Name: K Jaswanth Reddy

UFID : 22719671

CN - Assignment - 3

(1) Ch3 - P5 :-

Suppose UDP computes internet checksum and finds that it matches the value carried in checksum field. Can the receiver be absolutely certain?

Ans :- → The receiver cannot be certain absolutely of having no bit errors to occur. It can be convinced by evaluating the checksum for a packet.

→ for example, a message of 16-bit, and if any 2 corresponding bits gets changed from 1 to 0 or 0 to 1, and the checksum still remains the same.

→ This concludes, checksum will verify even if there was an error while transmission.

(2) Ch3 - P6 :-

Ans:

→ The Sender & receiver can enter into deadlock state for the given flow. for example assume sender is in the state : "wait for the call 1 from above".

⇓

Sender has to send a packet with the Sequence # 1 and transmits to.

⇓

wait for "ACK or NAK".

→ The receiver who is receiving, Sequence #1 packet, it correctly

has to send 'Ack' and has to turn to the state "wait for 0 below".

⇒ Lets say, Ack is lost/corrupted and transmitted as NAk then Sender thinks that Packet #1 was faulty and resends the respective packet #1.

⇒ whereas the receiver was in a state: "to accept packet: 0" and sends a message NAK.

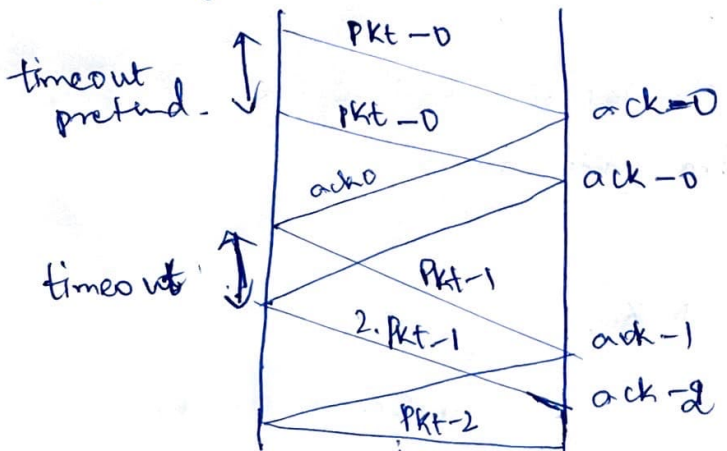
⇒ Then Sender will assumes packet #1 is corrupted and will again sends packet #1 forever and receiver will again NAK. This makes them to be in the same state forever without any progress.

③ Ch 3 - P12 :-

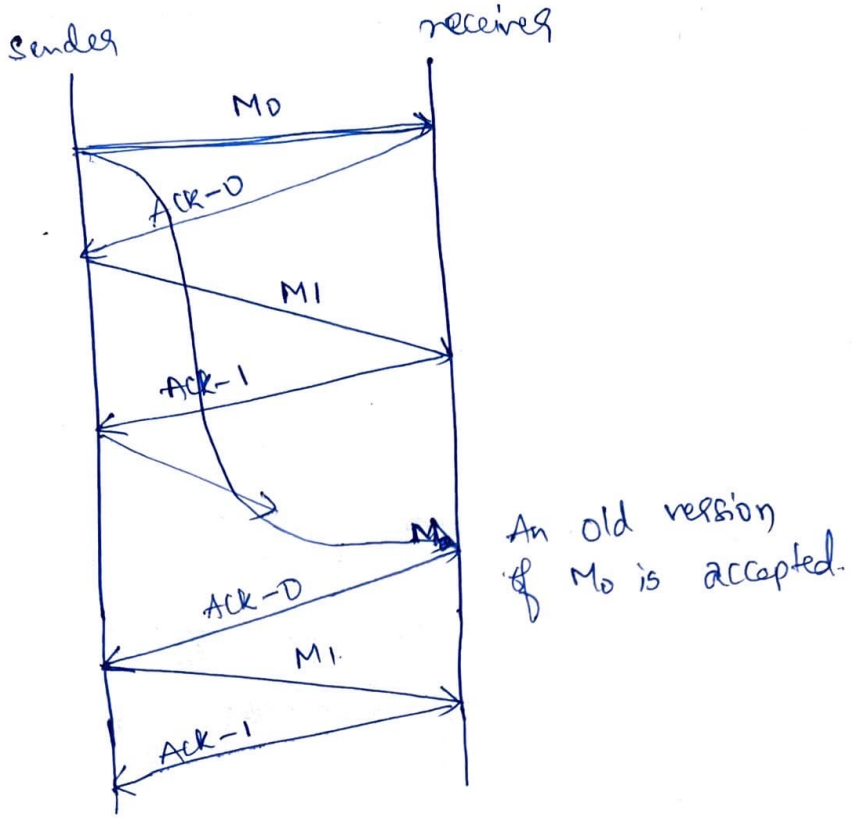
Ans: Even when any of the packet is lost, the protocol would still continue to work it will retransmit.

⇒ with the premature timeouts, for each of the extra copy of Ack packet then for this an extra copy an extra Ack has to sent.

*) This will make the number of packets to be sent, will increase without the bounds as 'n' approaches the infinity.



(4): Ch3 – P13 :-



→ If Sender sent menage 0 & receiver after receiving it as per expectation and ACK-0 is sent, and it is sent the same fore Menage 1.

→ Later, for the next time when menage 0 is lost, an older version of Menage 0 is accepted & sends to the receiver it continues with flow and progress further.

(5) Ch3 – P16 :-

Ans:

Packet size ⇒ 1500 bytes

Time to sent 1 packet ⇒ $\frac{\text{Packet size}}{\text{rate}}$

⇒ $\frac{12 \times 1500 \times 8 \text{ bits/packet}}{10^{6} \text{ bits/sec}}$

→ Probability for Sender to be busy ⇒ 98%

⇒ 12×10^{-6} sec ≃ 12 μs.

⇒ 0.012 ms.

Probability $\Rightarrow \dfrac{(\text{Time for 1 packet to sent}) * \#\text{ of packets}}{30 + 0.012}$

$$0.98 = \frac{(0.012) \cdot n}{30.012}$$

$$n \Rightarrow \frac{0.98 * 30.012}{0.012}$$

$n \Rightarrow 2450.98 \approx 2451$ packets.

$\Rightarrow$ we should have atleast 2451 packets to be sent for making 98% utilization of Channel.

(6) Ch3 - P19:-

Ans It was all about the protocol "stop & wait". Given, when the sender 'A' sends messages to receivers (B, C)

$\Rightarrow$ The channel sometimes/may loose messages and any one of them doesn't send back the ACK, the sender may have to resend the menage due to timeout/ incorrect transfer.

$\Rightarrow$ But, for the one (receives) which has correctly receive the menage will get the message twice. Thus, a 0-bit sequence number would be adequate.

(7) ch3 - P22 :-

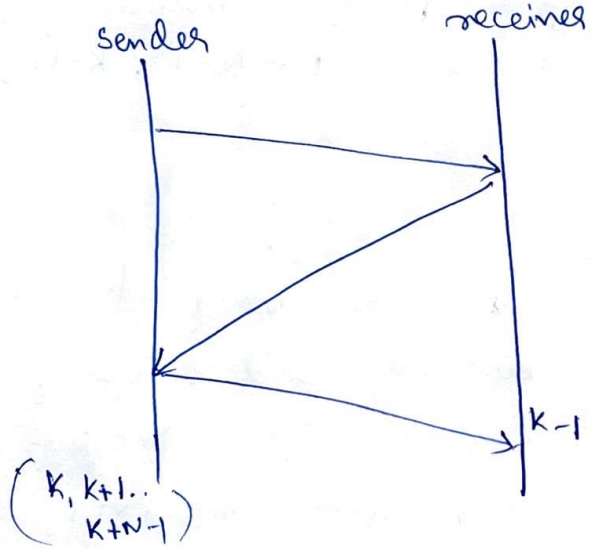
a). what are the possible sets of sequence numbers inside the senders window at time t ? Justify answer.

Ans:-

window size : 4

Range : 1024.

⇒ Let, if the receiver has got all (k-1) packets acks that sender has sent, then the sender window will only have the packets between k-N. Thus, window size at the sender will be [k, k+N-1]



⇒ If none of the Ack, has been received at sender, then k-1 and N packets till and including k-1, the senders window has to be [k-N, k-1]. Thus, making the window size will be as [k-N, k]

b). what are all possible values of Ack field in all possible messages currently propagating back at a time t ?

Ans:- when receiver was waiting for packet 'k', then it suggests that it has already received (k-1) packets succes-fully and has sent Ack to all of the received ones.

⇒ Lets say, those N Ack has not been received to the sender, then [k-N, k-1] Ack messages should be moving back to Sender.

⇒ Sender has sent [k-N, k-1] packets, which means Sender might have had received k-N-1 Ack.

*) If receiver sends Ack for any packet, it will not send for below packet. for [k-N-1] Ack is sent, it thus never send Ack for packets below that and having the window size to be [k-N-1, k-1] #

(8) Ch 3 - P 23:-

Ans:- for avoiding problems, we have to avoid leading edge of receivers window wrap around and overlap with traveling edge i.e., sequence number space should be long enough to fit both the sender and receiver window without being overlap.

To find range:-

If the lowest sequence number waiting to receive is m, which concludes that m-1 packets are received and w-1 packets before that and window size will be [m, m+w-1]

⇒ If there is no Ack received from the Sender, then Ack messages with values [m-w, m-1] will be the packets propagating back-to-sender.

⇒ If Ack are not received by sender, window size would be $[m-w, w-1]$ and the lower edge of the Sender's window is '$m-w$'. Therefore the sequence number must accomodate space for $2w$ sequence numbers. $\{k > 2w\}$

(9) Ch3 - P24:

a) with SR protocol, it is possible for sender to receive an Ack for a packet that falls outside of it's current window.

Ans:- True.

Let us understand it with a situation.

t_0 : Sender sent packet with window size 4: 1, 2, 3, 4.

t_1 : receiver will send Ack for 1, 2, 3, 4.

t_2 : Sender might times out & sends again 1, 2, 3, 4.

t_3 : Sender receives Ack for 1, 2, 3, 4 (sent at t_1) and will move to other packets 5, 6, 7, 8.

t_4 : receiver might send Ack for 1, 2, 3, 4 again.

t_5 : Sender receives Ack for 1, 2, 3, 4 at (t_4). The Acks which Sender receives are outside it's window.

b) with GBN, it is possible for sender to receive an Ack for a packet that falls outside of its current window.

Ans:- Correct. Consider the above Question, the re-acknowledgements also fall outside the window.

c) The alternating-bit protocol is same as SP-protocol with a sender and receiver size of 1.

Ans:- True. Considering window size '1', alternating-bit protocol and CR protocol are methodically equivalent.

⇒ Considering the possibility of out of order are precluded by window size-1 and Cumulative Ack becomes ordinary Ack when window size = 1.

d) The Alternating bit protocol is same as GBN protocol with a sender & receiver window size = 1.

Ans:- Yes, it is same. when the window size = 1. the Alternating bit protocol is same as GBN and when window size = 1, it masks the possibility of out of order and Cumulative Ack is treated/considered as Ordinary.

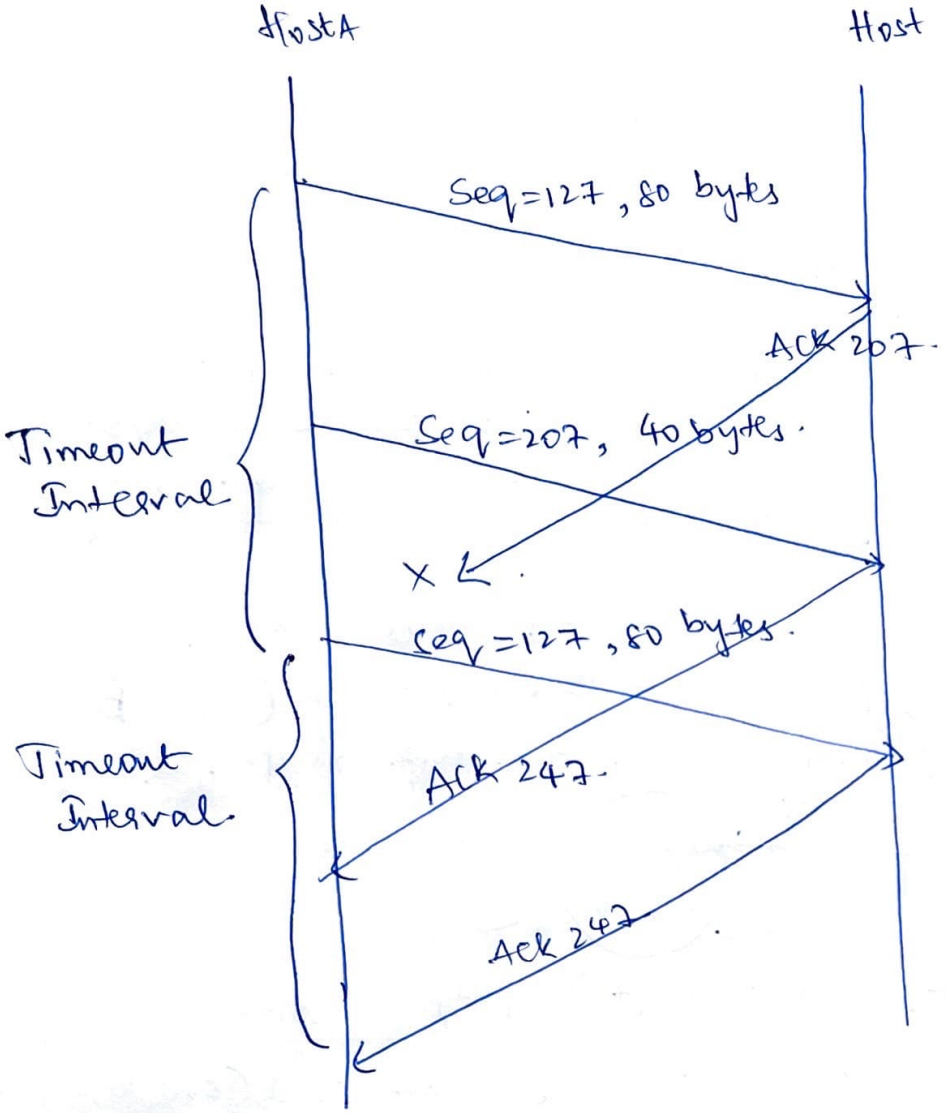
(10) ch3 – P27:

a) In the 2nd Segment from A → B, the Sequence # 207, Source port # 302, destination port # 80.

b) when 1st Segment arrives before 2nd in the Ack of first segment. The Ack # is 207, source port # 80 and destination port # 302

c) If the 2nd segment arrives before 1st segment. In the Ack of the first arriving segment, the Ack # 127, it signifies that it is still waiting for 127 Bytes and continues.

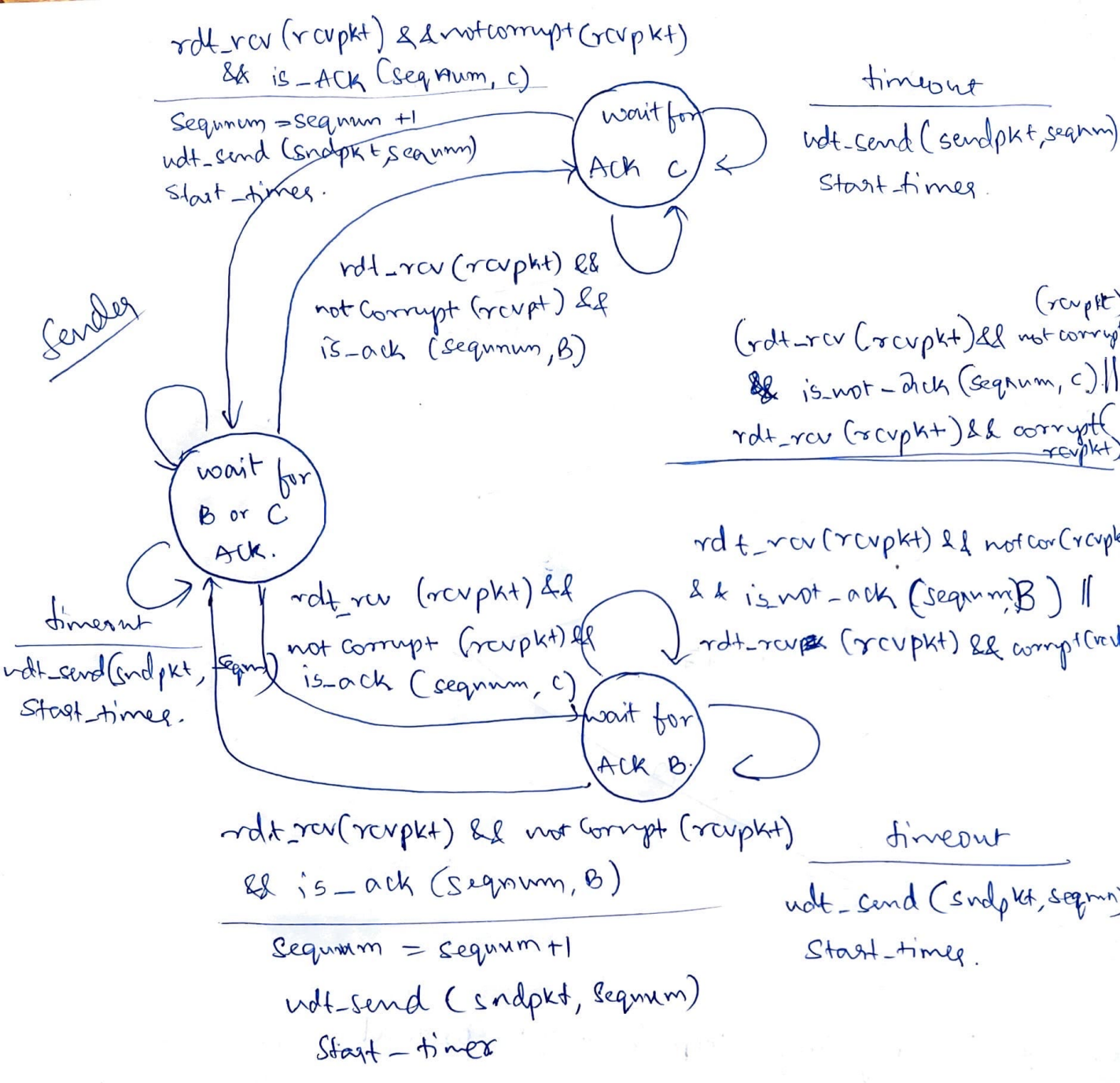
d).



⑥ Ch3–P19:-

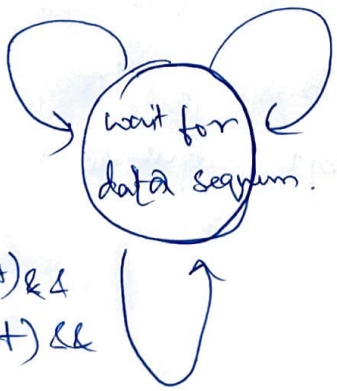
Ans:- It is regarding stop & wait protocol. Given in the question when A → (B, C), the channel sometimes looses messages & when any of them doesn't provide ACK, then the senders will resend the message which may be due to timeout/incorrect transfes.

→ The Correctly received message may be recordesed twice. And we can say 0-bit sequence number is adequate for this scenario.



→ Sender state indicates that whether Ack has received from B only, only or from neither B & C.

receiver B



4) for receiver 'C' it is same as that of B. The state indicates whether the seq.# it is waiting on and it checks if the menage is not lost and it has received a seq. number is some as it is waiting for and we increase the sequence number. when a message is corrupt / or when a packet is received it has package number different than the ~~other~~ one which is expected so we need to send NAK.