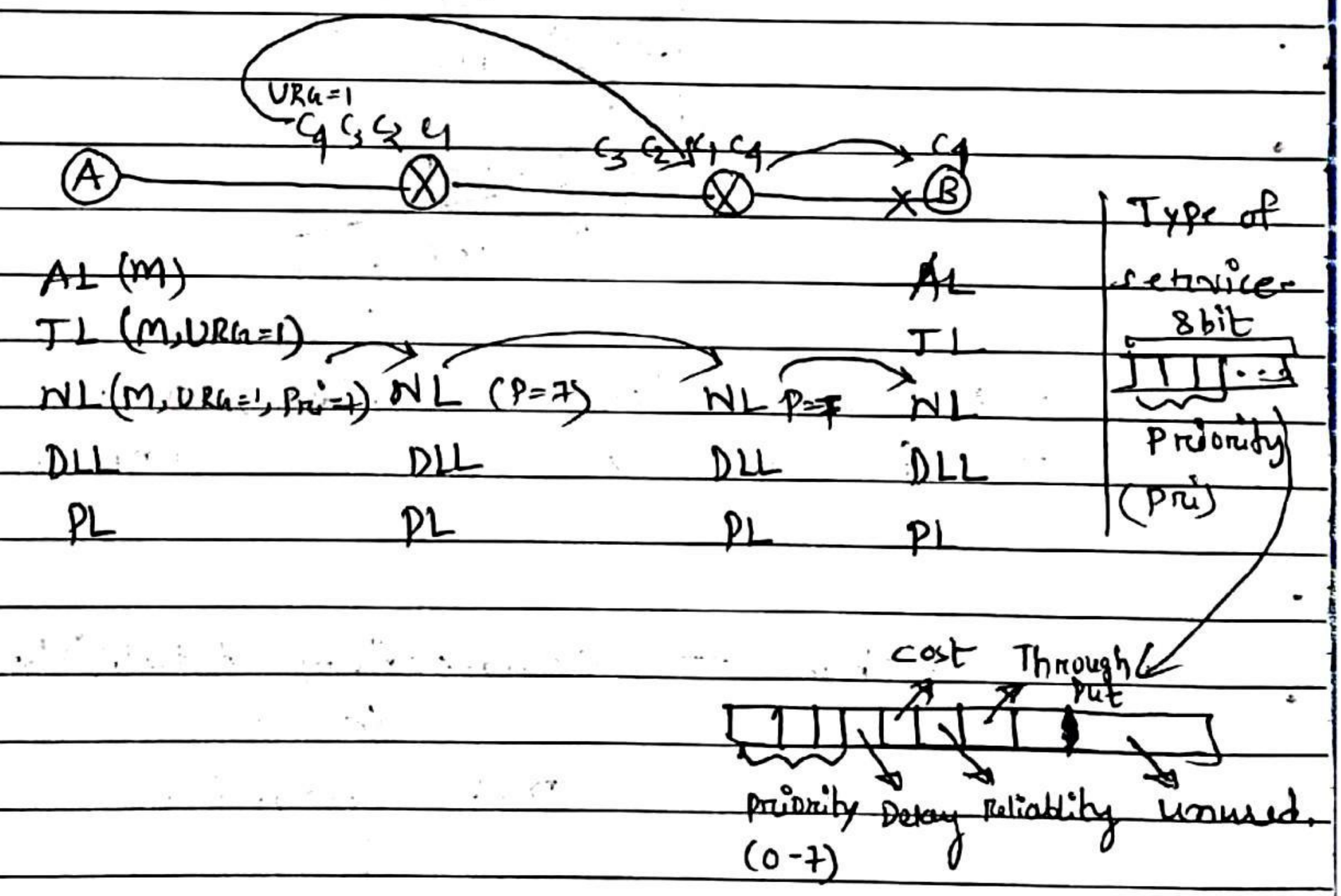
atlantis
Date
Page

- PSH flag - (push flag)

→ ~~PSH~~ PSH used to push the data immediately downward as well as Upward.



- URG flag and Urgent pointer -



When Delay = 1 means - send data with least Delay.
Cost = 1 " — " " " cost.
reliablity = 1 " — " " high reliablity

atlantis
Date
Page

Urgent pointer –

Urgent pointer indicates till what part Data is urgent.

Seq = 1000

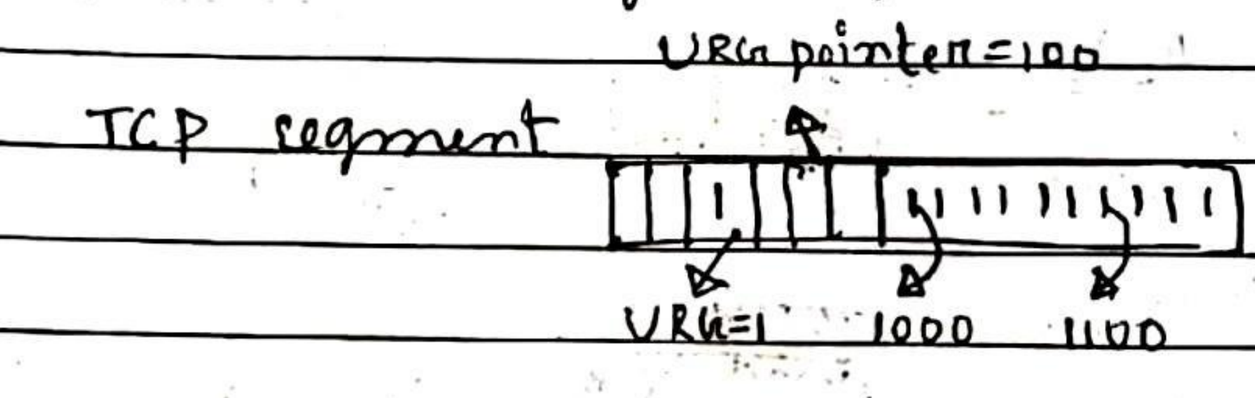
URG = 1

Urg pointer = 100

⇒ Seq + Urg pointer

⇒ 1000 + 100

⇒ 1100 — Urgent Byte.



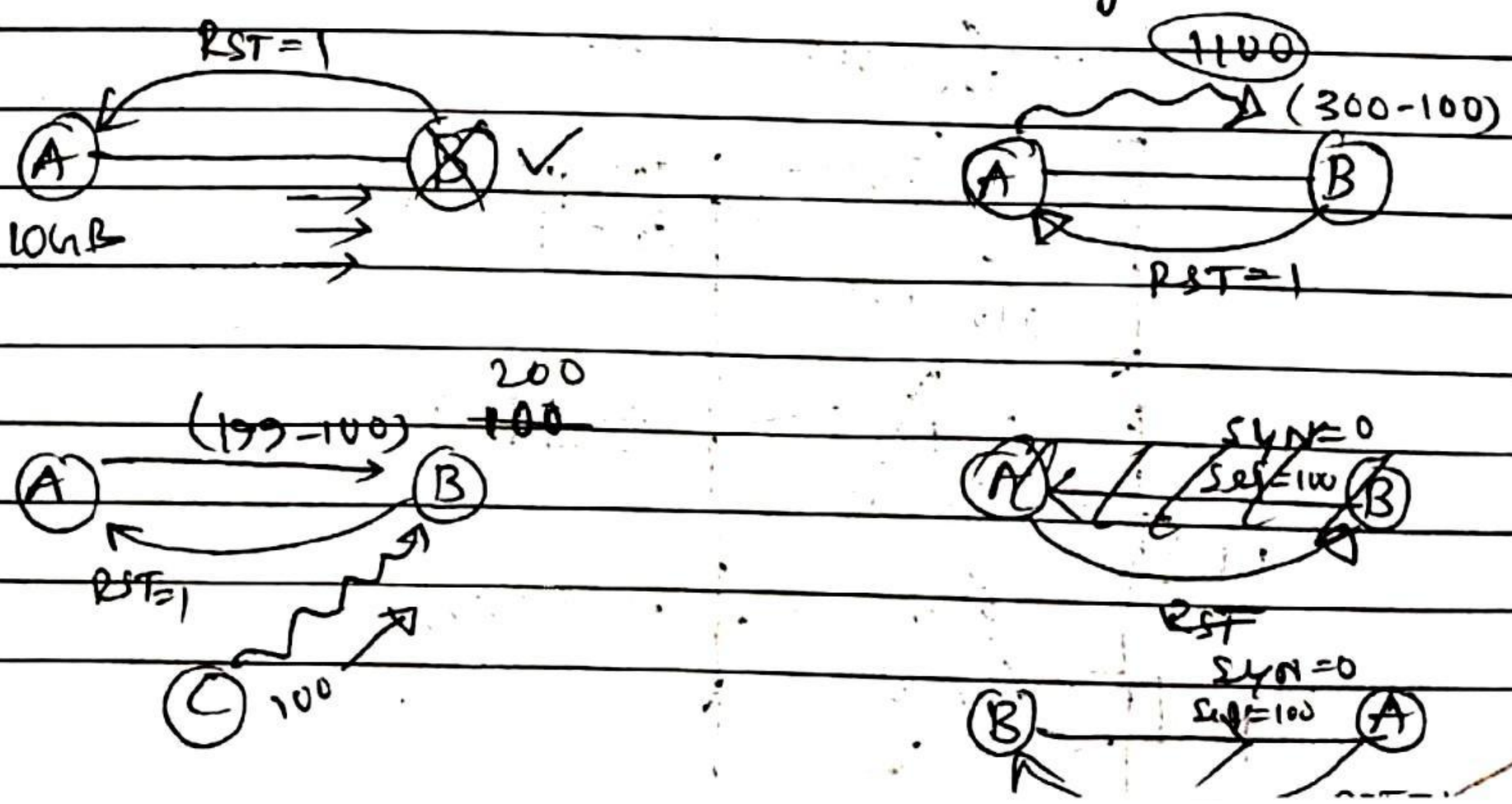
1000 — 1100

⇒ 101 – Urgent.

- **RST Flag (reset flag) –**

to terminate connection.

→ RST Flag used whenever any Wrong in connection.



- Flow control using advertisement window — (16 Bit)

- window size & Adv window -

↓

(flow control).

→ Sitsender window should not be greater than Rechiver window.

Window is free because old data sended to application layer.

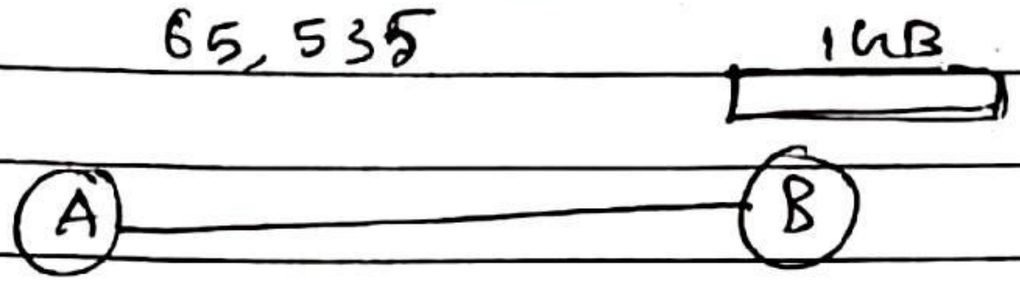
persistent timer - (PT)

Disadvantage of window size =>

→ adv window used for flow control.

$2^{16} - 1$

65,535



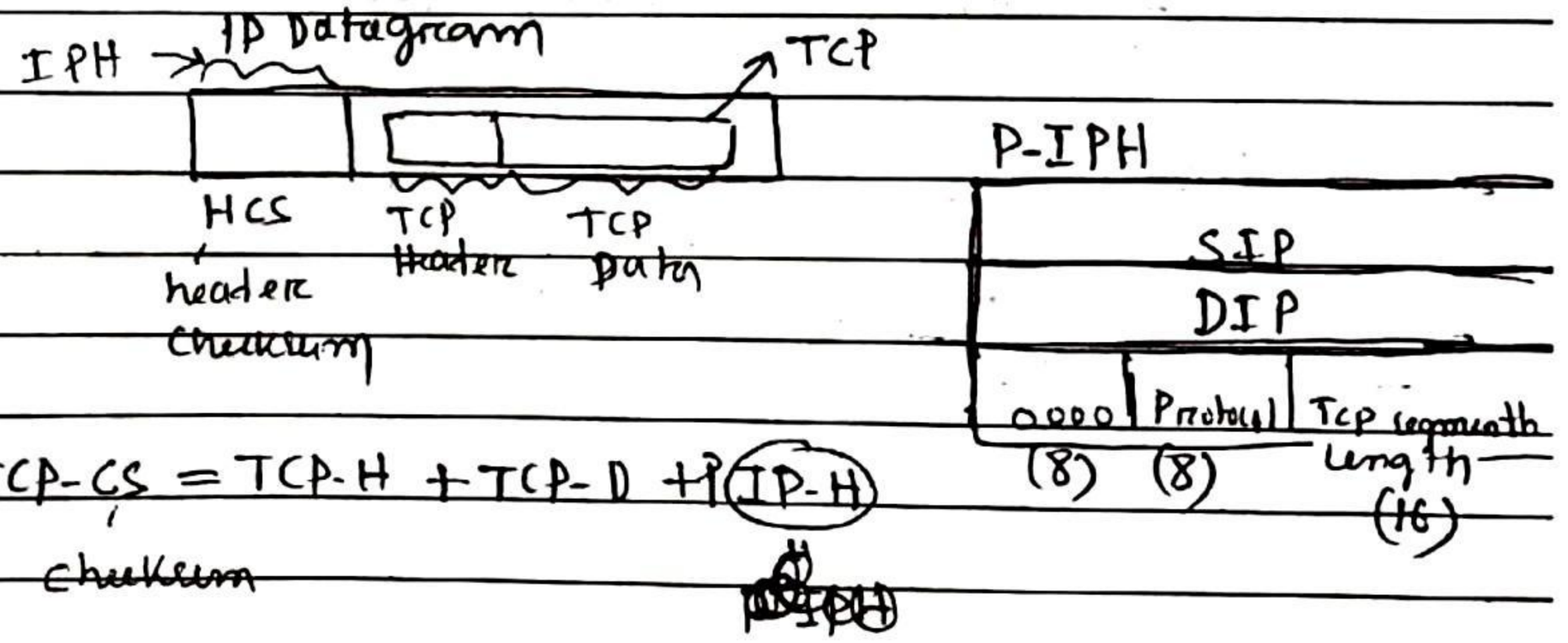
16 + 14 (option)

= 30 bit

= $2^{30} - 1$

= 1GB

- Checksum — (16)



TCP-CS = TCP-H + TCP-D + IP-H

checksum

Q. Why checksum used two time in IPH & TCPH.

→ for double checking.

• Options :

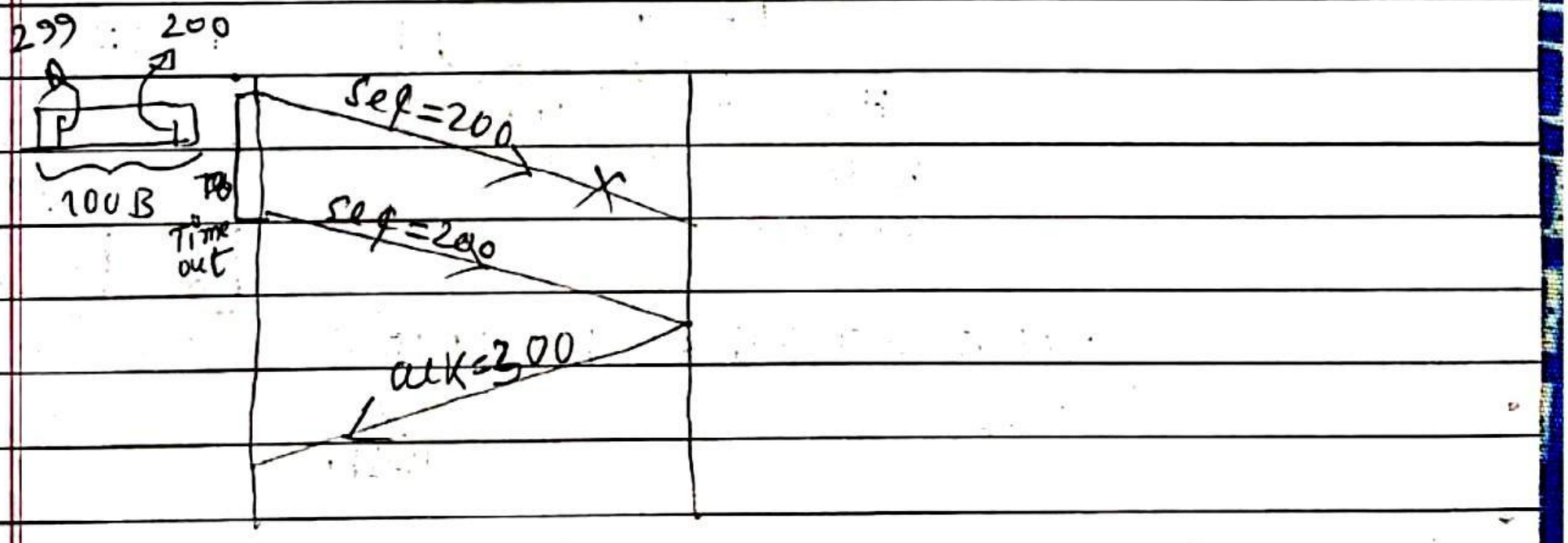
(I) Time stamp (WAT < LT) (wrap around time / life time) → (00) 100
(II) window size extension (01) 100
(III) Parameter negotiation.
(IV) padding.

• Retransmissions in TCP :

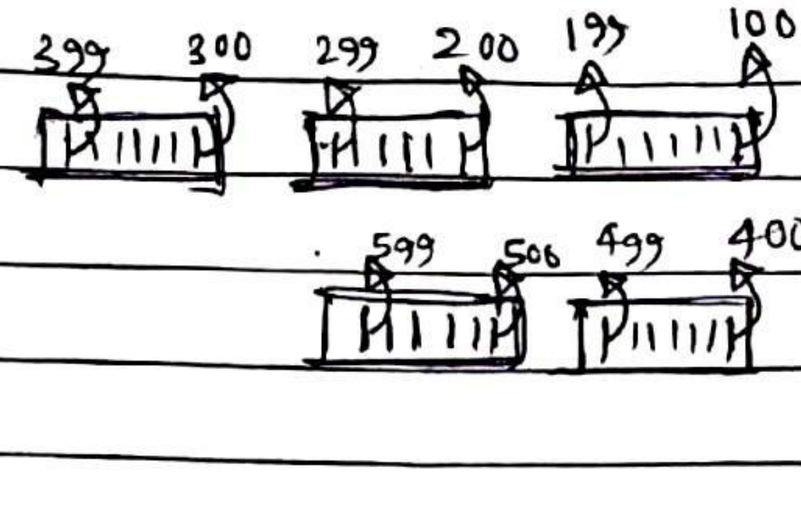
→ TCP use as (SR + GBN (Go Back N))

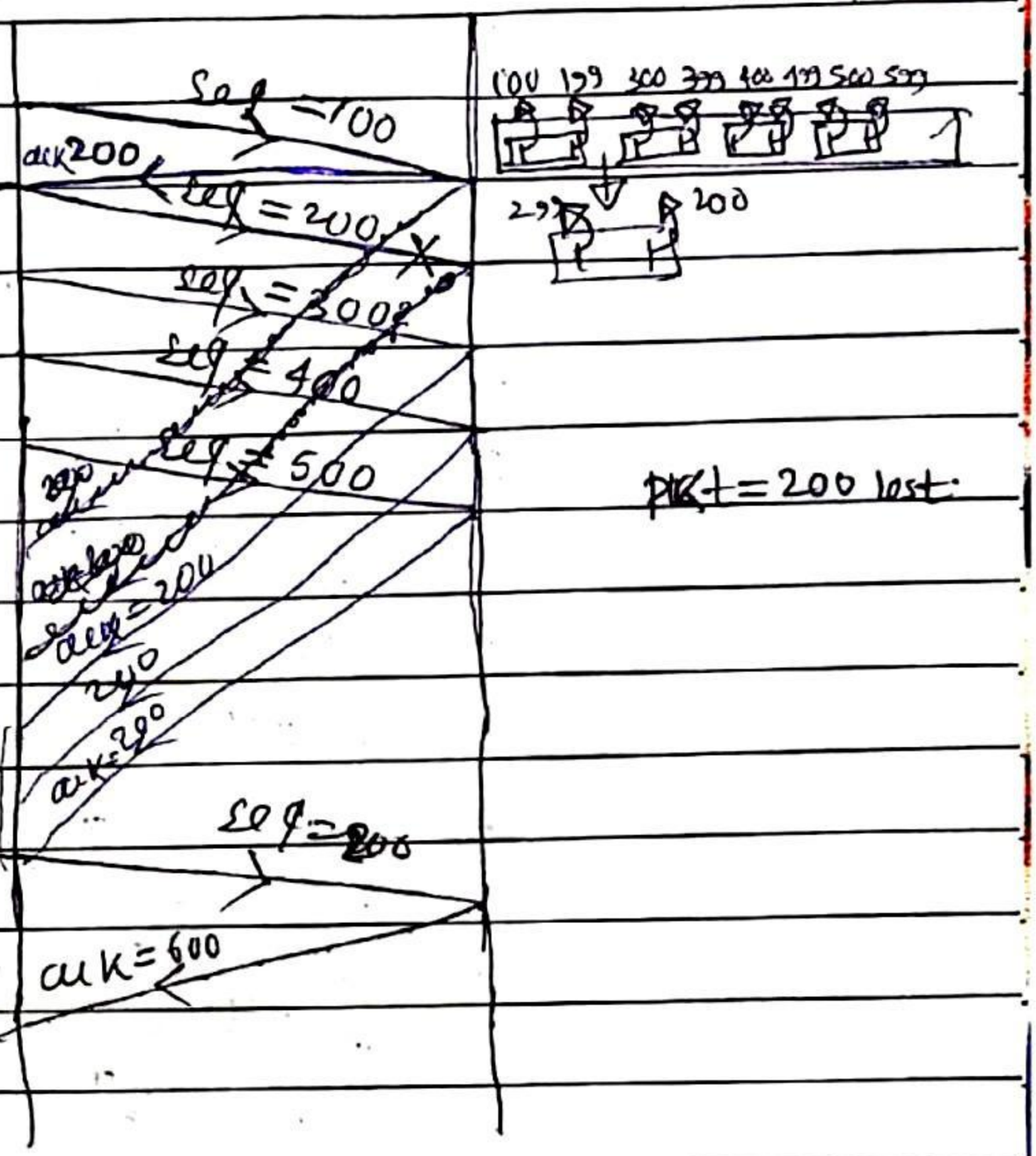
→ $W_S = W_R$	→ Acknowledge are accumulative
→ out of order	
75%	25%

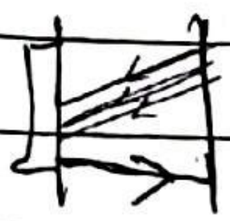
- Retransmission after time out :-



Retransmission after 3-duplicate ack : Early retransmission

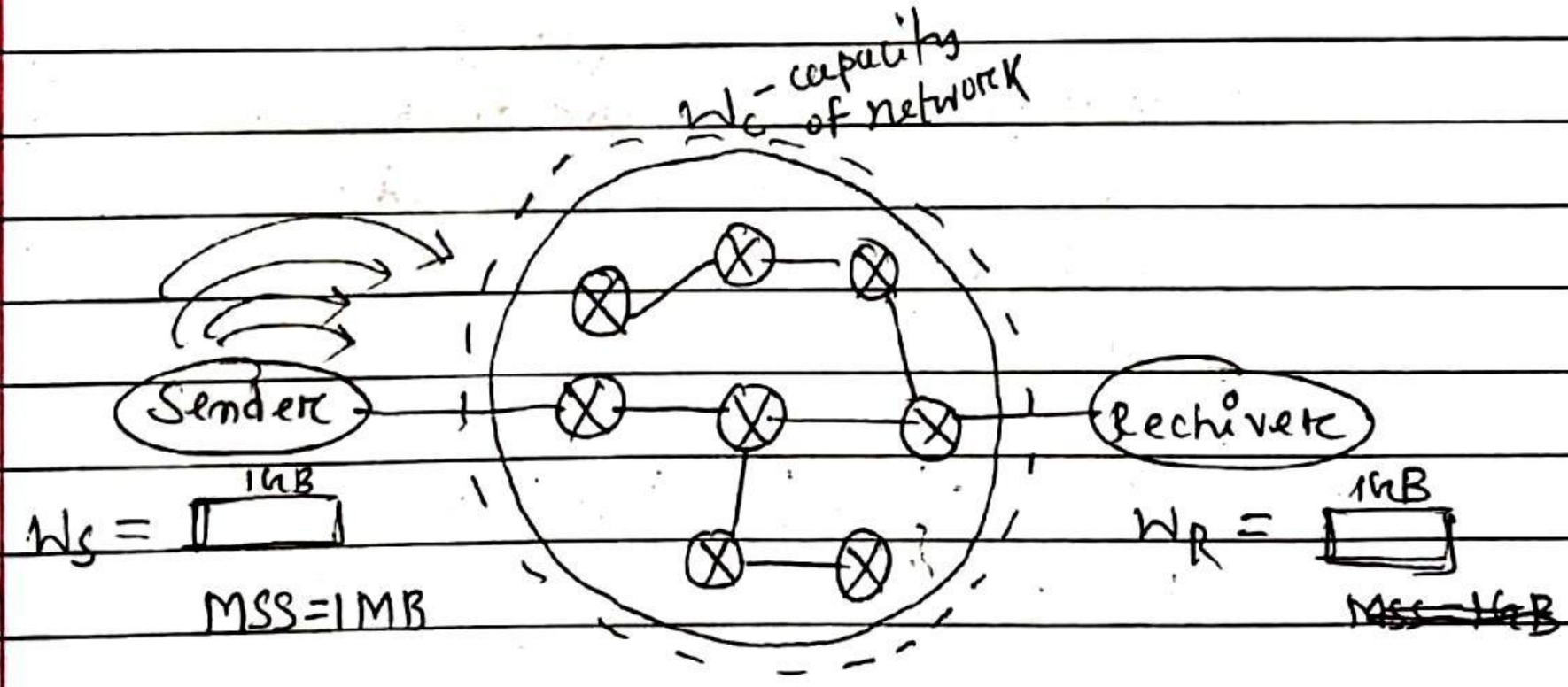






To
Time
out
(TO)

- TCP Congestion control —



No. of segment = 1024

~~before send the data, he should know capacity of = Wc~~

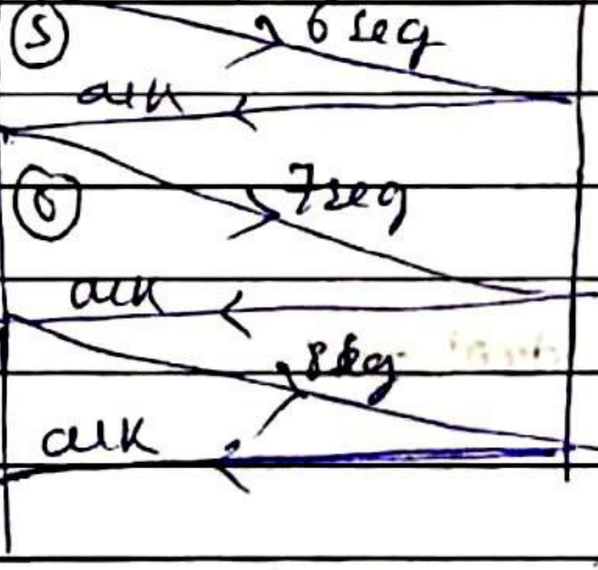
sending data = min (Wc, WR).

1MB 1GB

1GB 1MB

→ TCP congestion control is going to be protective layers for network.

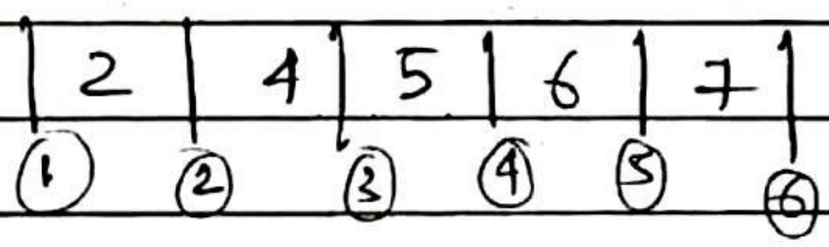
adv window = 8 KB
MSS = 1 KB



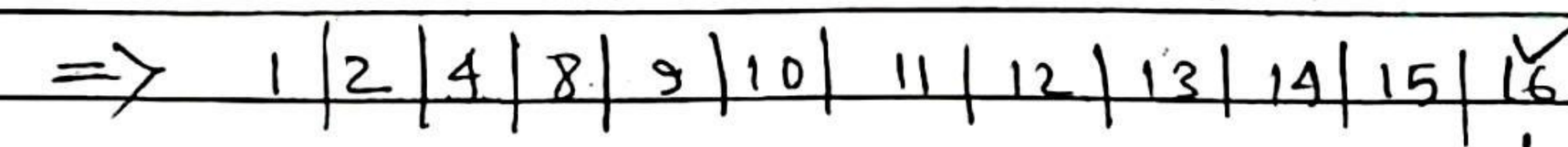
1	adv win = 8 KB.
2	MSS = 1 KB.
4	W_R = 8 segment
5	W_C = 1 segment (starting with 1 segment)
6	congestion window
7	RTT = 1
8	Threshold = $\frac{W_R}{2} = 4$
	WS = 1 segment
	2 seg
	4 seg
	5 "
	6 "
	7 "
	8 "

(RTT)

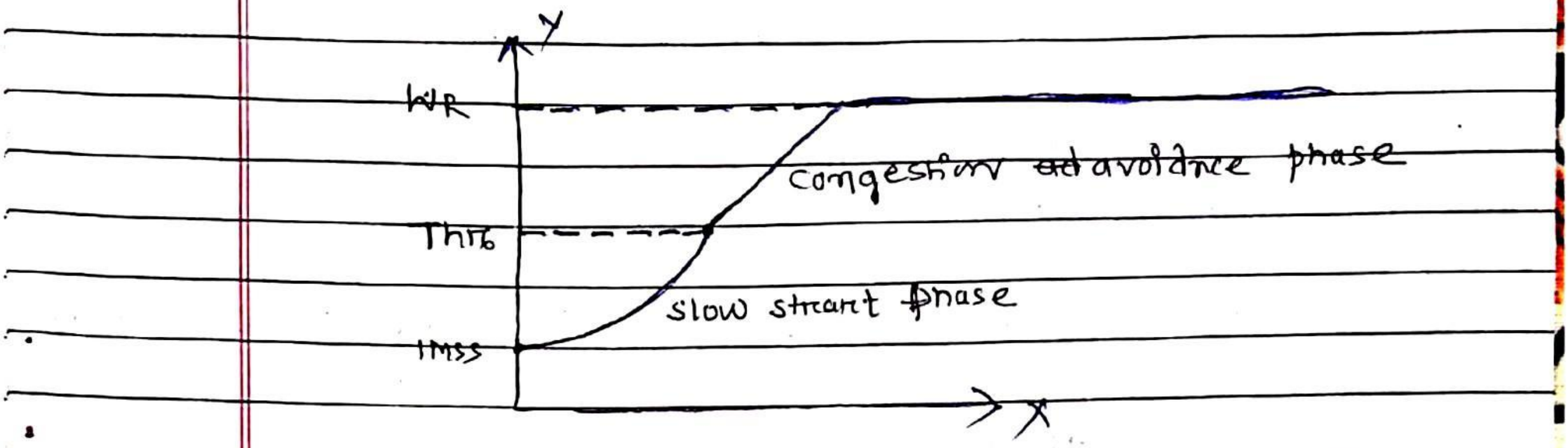
Q. After how many round trip time will be reach the max sender capacity?
⇒ Answer : 6



Q. What will be RTT = ? (to reach max sender capacity)
when, Threshold = 8 / W_R = 16



Answer = (11) — RTT



Congestion control algorithm —

(i). Slow start phase.

(ii). congestion avoidnce phase.

(iii). congestion Detection phase.

way to detect congestion { (a) Time out. (congation severe) newTh = ½ W_L
(b) 3 duplicate ack. (congation mild)
(c) ICMP source quench. x

ex: (with example)

Wp = 64KB

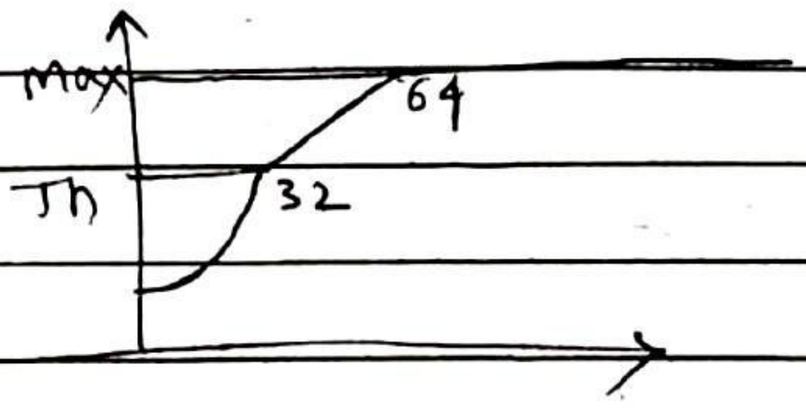
WSS = ~~1KB~~ 1KB

WR = 64 MSS

Th = $\frac{64}{2}$ = 32MSS

1 2 4 8 16 32 33 34 64

SS phase CA phase



congestion detect

(a) TO → newTh → ½ W_c → (ss phase)

(b) 3 dup acks → newTh → ½ W_c → (CA phase)

atlantis

Date

Page

W_R = 64KB (congestion detection)

MSS = 1KB

W_R = 64MSS

Th = 32MSS

Th=17

1 2 4 8 16 32 33 34 1 2 4 8 16 17 18

Time out

Th=10 Th=6 Th=4

19 20 10 11 12 1 2 4 8 6 7 8 4 5 6 7 ... 64

3 dup ack TO 3 dup ack

- TCP Timer Management —

(1) Time-wait timer. → adv (to take care of late pkt)

(2) Keep alive timer. → adv (close idel connection)

(3) persistent timer → used (window size = 0 is advertised)

(4) acknowledgement timer → (used cum ACK, PB ACK)

(5) Time-out timer.

Time out timer —

(1) Basic algo.

(2) Jacobsons algo.

Static - TO

Dynamic - TO

Jam

TO = 10ms

Gpm

PKT = segment

TO timer ↑
→ Time waste if packet lost

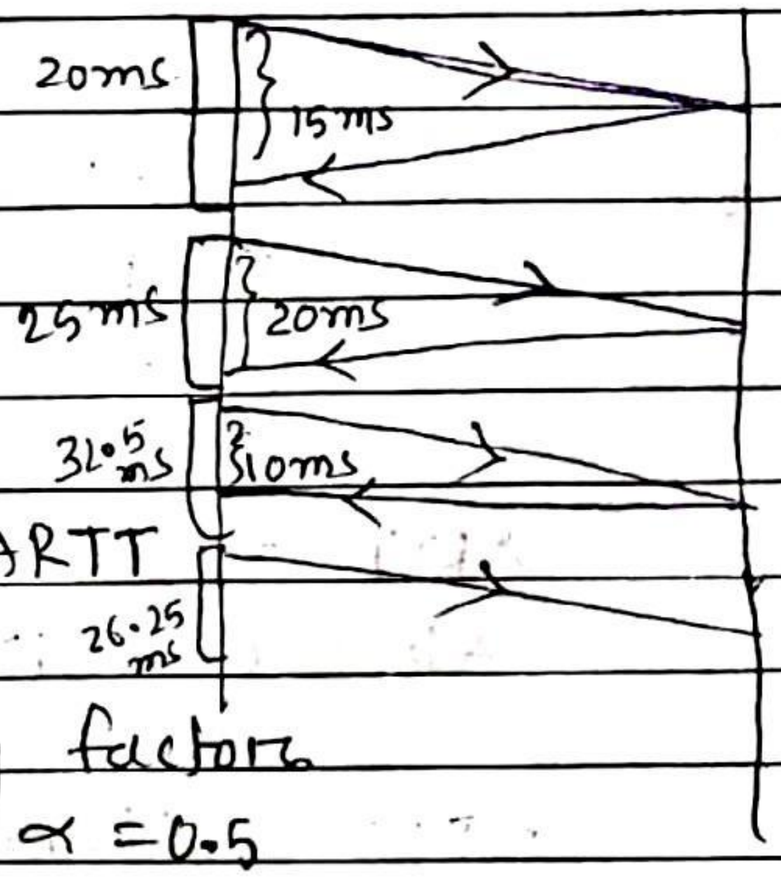
TO - timer ↓
→ Less recovery

• Basic algorithm for timeout timer —

IRTT
→ Initial Round trip time

IRTT = 10ms

TO = 2 * RTT
= 20ms



ARTT = 15ms

• TO
→ Time out

NRTT = α * IRTT + (1-α) ARTT

ARTT
→ actual RTT

α → Smoothening factor

$0 \leq \alpha \leq 1$ α = 0.5

• NRTT
→ Next RTT

= 12.5 ms

TO =

IRTT = 12.5 ms

TO = 25 ms

ARTT = 20ms

NRTT = α IRTT + (1-α) ARTT. (α = 0.5)

= 16.25 ms

IRTT = 16.25 ms

TO = (2 * 16.25) = 32.5 ms.

ARTT = 10ms

NRTT = 130.125 ms

IRTT = 13.125 ms

TO = (2 * 13.125) = 26.25 ms.

// dis advantage —

TO = (2) * RTT
↳ jacobson modify it.

atlantis

Date

Page

• Jacobson's algorithm —

ID → Initial Deviation

① IRTT = 10ms

ID = 5ms

TO = 4*D + RTT

= 4*5 + 10

= 30ms

ARTT = 20ms

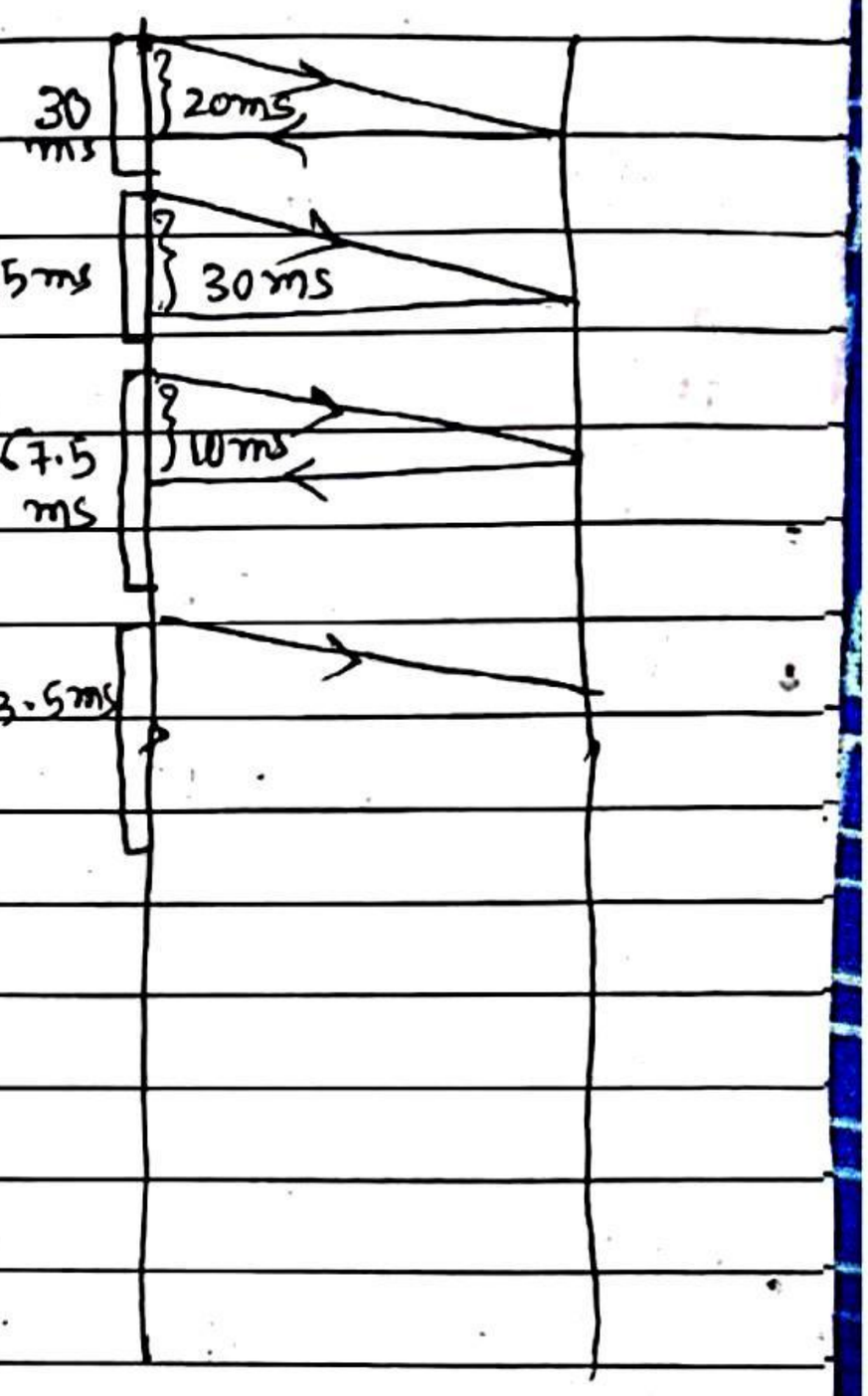
AD = 10ms (|IRTT - ARTT|)

NRTT = ∝(IRTT) + (1-∝)(ARTT)

= 15ms (∝ = 0.5)

ND = ∝(ID) + (1-∝)(AD)

= 7.5ms



② IRTT = 15ms

ID = 7.5ms

TO = 45ms

ARTT = 30ms

AD = 15ms

NRTT = 22.5

ND = 11.25

③ IRTT = 22.5

ID = 7.5

TO = 67.5ms

ARTT = 10 ms

AD = 12.5

NRTT = 16.25

ND = 11.875

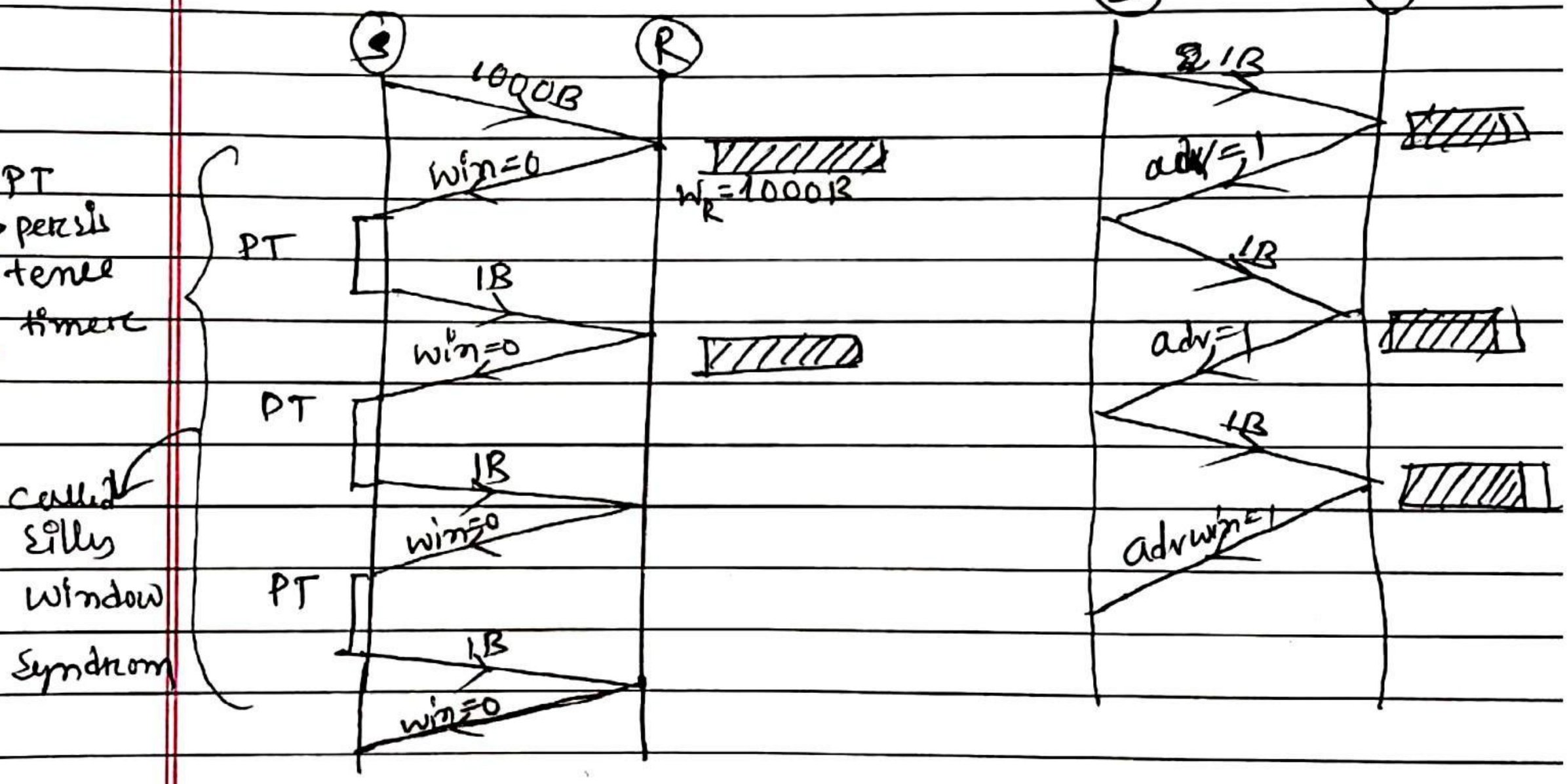
TO = 4*D + RTT

= 63.75

Karn's modification on Basic & Jacobson's algo-

→ Karn's ~~so i a~~ modification is said that, if the acknowledgement is ~~goes a~~ cross ~~that~~ the range of Time out timers then next time Resend the segment with Double TO. (2*TO).

- Silly window syndrome —



↳ Nagles algorithm —

→ If the sender is very slow then Transport layer should not send 1B of Data Transport layer should wait for 1 RTT.

↳ Clarck —

→ Destination should not advertised 1B until it should wait untill it can send atleast 1 MSS or 1/2 of Buffer.