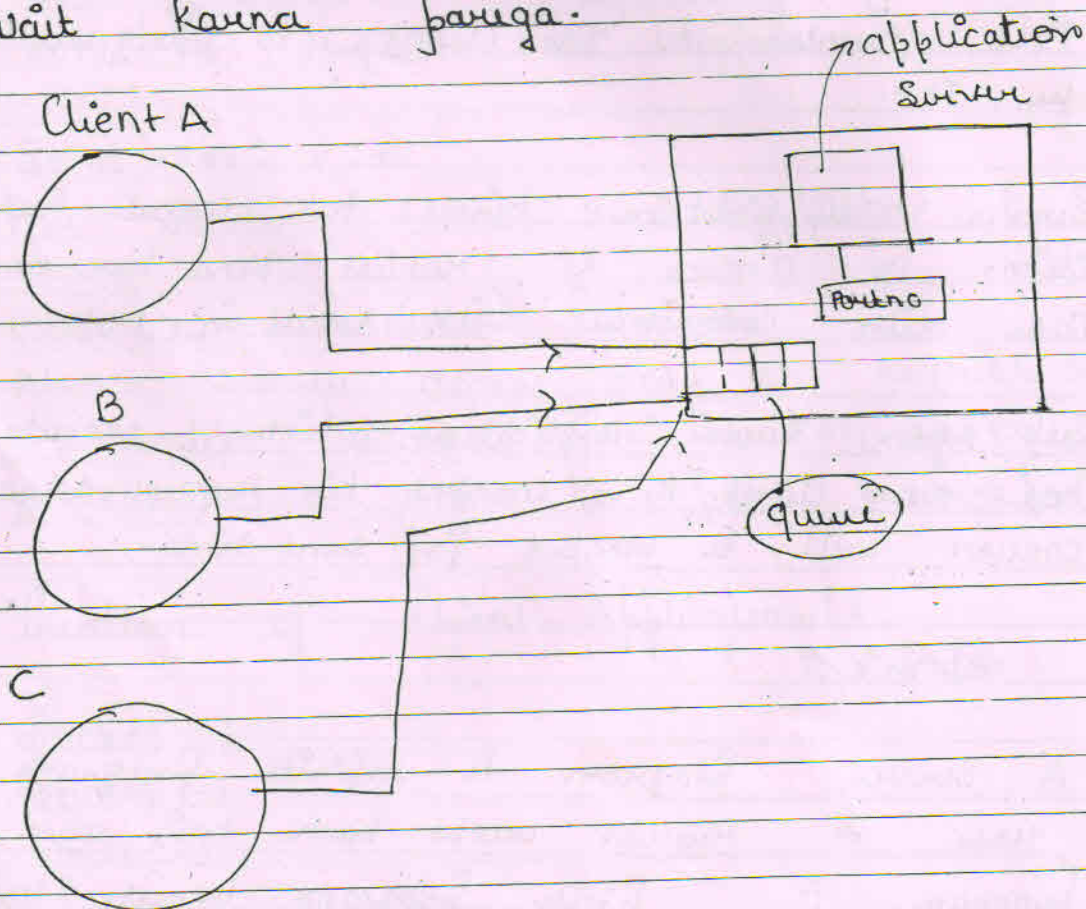


Server Char type k hosakte hai-

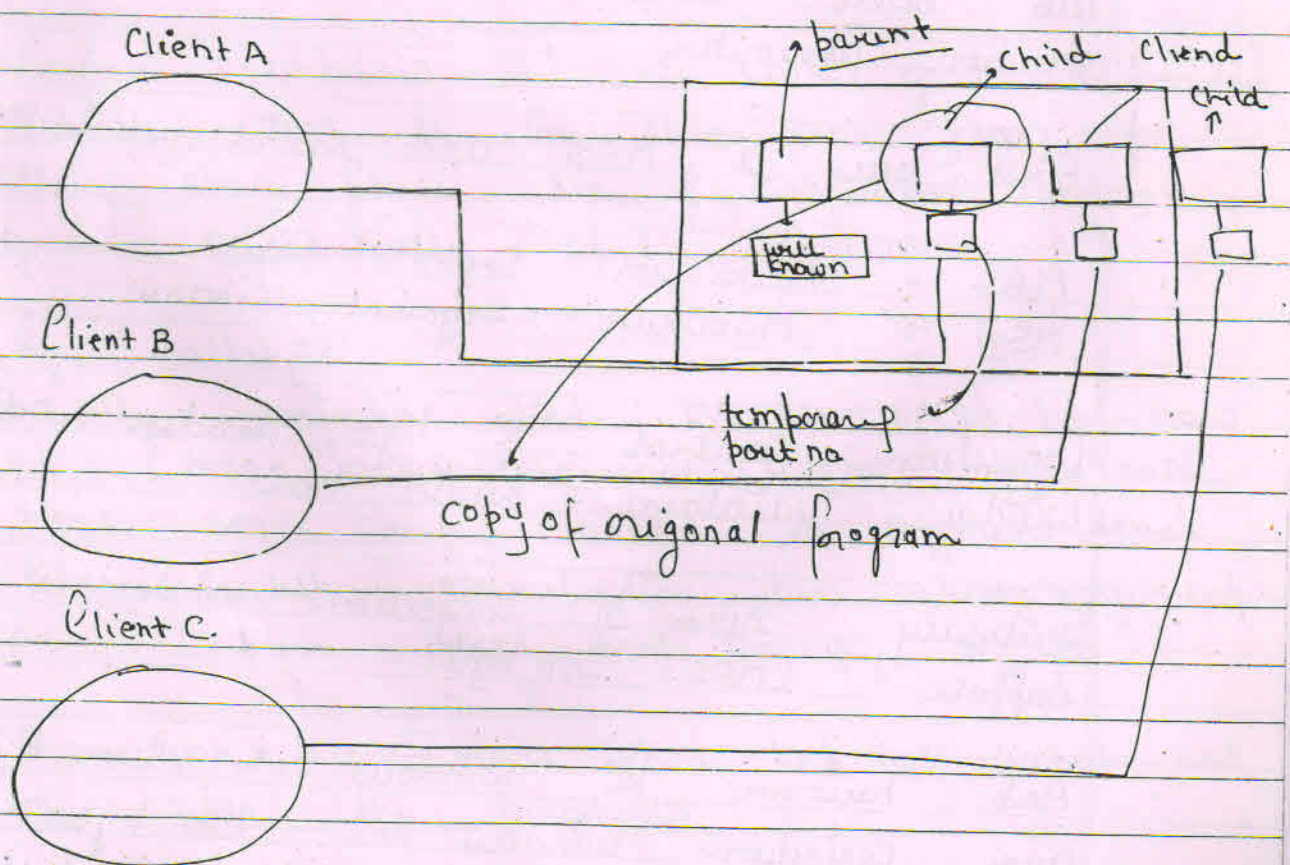
- 1) Connectionless- iterative → Normally UDP used
- 2) Connectionless- concurrent
- 3) Connection~~less~~-oriented iterative
- 4) Connection-oriented ~~less~~ concurrent:- → ~~Not~~ Normally TCP/SCPT uses..

• Connectionless iterative mein kya hota hai jab tak ek client pura apna kaam nahi karke hai jab tak dusre client ko wait karna parrga.



• Connection-oriented concurrent mein kya hota hai, ki saare client ek saath application (Server) se interact kar sakte hai.

- For example agar client A ki request aaye server par, toh hum uski fork number original application ki exact copy create karunga aur usse a temporary port number assign kar degey. Now jaise kisi client ki request aayi toh hum usse well-known port no hoga toh hum us application k copy banayenge aur usse kuch temporary port number degey aur ~~usse~~ server ko hum temporary port number send kardegey.
- Is tarike se hum concurrent access karwa sakte hai.



Connection-oriented concurrent server.

- * Fork() ki help se duplicate copy create kar sakte hai.

TCP Congestion Control:→

AIMD

classmate

Date _____

Page _____

243

#

- Network Layer ko congestion control karna chahiye but according to OSI model, Par TCP/IP model par IP-layer congestion control nahi karta hai

- In TCP/IP Model, transport layer performs congestion control.

Now How TCP handle congestion??

- • Joh, jab ^{connection} ~~congestion~~ establish hota hai, tab both sender and receiver agree on two parameters.

- First one is MSS and second one is MSW

- MSS → Maximum Segment Size
MSW → Maximum Segment window

- Congestion control k liye sender, congestion window maintain karta hai.

- Initially size of congestion window is 1 MSS
Suppose 1 MSS = 100 bytes.

- Ab receiver ki window toh 1000 bytes ki hai aur congestion window hai 100 bytes toh sender 100 byte bhejega ki 1000 bytes bhejega.

- Sender kitana send karega, sender ki window size kitani hogi??

Sender window size = $\min(C_{win}, R_{win})$

$C_{win} \rightarrow$ Conjection window

$R_{win} \rightarrow$ Reciever window.

• Agar $C_{win} = 5MSS$, $R_{win} = 2MSS$, then
Sender window = $\min(5, 2) = 2MSS$.

• But, Conjection window ki size constant nahi
rahati hai it increases. Conjection window ki
size increase hoti hai.

Ques:- Conjection window ki size kaise grow hoti hai??

Ans- Initially TCP is in slow-start phase. In
slow start phase size of conjection window
grows exponentially up to threshold.

Kya matlab??

Initially conjection window ki size $1MSS$ hai, now
us $1MSS$ acknowledgement aagaya uske baad
 $2MSS$ send karu matlab do segment send
karu uske corresponding two acknowledgement
aayega then $4MSS$ send karu so on

• Now yeh slow-start phase kab tak chalti hai.
Jab upto the threshold.

Initially threshold = $\frac{1}{2} (MSS)$ set karte hai
ko hum

(Multiplicative decrease)

So in our case threshold = 16.

- 16 MSS for aake hamara slow-start phase terminate hojayege.

- Uske baad linear phase ya phir usse ko hum Congestion Avoidance phase kahate hai, start ho jata hai. (additive)

- In additive (Linear) phase, size of congestion window growy linearly.

- So 16 MSS k baad 17 MSS, 17 MSS ka acknowledgement aayega uske baad 18 MSS on --

- linearly grow karta jayega tab tak??

Ans- Tab tak sender detect nahi kar leta congestion

- Congestion kisi bhi phase mein hosakta hai aur agar congestion detect hoia hai toh woh grow hona band kar deta hai.

Ques → How TCP detects congestion??

Ans → Network Layer par toh underlined bandwidth ko dek kar pata kar sakta hai ki congestion hua hai ki nahi.

- Par transport layer par aise kahi information nahi hoti.

- Toh TCP ki transport layer ko kaise pata chalta hai ki congestion hua hai??

- TCP assume whenever there is retransmission occur then it is due to congestion.

- Whenever retransmission occur it sets

$$\text{threshold} = \frac{1}{2} [\text{current cwin}]$$

- Jo retransmission ki hamne do policies padi hai ek toh retransmission due to time out and dursa retransmission due to 3-duplicate acknowledgement.

- Jo actual mein congestion kab hoga, toh actual mein congestion toh time-out k karan hoga.

- Kyoki 3-duplicate acknowledgement policy mein se retransmission hoga toh congestion nahi hoga kyoki humko acknowledgement toh milne ho hi nahai toh kayika congestion.

- * When TCP detects congestion due to time-out it starts from slow-start phase.

- * And when TCP detects congestion due to three duplicate acknowledgement then it starts from linear phase.

- Agar threshold ki value 21.5 hai toh lower bound hai ki upper bound nahi hai

- Jo since TCP/IP toh vendor implemented Model

hai, yeh vendor par depend karuga. Ki 22 lina hai ki 21 lina hai.

• So is problem ka koi standard nahi hai / Standard answer nahi hai.

• This algorithm is known as Additive Increase Multiplicative Decrease (AIMD) Algorithm.

Now Special point \Rightarrow

• Agar 1 MSS send kiya 1 MSS ka ack aayega toh C_{win} ek se increase hojayege. Now aab 2 MSS send kiya toh 2 ack aayenge toh $(2+2)$ MSS hojayege. Now agar 4 MSS kiya aur agar ek hi ack aaya toh C_{win} ek se hi increase hogi i.e 5 MSS in the case of Cumulative ack.

• Here Ack ^{come far} \sim ~~near~~ C_{win} ek se increase hote hai.

Ques: \Rightarrow Let the size of congestion window of TCP connection be 32 KB when a timeout occur. The Round trip time of the connection is 100ms and $MSS = 2KB$. The time take (in ms) by the TCP connection to get back to 32 KB congestion window is -

Ans \rightarrow

$$LMSS = 2KB$$

Now convert all the things in MSS.

$$C_{win} = 16MSS$$

(Slow start phase)

Now at point timeout occur so,

$$threshold = \frac{1}{2} \times \text{Current } C_{win} = 8MSS$$

1 2 4 8 9 10 11 12 13 14 15 (16)

Trans
mission

1 2 3 4 5 6 7 8 9 10 11 (12)

$$12RTT = (12 \times 100) ms = 1200ms.$$

Ans: Consider the effect of slow-start phase on a line with $RTT = 10ms$ and no congestion. The receiver window is $24KB$ & $MSS = 2KB$. How long does it take before the first full window send.

Ans
6

$$1MSS = 2KB$$

$$threshold = \frac{1}{2} \times [24]KB = 12KB = 6MSS$$

Transm
→
round

1 2 4 6 7 8 9 10 11 12
0 1 2 3 4 5 6 7 8 9

$$\text{Total Time} = 10 \times 10ms = 100ms.$$

Round 1 LMSS
Round 2 2MSS
Round 3 4MSS

Round 4:- 8MSS
3 Round 300ms

3-ack ka new fast threshold se start hoti hai congestion window.

classmate

Date _____

Page _____

249

Ques:→ Assuming TCP has a initial threshold value = 8
The TCP sender detect 3 duplicate ACK at the end of transmission Round 7 and time out at the end of 12 Round. What is the number of transmission Round require to transfer 50 data segment?? (suppose transmission Round start from 1)

Ans

↓

Threshold value = 8

	1	2	4	8	9	10
Transmission Round	1	2	3	4	5	6
	11	5				
	7	(8)	9	10	11	12

Now at 7 Round, threshold = $\frac{1}{2} \times 11 = 5$

Now 8th transmission min 50 data segment send kar sakta hai $(1+2+4+8+9+10+11+5)$.

TOP-2 TIPS →

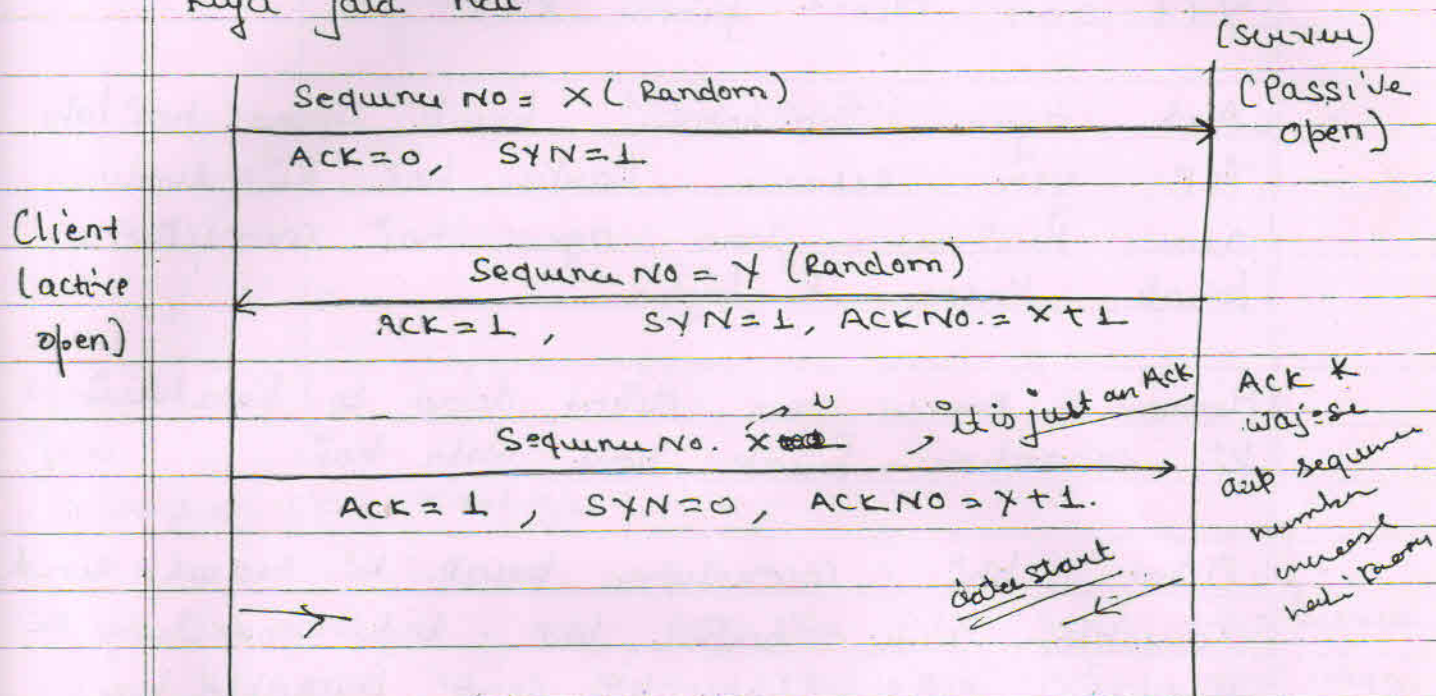
CONNECTION ESTABLISHMENT:-

To establish the connection TCP make use of three way handshaking.

2416 Connection establish karne k. liye client initialise leta hai.

- Any ack segment if carry no data, ~~consist~~
no sequence number.

- Jab connect() function call hoga hai toh ek packet send hoga hai client se server ki cur.
- Woh packet connection ki request k liye send kiya jata hai.



- Har packet ka sequence number hota hai toh first segment ka sequence number is assigned ~~Har~~ Randomly.

- $SYN = 1$:- Syn bit 1, it shows it is the request for the connection.

- Server aur client dono k first segment k sequence number randomly assign hote hai.

- Sabse pahale joh client send karega umme kahi data nahi hai because woh connection ki request send kar raha hai us segment mein, toh server acknowledgement X+1 bhejega. Aur hamne pata hai ki byte number starts from