CMSC 440 - Homework Assignment #3 - Solution

Due Date: Sunday April 2nd, 2023 - 11:59pm

Questions:

You are required to answer the following questions from the attached document:

Question
P.15
P.22
P.25
P.27

Question
P.37
P.40
P.43
P.44

[5 points] Problem 3.15

It takes 120 microseconds (or 0.12 milliseconds) to send a packet, as $1500*8/10^8=120$ microseconds. Given the one-way propagation delay is 25 milliseconds, the RTT is 50 milliseconds. In order for the sender to be busy 98 percent of the time, we must have:
util = 0.98 = (0.12n)/50.12 $\rightarrow$ n = 410 packets to be busy 98% or more of the time.

[10 points] Problem 3.22

a) Here we have a window size of N=3. Suppose the receiver has received packet k-1, and has ACKed that and all other preceding packets. If all of these ACK's have been received by sender, then sender's window is [k, k+N-1]. Suppose next that none of the ACKs have been received at the sender. In this second case, the sender's window contains k-1 and the N packets up to and including k-1. The sender's window is thus [k-N, k-1]. By these arguments, the senders window is of size 3 and begins somewhere in the range [k-N, k].
b) If the receiver is waiting for packet k, then it has received (and ACKed) packet k-1 and the N-1 packets before that. If none of those N ACKs have been yet received by the sender, then ACK messages with values of [k-N,k-1] may still be propagating back.Because the sender has sent packets [k-N, k-1], it must be the case that the sender has already received an ACK for k-N-1. Once the receiver has sent an ACK for k-N-1 it will never send an ACK that is less that k-N-1. Thus, the range of in-flight ACK values can range from k-N-1 to k-1.

[10 points] Problem 3.25

a) Consider sending an application message over a transport protocol. With TCP, the application writes data to the connection send buffer and TCP will grab bytes without necessarily putting a single message in the TCP segment; TCP may put more or less than a single message in a segment. UDP, on the other hand, encapsulates in a segment whatever the application gives it; so that, if the application gives UDP an application message, this message will be the payload of the UDP segment. Thus, with UDP, an application has more control of what data is sent in a segment.

b) With TCP, due to flow control and congestion control, there may be significant delay from the time when an application writes data to its send buffer until when the data is given to the network layer. UDP does not have delays due to flow control and congestion control.

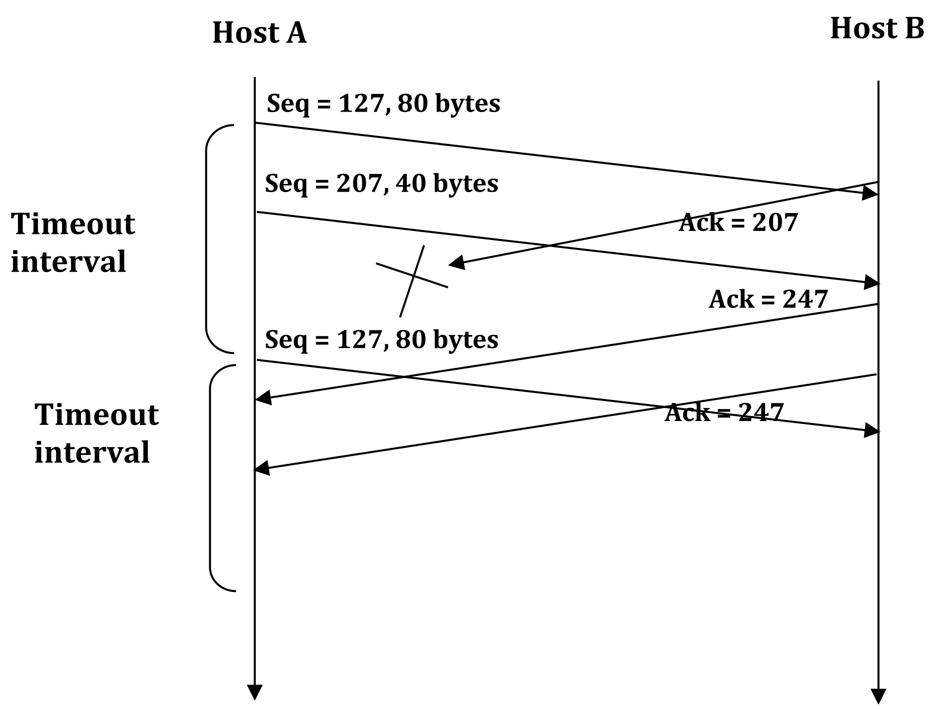
[15 points] Problem 3.27

a) In the second segment from Host A to B, the sequence number is 207, source port number is 302 and destination port number is 80.

b) If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 207, the source port number is 80 and the destination port number is 302.

c) If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, the acknowledgement number is 127, indicating that it is still waiting for bytes 127 and onwards.

d)



[15 points] Problem 3.37

a) GoBackN:
   A sends 9 segments in total. GBN initially sent segments 1, 2, 3, 4, 5 and later re-sent segments 2, 3, 4, and 5 since the sender will retransmit all segments in the buffer when the timer times out.
   B sends 8 ACKs. They are 4 ACKS with sequence number 1, and 4 ACKS with sequence numbers 2, 3, 4, and 5.

   TCP:
   A sends 6 segments in total. They are initially sent segments 1, 2, 3, 4, 5 and later re-sent segments 2. Not that TCP retransmit only the first packet in the buffer when the times out before it reset its CWND size to 1. Since CWND becomes 1, no other segment is retransmitted.
   B sends 5 ACKs. They are 4 ACKS with sequence number 2. There is one ACK with sequence numbers 6. Note that TCP always send an ACK with expected sequence number.

b) TCP. This is because TCP uses fast retransmit without waiting until time out. Recall that fast retransmit in TCP (ether Tahoe or Reno) happens when receiving 3 duplicate ACKs in which the TCP sender immediately retransmit only the first segment in the buffer. No other segment is retransmitted at this time even if CWND is greater than 1.

[25 points] Problem 3.40

a) TCP slowstart is operating in the intervals [1,6] and [23,26]
b) TCP congestion avoidance is operating in the intervals [6,16] and [17,22]
c) After the 16th transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.
d) After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.
e) The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.
f) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.
g) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion windows size is 29. Hence the threshold is 14 (taking lower floor of 14.5) during the 24th transmission round.
h) During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets 16-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets

64 – 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.

i) The threshold will be set to half the current value of the congestion window (8) when the loss occurred and congestion window will be set to the new threshold value + 3 MSS . Thus the new values of the threshold and window will be 4 and 7 respectively.

j) threshold is 21, and congestion window size is 1.

k) round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.

[5 points] Problem 3.44

In this problem, there is no danger in overflowing the receiver since the receiver's receive buffer can hold the entire file. Also, because there is no loss and acknowledgements are returned before timers expire, TCP congestion control does not throttle the sender. However, the process in host A will not continuously pass data to the socket because the send buffer will quickly fill up. Once the send buffer becomes full, the process will pass data at an average rate or R < S.

[10 points] Problem 3.44

a) It takes 1 RTT to increase congestion window (cwnd) to 7 MSS; 2 RTTs to increase to 8 MSS; 3 RTTs to increase to 9 MSS; 4 RTTs to increase to 10 MSS; 5 RTTs to increase to 11 MSS; 6 RTTs to increase to 12 MSS.

b) In the first RTT 6 MSS was sent; in the second RTT 7 MSS was sent; in the third RTT 8 MSS was sent; in the fourth RTT 9 MSS was sent; in the fifth RTT, 10 MSS was sent; and in the sixth RTT, 11 MSS was sent. Thus, up to time 6 RTT, 6+7+8+9+10+11 = 51 MSS were sent. Thus, we can say that the average throughput up to time 6 RTT was (51 MSS)/(6 RTT) = 8.5 MSS/RTT.