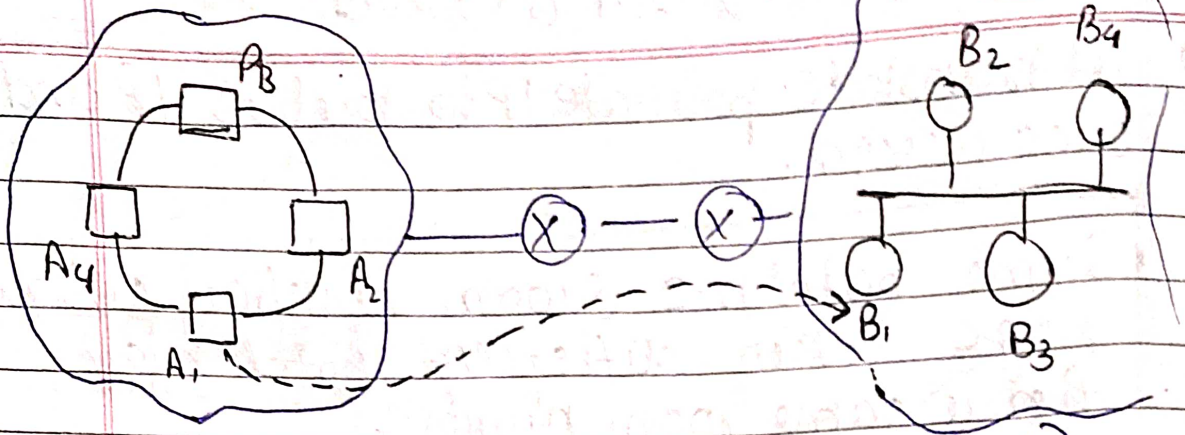
# Transport Layer




1. End to to End Delivery (port to port)
In machine $A_1$ there is possibility that more than 1 parts are processing for Example → multiple tabs are open in a System $A_1$. So Transport layer make sure that message received by Correct port it is port to port delivery.

2. Transport Layer uses TCP and UDP. int Network Layer messages and send with out any guarantee. so it was less reliable so So if we want to send message with a proper reliability transport layer's TCP (transfer control protocol) is used

| 4 messages send | → (order delivery) | 4 messages received |
|---|---|---|

No Loss of data. TCP (Connection oriented)

3 Error Control :-
It uses checksum method. also used by TCP.

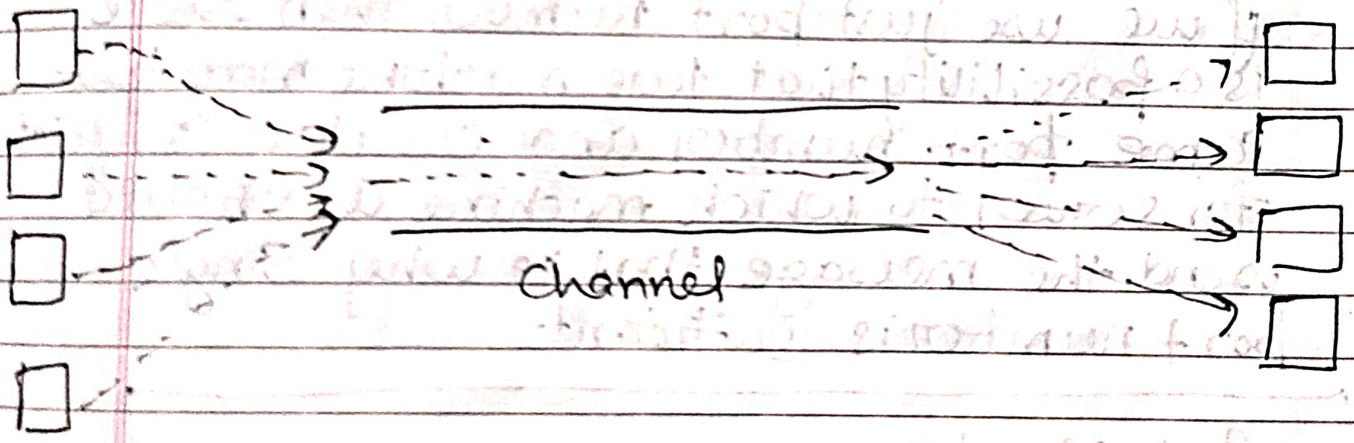
4. Congestion Control :-
AIMD → Additive inverse multiplicative Decrease that is method which is use to

control the congestion

5. Flow Control:-
Stop & wait, Go-Back N,
Advertising the window method is used when receiver told the capacity of itself. so that the sender will send accordingly.
Size of message and number of messages to be sent if it is not done then congestion can happen

6 Segment:-
divide the message into segments and then send it to Network layer

7. Multiplexing / Demultiplexing:-
Multiple Applications do sending data simultaneously so Transport layer will not create Network seperately it perform multiplexing and vice versa will happen at end



# Socket Address

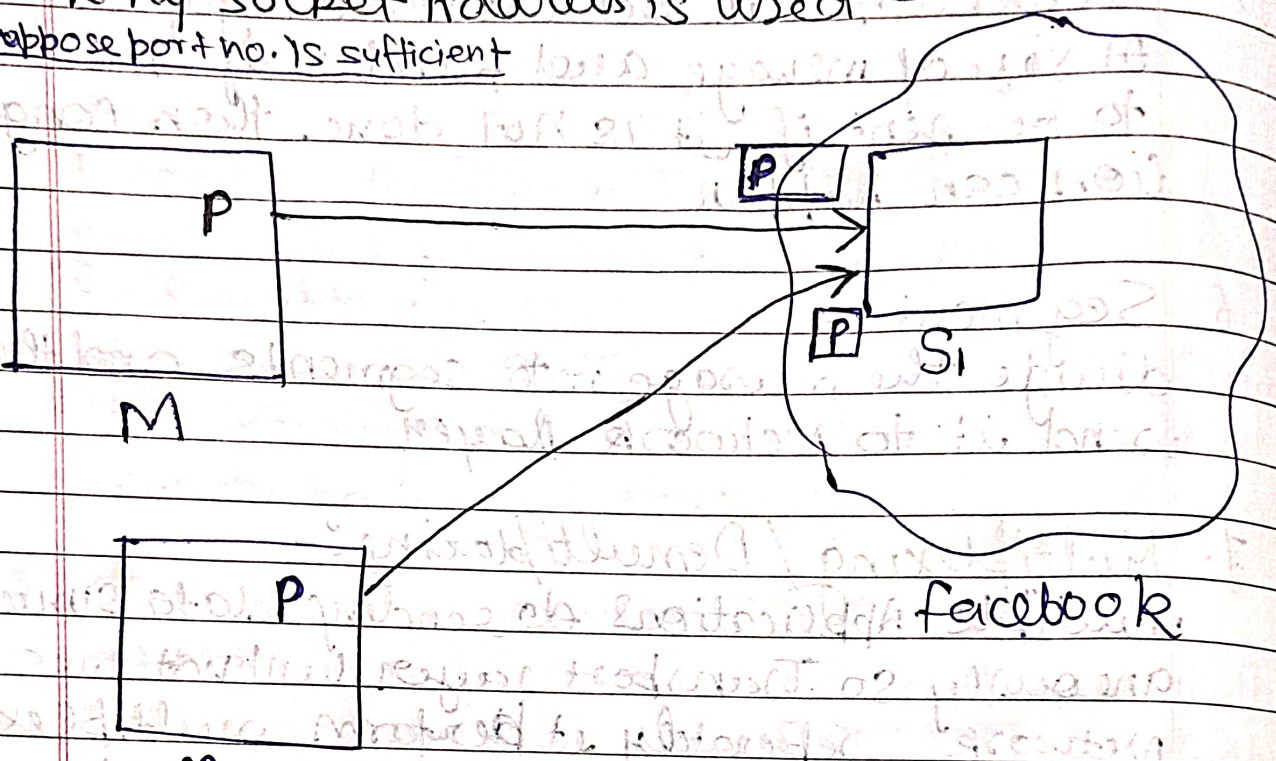
IP Address + Port Number

32 bit + 16 bit = 48 bits

used to ~~Conn~~ uniquely identify the connection

Why Socket Address is used

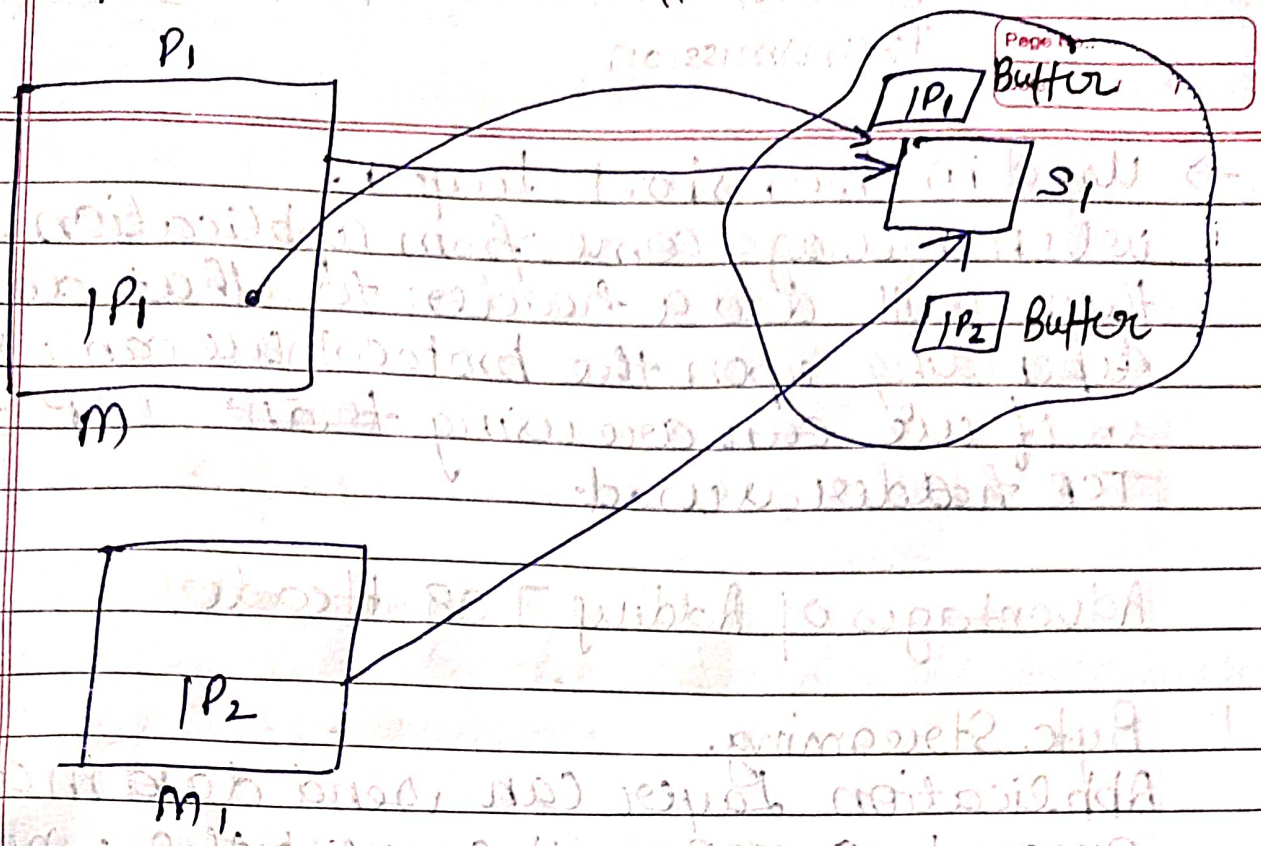
# Suppose port no. is sufficient




if we use just port number then there is a possibility that two machine may get same port number and create a conflict for server to which machine it should send the message that is why Single port number is sufficient

Port Number
16 bit
0 - 65535 ] - Total
0 - 1023 ] - well defined (Http, Smtp)
1024 - 49151 ] - reserved by Company.
49152 - 65535 ] - local port Number

Is Only IP address is sufficient




if a single machine send message from two different message then again ~~with~~ message will send from single IP address So IP address is also not sufficient alone.

∴ we need both combination of IP address and port address. which is known as socket address

# TCP (~~Transport~~ Control protocol)
Transmission

Page No.:
Date: / /

→ Used in transport layer.
When message come from application layer then will add a header to that data depending upon the protocol we can using so if we ~~can~~ are using ~~tcom~~ TCP then TCP header is used.

Advantages of Adding TCP Header

1. Byte Streaming.
Application Layer can send data in any amount so TCP will convert the whole data in bytes and divide them into segment. (segment is a collection of bytes)

2. Connection Oriented:
the connection is created and no data should be lost ~~or~~ it uses 3 way handshaking protocol to establishing a connection

3. Full duplex:-
One A and B are connected then that connection will work as full duplex which means both A and B send and receive messages a simultaneously.

4. Piggybacking:-
Whenever we send a message the receiver should send the acknowledgement and if that acknowledgement is send along with data then it is called piggybacking
GBN and SR used to send data

4 Error Control:-
If there is some change occur in data then receiver should atleast know that there is an error.

5. Flow Control:-
if we cannot control the flow then it receiver's buffer is overflowed and data will be lost.

6. Congestion Control:-
It take care of the capacity of receiver as well as Network.

TCP header size = 20-60B

| Source port 16bit | | | | | | | | Destination port 16 bit |
|---|---|---|---|---|---|---|---|---|
| Sequence Number 32 bit | | | | | | | | |
| Acknowledgement No. 32 bit | | | | | | | | |
| HLEN 4bit Scale of 4 bits x4 | | 6 | U R G | A C K | P S H | R S T | S Y N | F I N | Window Size 16bit |
| Checksum 16 bit | | | | | | | | Urgent Pointer |
| options and padding - 40 bytes | | | | | | | | |

is HLEN = 0100 is valid?
No because 0100 = 4
Scale by 4 = 4x4 = 16 < 20 so it is invalid.

URG → If some data is urgent then set URG as 1

ACK → If we are sending acknowledgement then set ack as 1

PSH → [Push] (If the sender wants to send the data to receiver without filling buffer)

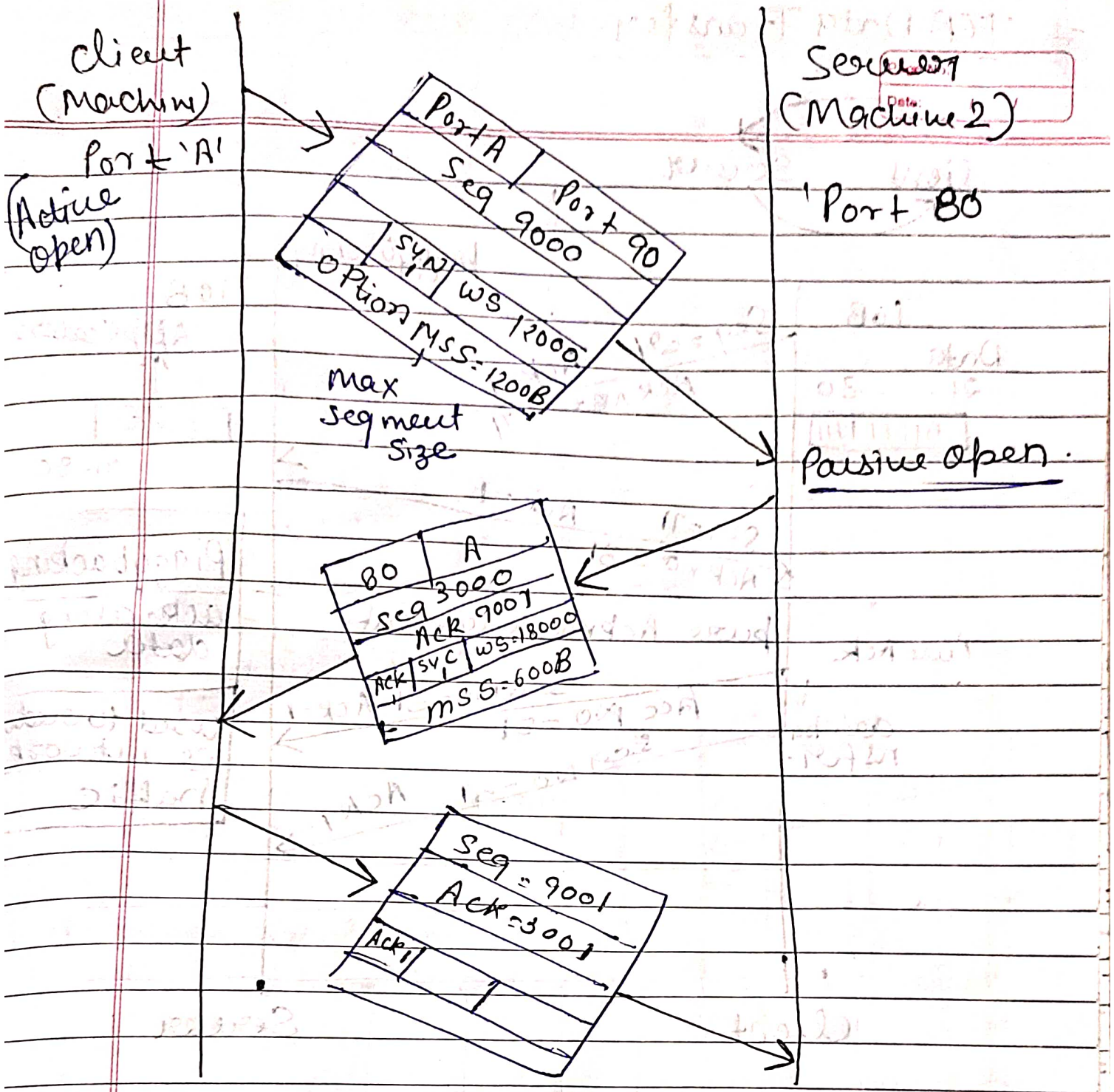
RST → [Reset] to reset connection

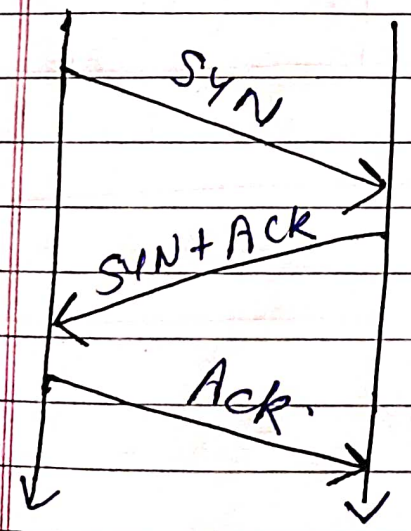
SYN → [Synchronization] if client wants to create a connection to server set syn to 1

FIN → [finish] to terminate connecⁿ. set FIN := 1

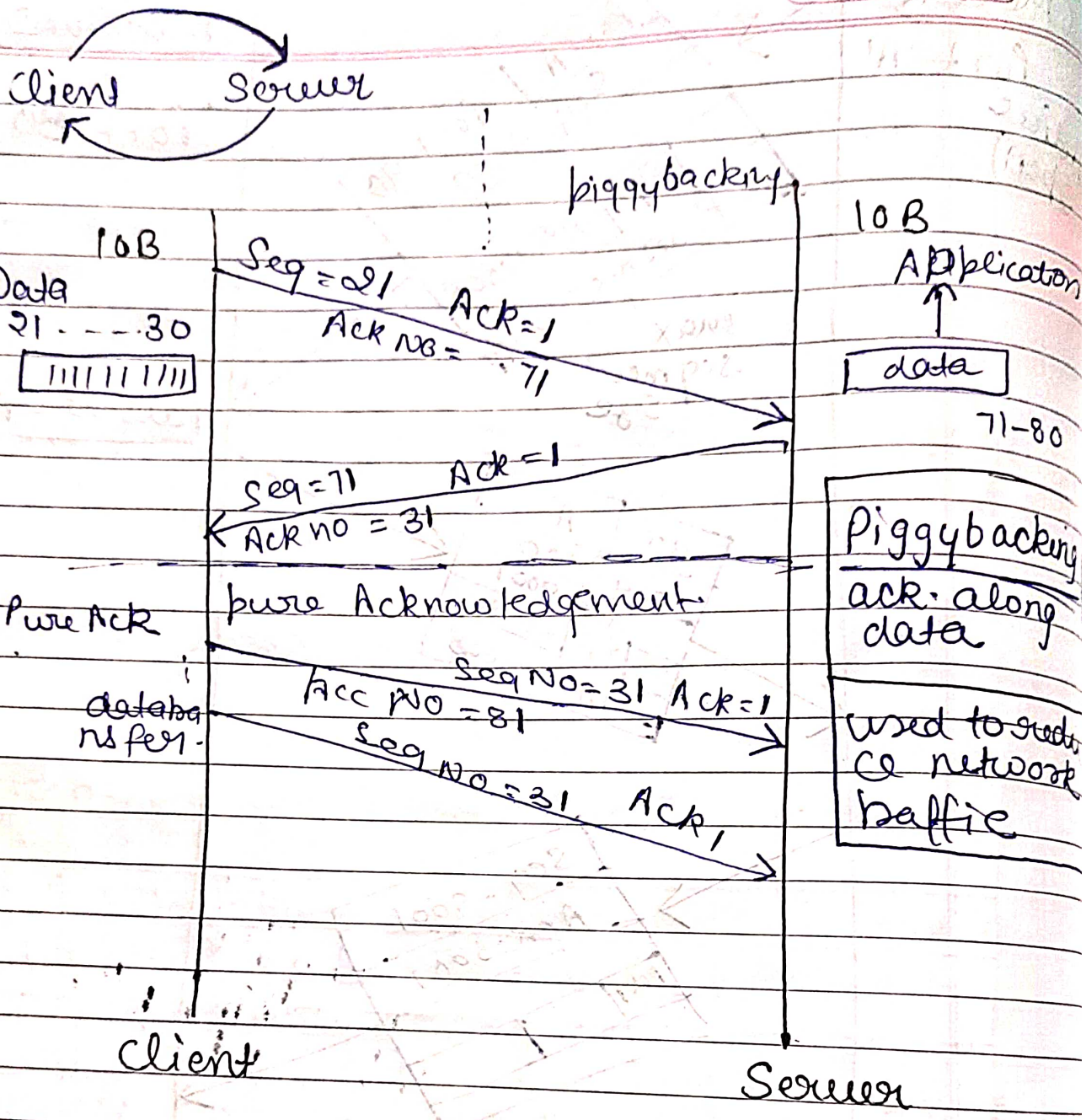
# TCP Connecⁿ Estblishment

if A estabishing connection with B, then A will send some information like buffer, CPU, bandwith should reserve and B will also send some information to ~~send~~ reserve buffer, CPU, bandwidth.

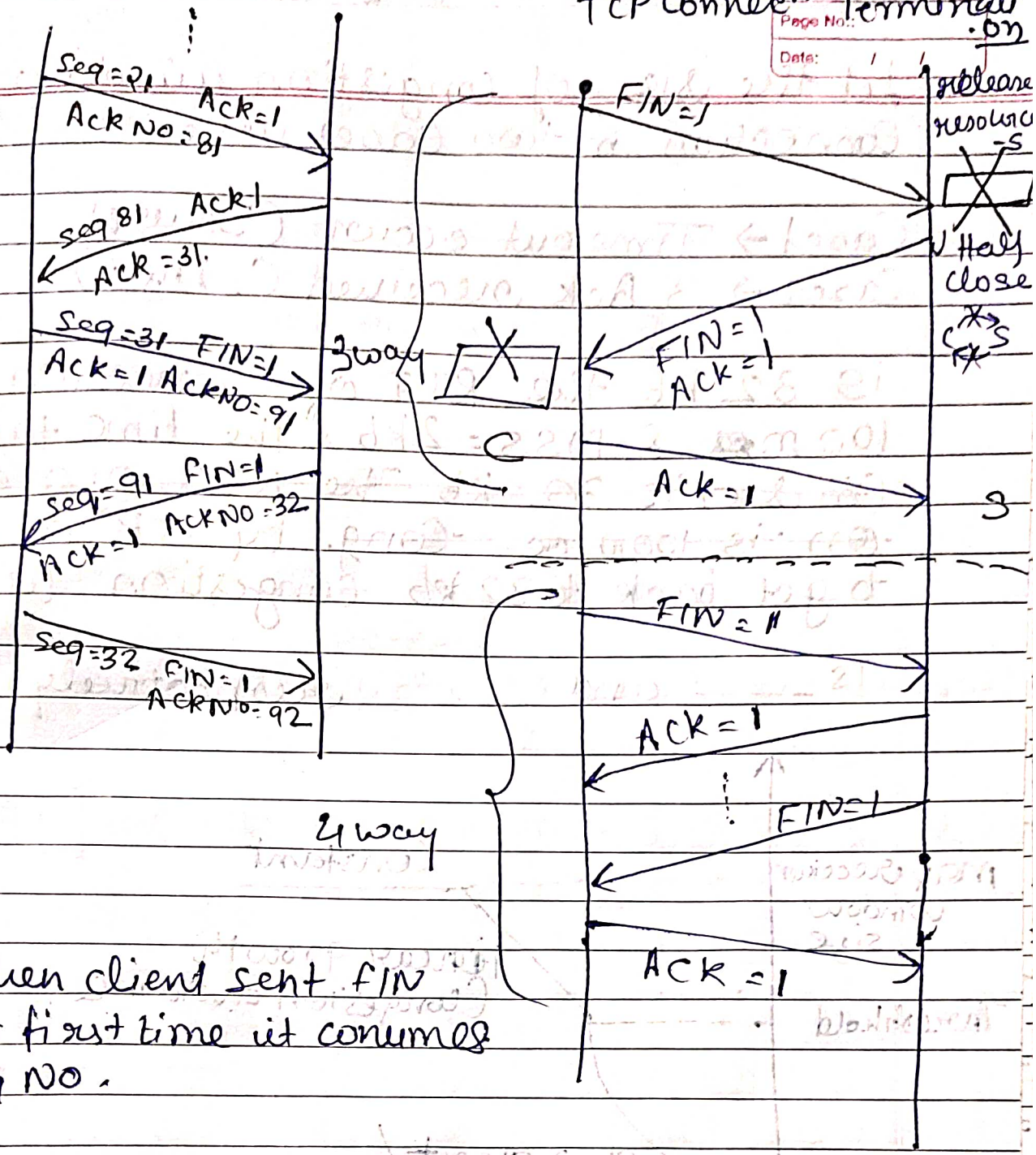

Client (Machine)
Port 'A'
(Active open)
Server (Machine 2)
Port 80
Port A
Port 80
Seq 9000
SYN
WS 12000
Option MSS = 1200B
Max segment Size
Passive open.
80
A
Seq 3000
Ack 9001
Ack
SYC
WS=18000
MSS=600B
Seq = 9001
Ack = 3001
Ack


SYN
SYN+ACK
Ack.

# TCP Data Transfer



# Connection Termination




1. When client sent FIN for first time it consumes Seq NO.

2. When server send FIN+ Ack=1 + consumes SeqNO

3. End, when client send Ack then seq.NO is not consumed

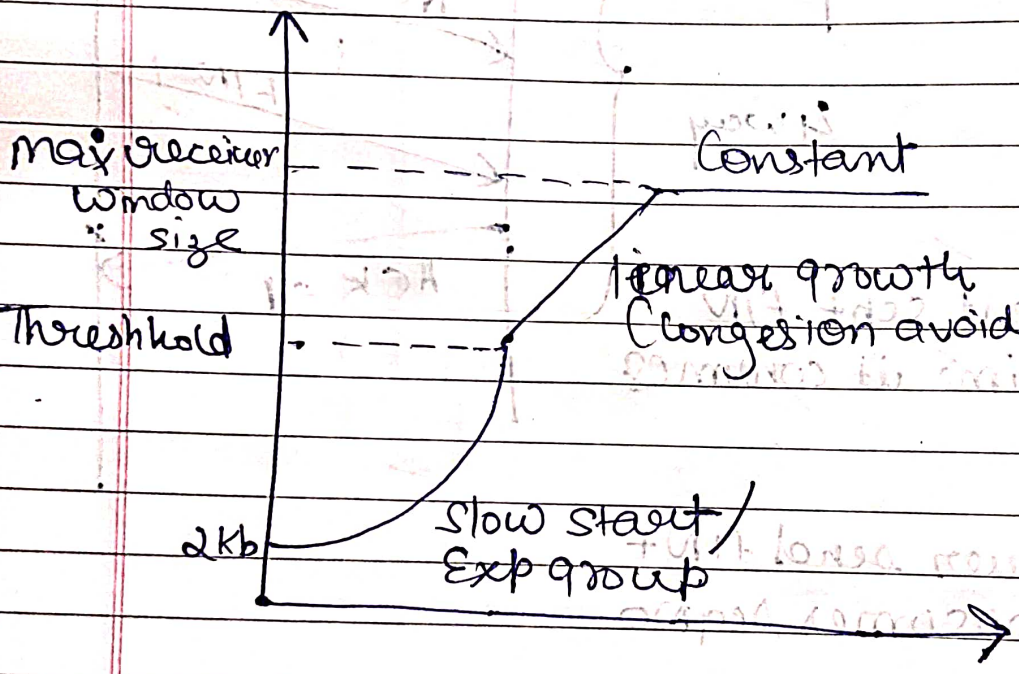
# Numerical on Congestion Control

Page No.:
Date: / /

Let the size of congestion window of TCP Connection in two cases when

Case 1 → Time out occur (severe)
Case 2 → 3 Ack received (Light)

is 32 kb the RTT of connection is 100 ms & MSS = 2kb. The time taken ~~(in sec) is 32 kb The RTT of a connecti on is 100sec Cong~~ by TCP connection to get back to 32 kb Congestion window is ____ and ____ respectively




Case 1

Step 1 Timeout Threshold ⇒ $\frac{32}{2} = 16$

32kb ↑ 2 4 8 16

When timeout occurs protocol says that go to slow start / exp growth phase and go till threshold

Threshold = Size of congestion window in MSS

Page No.:
Date: / /

(2) Now Reached threshold so grow linearly.

32↑ 2 4 8 16 18 20 22 24 26 28 30 32

12 times.

time taken every time = 100.

12 X 100 = 1200 msec.

Case 2

(1) donnot go to slow start phase.

(2) directly start with phase 2 (Linear growth)

32↑ 16 18 20 22 24 26 28 30 32

9 X 100 = 900

Ans 1200, 900

# User Datagram Protocol

→ His connectionless (Unreliable)

C ⟷ S

Payload
Data

→ No Order.

| Source Port 16 bits | Destination Port 16 bits. |
|---|---|
| Length 16 bits. | Checksum 16 bits |

UDP Header
8 Bytes fixed.

Length $2^{16}-1 \Rightarrow 65535$

| # | Payload |
|---|---|
| 8 | 65527 |

← 65535 →

→ Checksum is used for Error Control

Checksum = UDP Header + UDP Data + Pseudo header of IP

# UDP Applications

1. Query Response Protocol (One-request one reply (DNS)

2 Speed (Online games, Voice over IP)

3. Broadcast and multicast [RIP]

4. Continuous streaming [Skype, Youtube]

# Differences between TCP and UDP

| Transmission control protocol | User datagram protocol |
|---|---|
| 1. Connection oriented | connectionless |
| 2. Reliable | Less Reliable. |
| 3. Ordering | No ordering. |
| 4 Error control is mandatory | Error control is optional. |
| 5. Slow transmission | Fast transmission |
| 6 more overhead (20-60B) | Less overhead (8B) |
| 7 flow control and congestion control | No flow & Congestion control |
| 8 Ex → HTTP, FTP | DNS ← Ex<br>BOOTP<br>DHCP<br>RIP |