Homework Solution 4

1. (20%)
   a. (10%) Consider a reliable data transfer protocol that uses only negative acknowledgments. Suppose the sender sends data only infrequently. Would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

      No.

      In a NAK only protocol, the loss of packet x is only detected by the receiver when packet x+1 is received. That is, the receivers receives x-1 and then x+1, only when x+1 is received does the receiver realize that x was missed. If there is a long delay between the transmission of x and the transmission of x+1, then it will be a long time until x can be recovered, under a NAK only protocol.

   b. (10%) Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

      Yes.

      On the other hand, if data is being sent often, then recovery under a NAK-only scheme could happen quickly. Moreover, if errors are infrequent, then NAKs are only occasionally sent (when needed), and ACK are never sent – a significant reduction in feedback in the NAK-only case over the ACK-only case.

2. (20%) Consider the Go-Back-N (GBN) protocol with a sender window size of 3 and a sequence number range of 1024. Suppose that at time *t*, the next in-order packet that the receiver is expecting has a sequence number of *k*. Assume that the medium does not reorder messages. Answer the following questions:
   a. (10%) What are the possible sets of sequence numbers inside the sender's window at time *t*? Justify your answer.

      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. (10%) What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time *t*? Justify your answer.

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 $k-1$.

3. (20%) Answer true or false to the following questions and briefly justify your answer:

a. (5%) With the Selective Repeat (SR) protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

True.

Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at t_0. At t_1 ($t_1 > t_0$) the receiver ACKs 1, 2, 3. At t_2 ($t_2 > t_1$) the sender times out and resends 1, 2, 3. At t_3 the receiver receives the duplicates and re-acknowledges 1, 2, 3. At t_4 the sender receives the ACKs that the receiver sent at t_1 and advances its window to 4, 5, 6. At t_5 the sender receives the ACKs 1, 2, 3 the receiver sent at t_2. These ACKs are outside its window.

b. (5%) With Go-Back-N (GBN), it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

True.

By essentially the same scenario as in (a).

c. (5%) The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.

True.

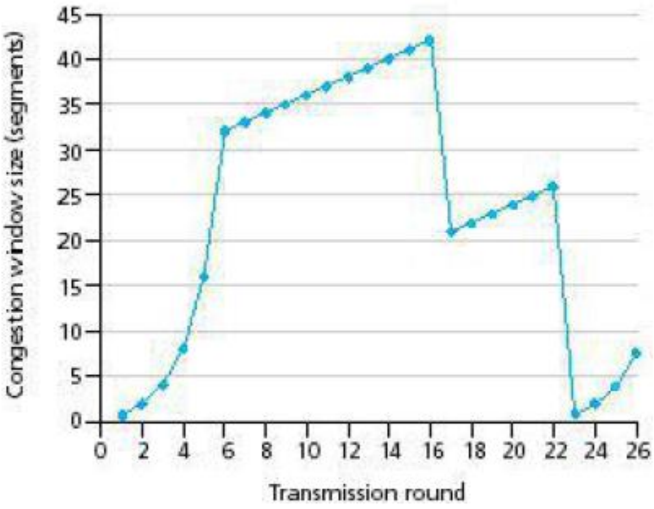
Note that with a window size of 1, SR, GBN, and the alternating-bit protocol are functionally equivalent. The window size of 1 precludes the possibility of out-of-order packets (within the window). A cumulative ACK is just an ordinary ACK in this situation, since it can only refer to the single packet within the window.

d. (5%) The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.

True.

The explanation is the same as (c).

4. (40%) Consider the Figure below.



Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.

a. (3%) Identify the intervals of time when TCP slow start is operating.
TCP slow start is operating in the intervals [1, 6] and [23, 26].

b. (3%) Identify the intervals of time when TCP congestion avoidance is operating.
TCP congestion avoidance is operating in the intervals [6, 16] and [17, 22].

c. (3%) After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
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. (3%) After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.

e. (4%) What is the initial value of *ssthresh* at the first transmission round?
The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.

f. (4%) What is the value of *ssthresh* at the 18th transmission round?
The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during 16th transmission round, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.

g. (4%) What is the value of *ssthresh* at the 24th transmission round?

The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during 22nd transmission round, the congestion windows size is 26. Hence the threshold is 13 during the 24th transmission round.

h. (4%) During what transmission round is the 70th segment sent?

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. (4%) Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of *ssthresh*?

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

j. (4%) Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the *ssthresh* and the congestion window size at the 19th round?

Threshold is 21. The congestion window size is first reset to 1, and then grows to 4 in 19th round.

k. (4%) Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

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