Computer Networks

Fall 2023/24
Exercise 3 -- Solution

Submission by Sunday, 4-2-2024. Submit by uploading your work to the course Moodle website.
>>> No late submissions will be accepted!
The name of the submitted file must be Exercise3_firstname_lastname1.[suffix].
For example, Exercise3_israel-israeli.pdf. The first and last name of the student must appear.

Problem 1.

Assume at time t sender sends k packets from the sender to the receiver, which is also the window size. The transmission delay of a packet d, the round-trip-time is RTT and the timeout-timer time is $timeout$.

(a) If Selective Repeat protocol is used, at what time, at the earliest, the receiver receives all k packets assuming that only the third packet is lost, and that there are no losses afterwards?

Explain your answer.

To send the first 2 packets it takes the transmission delay to write both packets, 2d. Since the 3rd packet got lost, the sender waits 1 timeout, then resends the 3rd packet, which again takes 1 d. After resending the 3rd packet, the sender then sends the remaining packets it hasn't sent yet. During the timeout, the sender continued sending packets, so to calculate the remaining packets to be sent after the timeout we have: (k-3 – timeout/d) which is all the packets except the first 3 minus the number of packets sent during the timeout. Each one of those packets takes time d to transmit which results in (k-3 – timeout/d)(d) time. Then the time it takes for the last packet to reach the receiver, which is RTT/2. Therefore the time the receiver reciebes all k packets is at t0 + 2d + (timeout + d) + (k-3 – timeout/d)(d) + RTT/2, which simplifies to t0 + (k)(d) + RTT/2.

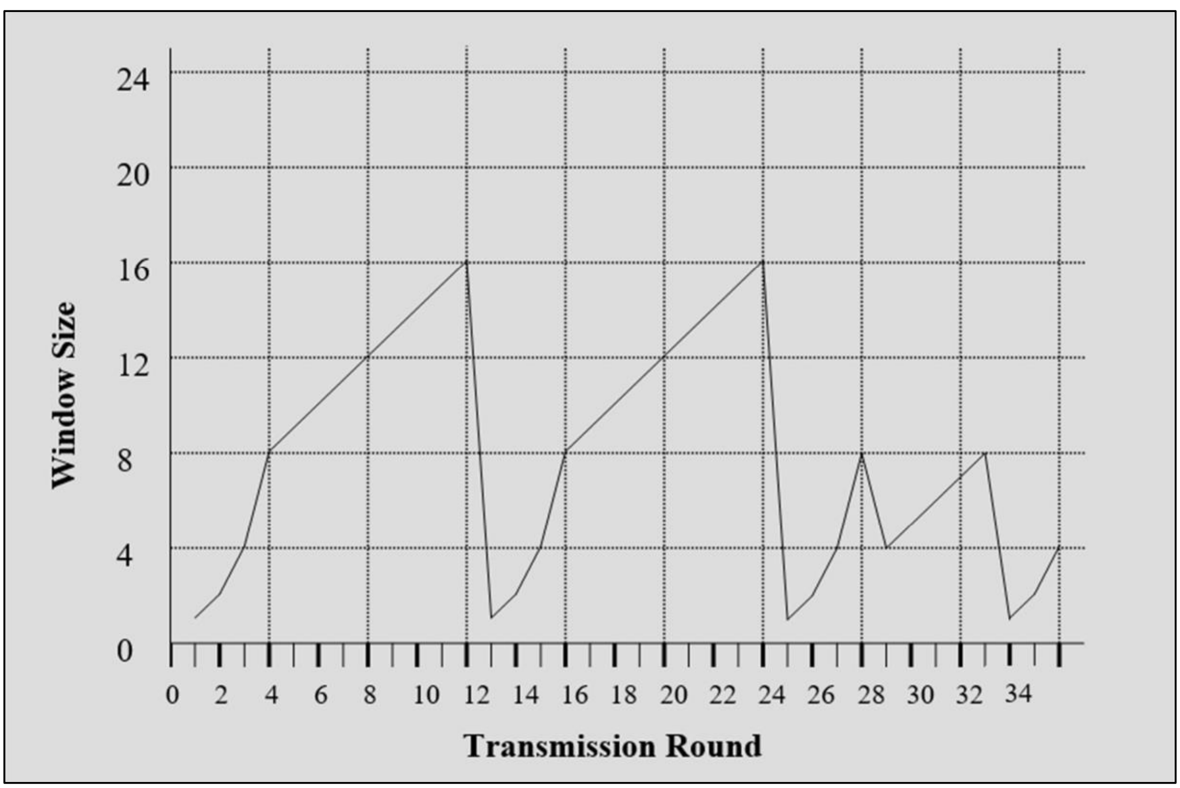
If Selective Repeat protocol is used, at what time, at the earliest, the receiver receives all the k packets assuming that the third packet and the fourth packet are lost, and that there are no losses afterwards?

Explain your answer.

To send the first 2 packets it takes the transmission delay to write both packets, 2d. After the transmission of packet 3, the timeout for packet 4 starts. Therefore, the time passed until the end of packet 4's timeout is d + timeout (d is the transmission delay for the 3rd packet). Note that packet 3 is resent during the last d seconds of packet 4's timeout (This is because the timeout of packet 3 started d seconds before packet 4's timeout). During packet 4's timeout, the sender continued sending packets, so to calculate the remaining packets to be sent after the timeout we have: (k-4 – timeout/d) which is all the packets except the first 4 minus the number of packets sent during the timeout. Each one of those packets takes time d to transmit, which results in (k-4 – timeout/d)(d) time. Then the time it takes for the last packet to reach the receiver, which is RTT/2. Therefore the time the receiver receives all k packets is at t0 + 2d + (timeout + d) + (k-4 – timeout/d)(d) + RTT/2, which simplifies to t0 + (k+1)(d) + RTT/2.

Problem 2.

Consider the following plot of TCP window size as a function of time, that is, as a function of the number of rounds. (In each round the sender transmits all the segments in its congestion window and either receives acknowledgements for them or there is a loss event.)



Assuming TCP Reno is the protocol experiencing the behavior shown above and answer the following questions. Justify all your answers shortly.

a. Identify all the intervals of time when TCP slow start is operating, and identify the intervals of time when TCP congestion avoidance is operating.

TCP slow start: [1,4] , [13,16] , [25,28], [34,36]
TCP congestion avoidance: [5,12], [17,24], [29,33]

b. What are all the transmission rounds in which segment loss is detected by a triple duplicate ACK?

At 28

c. What are all the transmission rounds in which segment loss is detected by a timeout?

12, 24, 33

d. What are all the values of Threshold during the first 34 transmission rounds (starting from round 1)?

8, 4

e. During what transmission round is the 75th segment sent?

Round	Sent	Total
1	1	1
2	2	3
3	4	7
4	8	15
5	9	24
6	10	34
7	11	45
8	12	57
9	13	70
10	14	84

Therefore the 75th segement is sent during the 10th transmission round.

f. Assuming a packet loss is detected during the 36 round by the receipt of a triple duplicate ACK, what will be the new values of the congestion window size, and of Threshold?

2 for both the window size and the threshold.

Problem 3.

To get around the problem of sequence numbers wrapping around while old packets still exist, one could use 64-bit sequence numbers. However, theoretically, an optical fiber can transmit at 75 Tbps (75 x 1012 bits per second).
Assume that each packet is of size 1500 bytes and has a lifetime after which if it does not arrive to it destination it simple disappear.

What is the maximum packet lifetime required to make sure that future 75 Tbps networks do not have a wraparound problem even with 64-bit sequence numbers?

Assume that each byte has it own sequence number, as TCP does. Justify your answer.

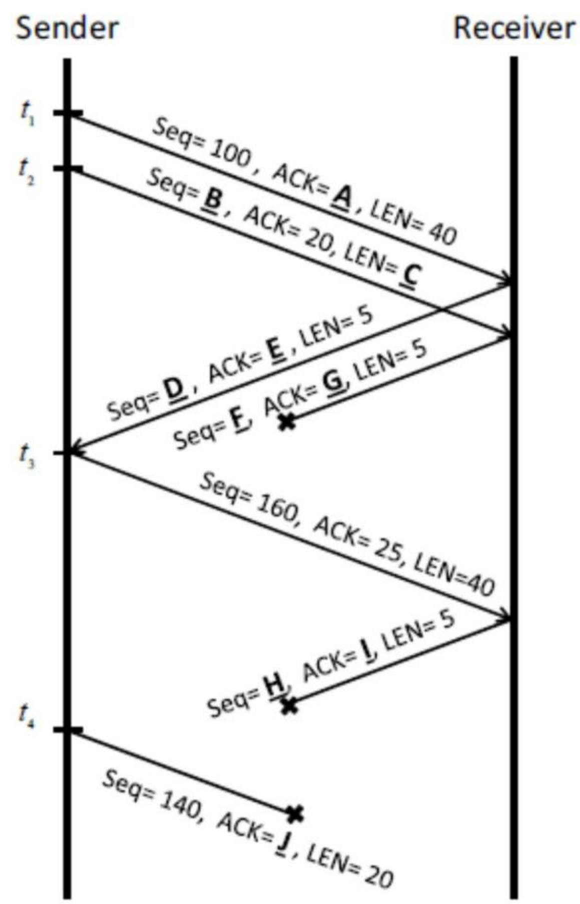
The size of a segment (the packet and its sequence number) is 1500 + 2^64 bytes = 1.84 x 10^19. The speed of the link in bytes is 9.375 x 10^12 bytes per second. Therefore the time it takes for a segment to reach the receiver is (1.84 x 10^19) / (9.375 x 10^12) = 1967652.701 seconds.

Problem 4.

Consider the TCP connection in the figure below, where the sender starts the timer at t . a. Complete the missing sequence numbers (Seq), acknowledgment numbers (ACK), and segment length (LEN), in the table.

b. Calculate the value of timeout for the segment with the sequence number 140.

A	20
B	140
C	20
D	20
E	140
F	20
G	160
H	25
I	200
J	25



The timeout for segment with seq #140 is t4 – t2

Problem 5.

Consider transferring an enormous file of L bytes from host A to host B.
Assume the MSS is 536 bytes.

a. What is the maximum value of L such that the TCP sequence numbers are not exhausted?

   The maximum value of L possible is 2^32, because in a TCP segment, the sequence number is a 32 bit long integer and 2^32 is the largest 32 bit integer.

b. For L length you calculated in (a), find how long it takes to transmit the file.
   Assume that a total of 66 bytes of headers (transport, network, and data-link) are added to each segment before the resulting packet is sent out over a 100 Mbps link.
   Assume that A can send out segments back to back and continuously, without interruptions.
   Assume that 1 Mbps = 1,000,000 bits per second.

   The number of segment required to send the whole file is L/536 = 2^32/556 = 8012998.687, which we round up to 8012999 segments. Each segment is sent with 66 bytes of headers, therefore an additional 66(8012999) = 528857934 bytes. Therefore the total bytes of data ti be transferred is 2^32 + 528857934. The link is 100 Mbps = 12500000 bytes per second. Therefore the amount of time taken to transfer the file is (2^32 + 528857934)/12500000 = 385.91 seconds.

Problem 6.

Consider Selective Repeat protocol and assume that messages can be duplicated or lost, but messages cannot corrupt, and the medium does not reorder messages.
Let w denote the window size.
What is the maximum number of <u>different</u> messages that can be in transit from the sender to the receiver at the same time? (Different messages are messages which have different data content.)

What is the answer to the above question in the case of Go-Back-N protocol?

Justify your answers.

For Selective Repeat protocol, the maximum number of messages that can be in transit is w. By design, the window size limits the number of packets that can be send at the same time. Each of these messages can be unique and therefore the maximum number of different packets can be w.

For Go-Back-N protocol the maximum number of messages that can be in transit is again w. If all the messages in the window are unique, then when the window is sent, w different message are in transit. If a message gets lost, the whole window is resent. While there can now be more than w messages in transit, there is only w unique messages as for every message that is in transit and was lost the one being resent is a duplicate.

Problem 7.

Consider the Go-Back-N protocol with window of size W and infinitely many sequence numbers. Suppose that a time t, the next in-order packet that the receiver is expecting has a sequence number k.

a. What are all the possible sequence numbers (as a function of W and k) that can appear at (any place in) the sender's window at time t?
   Justify your answer.

   The possible sequence numbers are k-w to k+w-1. The 2 extremes are if the receiver is expecting packet k and the sender has received acknowledgement for packet k-1. This means that the window include packets k to k+w-1. The other extreme is if the receiver is expecting packet k but the sender has not yet received acknowledgements for the previous w-1 packets (since there are still in transit). This means the window includes packet k-w to k. Any options between these 2 extremes are possible (i.e any size w group of consecutive sequences between k-w and k+w-1)

b. What are all the possible values of the ACK field in all possible messages currently propagating back from the receiver to the sender at time t?
   Justify your answer (Assume that the medium does not reorder messages).

   The possible values of the ACK field are ACK(k-w) to ACK(k-1). If the receiver is expecting packet k, then it hasn't received it yet and therefore has not sent ACK(k). Therefore, the remaining possible ACKs are all the ACKs before ACK(k) that are in the window. According to our answer above, this could be from ACK(k-w) to ACK(k-1).