- Now aab hum dekte hai problem kaise nahi aayegi.
- Sabse pahale Sender ne pahala frame send ki with frame number 0, reciever ne usko recieve kar liya, now aab reciever kya accept karega frame with sequnce number 1.
- Now frame 0 ka acknowledgement Network mein lost hogaya, toh Sender ne phir se frame 0 Send ki.
- Now Reciever kya karega, woh toh frame 0 ko except hi nahi karraha tha, toh aab reciever kya karega toh it will discard that dulplicate frame.

Lekin reciever us frame k liye acknowledgement Send karega.

- Now reciever kya expect karrahai hai frame with Sequnce number 1, now Sender ne frame with Sequnce number 1 Send ki, reciever ne recieve kari.
- Now suppose frame 1 network mein lost hogi, toh aaisi situation mein kya hoga, toh jab tak frame with Sequnce number 1 ka acknowledgement Sender k pass nahi pachuchega jab tak Sender next frame send hi nahi karega.
- Joh timout hone k baad Sender woh wali frame retransmit kardega.

* Joh Stop and wait mein hum 1 bit ka sequence number leta hai.

- Pahali frame ka sequence number 0 rakta hai next wali ka 1, phir third ka phir 0 and so on.

- 1 bit se hamari problem solve horahai hai, ek bit se jada ki jarurat hamne nahi hai.

- Joh hamare numbering kyo ki, duplicate aur original frame ko differentiate karne k liye.

- Agar frame number corrupt hogaya 1 ka 0 hogaya, hamari error control technique ne usse detect kar liya, ki error hai toh woh detect hojayegi.

- But agar error control bhi detect nahi karpayi ki kohi error hai toh woh usko nayi frame samaj k accept karlegi.

- Agar undetect error hai toh woh us frame ko nayi frame samaj k detect karlega.

- Error control hum kon-kon si layer par karte hai, data-link layer aur transport layer par.

- Transport Layer par toh error kaha check hoti hai kewal ultimate host aur ultimate destination par, bich k station par nahi hoti hai.

- Aur data-link ki error A se B par gayi toh B ki datalink layer ne check karliya ki sahi aaya hai kya.

tabhi.

- ~~Abb~~ B kisko send karega C ko;

- Now datalink layer har system par error check karrahai hai toh phir transport layer par ek aur baar check karne ki kya jarurat hai.

- Woh isliye check karte hai ki agar data-link layer detect nahi kar payi, kohi bhi error-control technique yeh gaurantly nahi deti hai ki woh error ko 100% detect kar legi.

- Agar woh error datalink-layer par detect nahi hopayi ~~tt~~ toh ~~#~~ shayad woh transport layer par detect hojaye.

- Isliye datalink layer par error control k. liye dusri technique use karti hai aur transport layer par different.

- Hosakta ki datalink layer detect nahi karpayi, usko kon detect kar ie transport layer ki error control ki technique.

- Yeh bhi possible hai ki dono hi detect nahi kar paye.

- Kohi bhi protocol ho, uska kya important hoti Efficiency important hoti hai.

Q → How to compute Efficiency of stop and wait protocol :→

→ In general efficiency kaise nikalte hai.

$$\text{Efficiency} = \frac{\text{Useful Time}}{\text{Total Time}} = \frac{\text{Output}}{\text{Input}}$$

- Yaha par input kya hai total hamara kitana time invest kiya.
- Aur output kya hai useful time.
- Yaha par puri overall activity mein system sirf kitani der busy hai t_x, jab woh data transfer kar raha hai tab woh busy otherwise woh busy nahi hai.
- Total time :→ pahali frame send ki jab tak agali frame send karna ka time nahi aajata woh uska total time hojayega.
- Pure time mein effective kya send karraha hai ek frame.
- Jo EK frame ko send karne mein kitana time lag raha hai yaha par- RTT.
- Jo total time kitana RTT. RTT mein kawal ek frame send karraha hai.
- Useful time kya hai hamara, transmission time

hamara useful hai i.e t_x, uske aalawa hamara System Ideal hai.

- Jotak kitana time invest karraha hai hum isme RTT.
- RTT kitana hai hamara $t_x + 2t_p$.

$$\text{Efficiency} = \frac{t_x}{t_x + 2t_p} = \frac{t_x}{RTT}$$

Ques: A/ 4 Kbps / channel

Ans: Frames of 1000 bits are to be send over 4 Kbps channel where signal traverse at the speed 2/3 of the light and distance between the station is 1 km. What is the efficiency if stop & wait protocol is used.

$(c = 3 * 10^8 \text{ m/s})$

$$\eta = \frac{t_x}{t_x + t_p + t_p} =$$

$$t_x = \frac{\text{Length of data}}{\text{Data-transfer Rate}} = \frac{1000}{4 * 10^3} = 0.25 \text{ sec}$$

$$t_p = \frac{\text{Distance}}{\text{Signal Propogation speed}} = \frac{1 * 10^3}{2 * \frac{2}{3} * 10^8} = 10^{-5} * 0.5$$

$$\eta = \frac{0.25}{0.25 + 2 * 0.5 * 10^{-5}} = \frac{0.25}{0.25 + 10^{-5}} = 0.99 \text{ f}$$

Ans → = 99%.

IT-2005 → Ques - 72 (d)

• Timer kab start hota hai frame transmit karne k baad.

Question:

• Under what condition does stop and wait protocol results or returns 100% efficiency ??

Ans:

$$\eta = \frac{t_x}{t_x + 2t_p} \Rightarrow 1 = \frac{t_x}{t_x + 2t_p}$$

$$= t_x + 2t_p = t_x \Rightarrow \boxed{t_p = 0}$$

Jab $t_p = 0$ hoga tabhi 100% efficiency yeh protocol return karega.

* Matlab yeh kabhi bhi 100% efficiency return hi nahi karega.

• Because do system k bich kuch na kuch distance hoga hi.

** Toh stop and wait kabhi bhi 100% efficiency nahi dikhakta.

• Toh hum efficiency kaise improve kare ??

→ Pehla question toh yahi hai efficiency 100% kyo nahi hosakta, kya problem hai, toh problem yeh hai ki Sender send karne k baad acknowledgement ka wait karta hai,

• Yeh waiting time efficiency ko kya kar deta hai reduce kar deta hai.

• Toh kya kare is problem ko dur karne k liye??

- Round trip delay product = RTT * Bandwidth
- Bandwidth delay product = 2tp * Bandwidth

- Sender wait nahi kare, Matlab woh continous send karta rahai
- Toh continous kab tak send karta rahai, toh sender ko kab tak send karna chahiye, jab tak first frame ka acknowledgement nahi aa-jata.
- First frame k acknowledgement ko aane mein kitana time lagta hai equal to RTT.
- Toh Sender ko kab tak send karna chahiye frame, ufio the time period equal to RTT,
- Agar suppose RTT Time mein hum 100 frames send kar sakta hai toh Sender 100 frame send kare kahi wait nahi kare.
- 100 frame send karne k baad dekte ki first ka acknowledgement aaya hai ki nahi.
- Agar 100 frame send karne k baad first ka acknowledgement nahi aaya, toh sender first frame k liye kya hojayega timeout.
- Aur phir se I^st frame ko sent karega.
- Agar transmission time i.e tx = RTT hojayega tab hum 100% efficiency milegi.
- Matlab jab tak first frame ka acknowledgement nahi aajata Sender frames send karta rahai., toh hi hamne

Efficiency kitani milegi 100%.

- Lekin problem kya hai, ki RTT time mein hum 100 frames send kar sakte hai, toh sender ko 100 frames apne pass buffer karke rakhni pargi.

- Kyo rakni pargi, agar unka acknowledgement nahi aaya toh sender ko un frames ko retransmit karna parega.

- Toh Sender k pass itani memory/buffer bhi hona chahiye ki woh 100 frames ko temporary store kar sake.

- Toh Stop and wait ki efficiency ko improve karne k liye hamne dekha, ki agar hum continous frame transmit kare without waiting for acknowledgement, toh is case mein hamko efficiency 100% mil sakta hai, par isme problem kya hogayi, agar RTT = 10ms hai aur 10ms mein 100 frames send karsakte hai, toh 100 frame store karne ki capacity Sender end par honi chahiye.

- Agar itani memory nahi hai toh aap itani frame ek saath nahi send karsakte.

- Definitly itani nahi hai toh 40, 50 toh lesakte hai, 100% efficiency toh nahi hui, toh bhi Stop & wait se toh aachi efficiency milegi.

- Aur yaha protocol kya kahalga hamara Sliding

Send kar dega.

- Ek-ek frame ka ack aata jaraha hai, aur ek-ek frame ko send karta jaraha hai.

- Generally jo misconcept hota hai Sliding Window protocol mein, woh yeh hota haiki 1 se 10 frame send kar di, un 10 frame ka acknowledgement jab tak aajayega uske baad next 10 frame send karegi.

- Par aaisa kuch nahi hota hai sliding window protocol mein.

- Agar window size 10 hai toh sender window always contain 10 unacknowledgement frame.

- Suppose hamari window size 3 hai, toh sabse pahale Sender ne kya Send kiya frame number 1, 2, 3