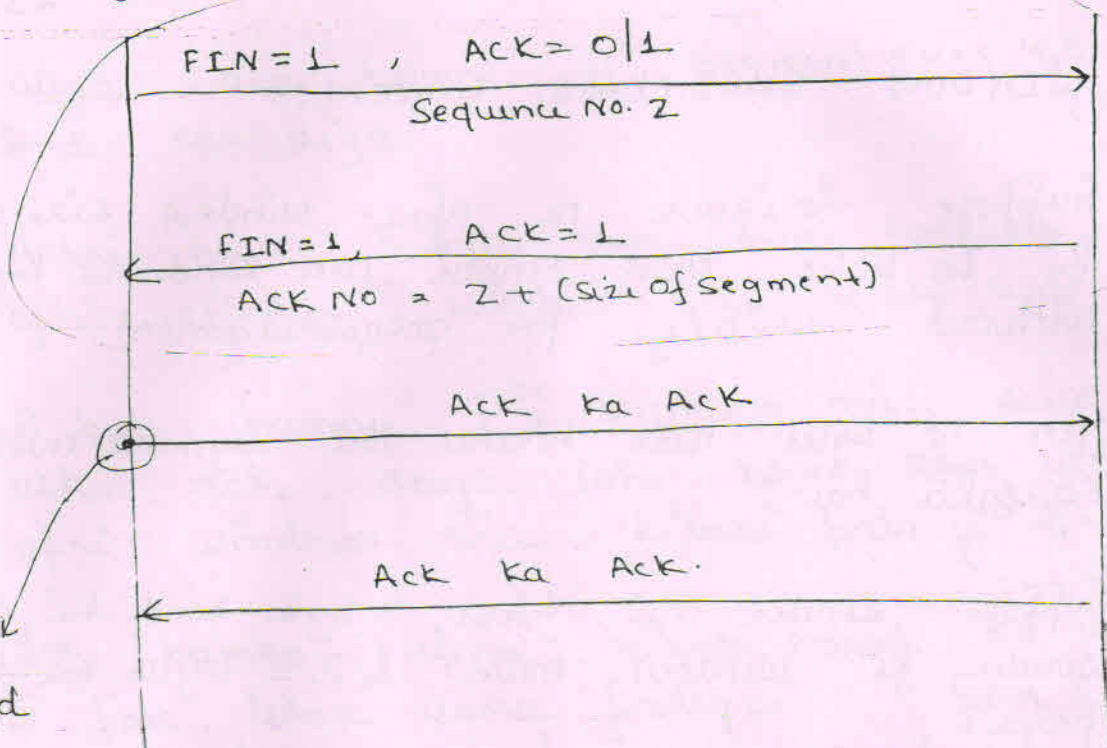


Some Random number and that Random number is $X+1$.

- Server will start byte numbering from $Y+1$ and client from $X+1$.
- Aab agar connection break karna hai toh TCP yeh assume karta hai ki sender aur receiver dono agree hai connection break karna k liye.
- Means server aur client dono ko pata hai ki connection - break hone wala hai.
- Client bhi connection break ki request send kar sakta aur server bhi. but connection ki request sirf client hi send kar sakta hai.
- * Now connection - termination k liye TCP make use of 4-way handshaking.
- # FIN: Connection termination k liye client or receiver jo segment send karega usme $FIN = 1$ hoga.
 - Us segment mein data bhi ho sakta hai.

Half-closed client request 252

Client



afere end
fare

connection close karlega.

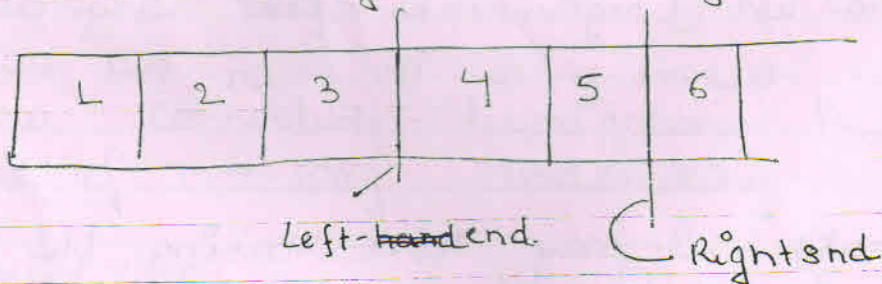
- Tab client ne Ack ka Ack send kiya suru ko aur us segment ka kahi reply nahi aya then it assume at its own end that connection is closed.

• kuch books 3-way hand shaking bhi kabati hai.

- RST aur FIN mein yeh difference hai ki FIN buffer mein joh hai, usko transfer karke connection break karta hai fare RST, bit jaise hi on hue wahi connection break hojayege.

WINDOW SHRINKAGE CONCEPT:→

- Suppose Receiver ne apni window size advise ki 10 bytes then sender can send 10 bytes without waiting for acknowledgement.
- Yen 10 bytes hum kitare bhi segment mein send kar sakte hai.
- Suppose sender ne first 3 byte send ki, abhi sender ki window mein 1, 2, 3 bytes ka data hoga.
- Now aab next segment mein 4th, 5th byte send kar diya aur 3rd ka acknowledgement aagaya, toh uska left-hand end 4 par aagaya.



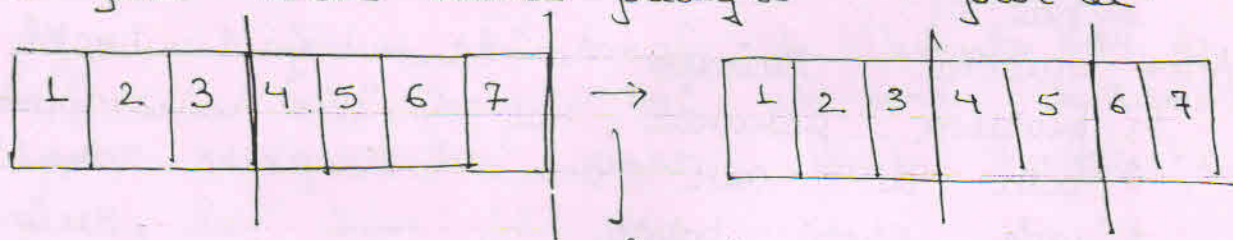
- Right hand k baad jo bhi data hai woh transmit karنا baaki hai. aur left end k pahal wale saab acknowledged hai aur current window mein jo hai woh transmit toh ho-gaye but uska acknowledgement aana baaki hai.
- Now, Receiver ne window size 10 bytes batayi, hamne 1st segment mein 1, 2, 3 send kar diya abhi uska acknowledgement nahi

aaya aur hamne IInd Segment mein 4, 5, 6, 7
Send kar diya.

• Abhi hum 3 byte aur send kar sake hai toh
afke effective window size kitani hogi 3.

• 3 byte hamne Ist Segment mein send ki the
uska ack aaya toh ^{usne} ~~usne~~ ~~afke~~ usme
apni window size kitani batayi 2.

• Toh hamko apna right hand ko wapas
5 par likar aana padega 7 par se.



after that
ack in which
Receiver ne apni size
2 bata di

Now Receiver 4, 5 toh accept karlega but 6, 7
ko reject kar dega unko phir se send karna
pavega.

• Aur ab jab tak 4, 5 ka acknowledgement
nahi aata tab tak 6, 7 ko retransmit nahi
karukey

• Aur isse concept ko hum window shrinkage
kahate hai, yaha par window shrink hogayi hai.

- Yeh problem isliye aarahai hai kyoki hum 10 byte ek segment mein send nahi kar sakte hai.

- Agar 10 byte ek hi segment mein send kar dete toh yeh problem aati hi nahi hai.

* Most of the TCP implementation does not support window shrinkage event.

- Phir aaise problem aane par kya karte hai ??

Ans → Jaisehi receiver ne detect kiya aise ky o problem aarahai hai jabaki bad window size bata di aur ab choti bata raha hai toh woh choti batata hi nahi hai, receiver apni window size kitani advertise karata hai 0.

- 3rd Ka acknowledgement aaya aur unme receiver ne window size 2 batayi toh receiver apni window size 0 advertise karata hai aur isse sender ko yeh pata chahi jayega ki usko 3rd byte k baad phere retransmit karana hai.

- Par abhi kuch bhi transmit nahi kar sakte because window size receiver ne 0 batayi hai.

- Now question arises, toh kab sender transmit karuga ??

Ans: Jab receiver ka buffer bhar jaye jab receiver ek acknowledgement packet send karuga.

usme apni window size badi badayega, tab sender data ko phir se transmit karna start karuga.

- Us acknowledgement mein data nahi hoga.
- Agar receiver ne woh special segment send kiya but woh sender tak pachuchahi nahi aur lost hogaya toh sender ko kaise pata chaliga ki receiver ne apni window size badi bata di hai. (Retransmission nahi karuga).
- Agar receiver ki taraf se bahut ~~der~~^{time} se response nahi aata hai toh sender ek segment send karuga hai usko probe kahate hai.
- Probe kya karta hai yeh, yeh check karta hai ki receiver ki window size badi hai kya.
- Agar probe k response mein agar receiver apni window size badi badata hai toh data transmission start hojayega agar uska koi response nahi aaya toh phir sender assume karuga uske ~~base~~ pass koi buffer available nahi hai.

Silly Window Syndrome :-

257

- Jab network mein aapka data bahut kam jaraha hai, par header k karan aapki bandwidth waste horaha hai aur yahi wali problem hamari kya kahalati hai silly window syndrome.
- Jab interaction application hoti hai waha arise problem aati hai.
- Yaha par do possibility hosakte hai ya-toh server slow hosakta hai ya phir client slow hosakta hai.

Nagal's algorithm :- (Sender slow)

- Agar server slow hai toh aap uske ek-ek byte buffer mein store karlo, agar buffer full hojaye uske baad send kardo.
- Ya phir client se acknowledgement aaye ki usko data chahiye hi.
- Usually ~~buffer size = 1K~~
- Interactive - application mein hum buffer nahi karvakte.

Clerk Algorithm :- (Client slow)

- Agar receiver slow hai receiver ek-ek byte mang raha hai toh hum server ka data receiver

TCP ki buffer mein rak deguy, ab Client application jitana manglegy usko de deguy.

How TCP estimate RTT??

Ans

↳ TCP mein RTT estimate karne k bahut saare algorithm hoti hai, aur RTT ek baar estimate nahi karta hai baar-baar compute karta hai.

• Usually :-

RTT \rightarrow 2bp (usually)
tx \rightarrow negligible

• RTT ko estimate karne k bahut saare different algorithm hoti hai, usually used algorithm is Kail Jacobson.

$$RTT_{n+1} = \alpha (\text{Initial}) + (1-\alpha) RTT_{\text{initial}}$$

$$(RTT)_{\text{new initial}} = \alpha * RTT_{\text{initial}} + (1-\alpha) RTT_{\text{given.}}$$

Application Layer protocol: UDP

• DHCP: (Dynamic host Configuration Protocol)

- 1] • It is a client-server application program.
- 2] • It is the first program executed on client after booting.

3] → Jab computer boot hola hai uske baad agar apko net se internet se connect hona hai, toh apko somehow apka IP-address pata hona chahiye.

• But Agar host ko nahi pata hai uska IP-address toh DHCP ki help se hum pata kar sakte hai.

• Nowadays hamne sirf IP-address hi nahi hamne subnet mask, IP-address of default Router jisse hum connected hai aur DNS IP address pata hona chahiye, agar internet se connect hona hai toh.

• DHCP protocol se pahle hamare pass RARP aur BOOTP protocol the, but RARP mein problem kya thi ki 100% sirf computer ka IP address hi detect tha.

• But BOOTP (Bootstrap protocol) ne ~~ye~~ ~~overcome~~ RARP k same disadvantage overcome kar diya but BOOTP is static configuration protocol. Matlab BOOTP mein ek table create hote hai aur uske through apko IP, address, subnet mask, gateway address, DNS IP-address milta hai.

- But agar koi naya system aaya toh aapko manually us table ko edit karna punga.
- Is problem to dur karne k liye DHCP laga gaya tha.
- DHCP use UDP. Therefore DHCP must provide error control. Usse liye UDP uses the checksum. Remember that the use of checksum in UDP is optional.
- DHCP server port no → 67
and client side port no → 68.
- DHCP, Dynamic host Configuration Protocol, do tariko se address assign karta hai, ek toh static allocation hota hai aur dusra dynamic allocation hota hai.
- # Static ~~location~~ allocation, toh similar hota hai BOOTP Protocol jaisa.
- Dynamic allocation mein kya hota hai sabse pahle client k request aayi ^{DHCP} ~~server~~ server k pass toh sabse pahle jeh aapne database mein check karunga ki is mac-address k corresponding apke pass IP-address hai kya, agar nahi hai toh DHCP k pass second database hota usme unused IP address hote hai unme se temporary koi bhi IP-address dedega, kuch time k liye aur yeh entry aapne database mein store karunga.