Problem 1

In the rdt3.0 protocol, the ACK packets that flow from the receiver back to the sender don't have sequence numbers, even though they do have an ACK field that contains the sequence number of the data packet that they are acknowledging. Why don't these ACK packets need sequence numbers?

Consider why we needed sequence numbers in the first place. We saw that the sender needs sequence numbers so that the receiver can tell if a data packet is a duplicate of an already received data packet. In the case of ACKs, the sender does not need this info (i.e., a sequence number on an ACK) to tell detect a duplicate ACK. A duplicate ACK is obvious to the rdt3.0 receiver, since when it has received the original ACK it transitioned to the next state. The duplicate ACK is not the ACK that the sender needs and hence is ignored by the rdt3.0 sender.

Problem 2

Consider the Go-Back-N (GBN) protocol with a sender window size (N) of 3 and a sufficiently large sequence number range. 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.

a. 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.

b. Following (a), what are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer.

a. Given that the receiver is waiting for packet k, then it has already 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 $[k-N, k-1]$ (i.e., $[k-3, k-1]$) may still be propagating back.

b. Let's discuss two boundary situation:

   Suppose *all* of these ACKs in (a) have been received by the sender, then sender's window is $[k, k+N-1]$ (i.e., $[k, k+2]$).

   Suppose *none* of the ACKs have been yet 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 $[k-N, k-1]$ (i.e., $[k-3, k-1]$).

   By these arguments, the senders window is of size 3 and begins somewhere in the range of $k-3$ and k.

Problem 3

Consider an HTTP client that needs to retrieve a single object of size $O = 100$ KBytes from an HTTP server. Assume we are using non-persistent HTTP. If the maximum segment size S is 536 Bytes, $RTT = 100$ msec, and the transport protocol is TCP with a static window size W:

a. For a transmission rate of 28 kbps (that's kilo*bits* per second), determine the minimum possible latency. Determine the minimum window size that achieves this latency.

b. Repeat (a) for 100 kbps.

c. Repeat (a) for 10 Mbps.

Problem 3 continued on next page...

The object consists of 100 KBytes/536 Byte ~ 187 packets (segments).
The minimum window needs to be greater than the (bandwidth-delay product/536) to fully utilize the link, and no greater than 187.

a. Latency = TCP setup + request/response = 0.1 sec + [0.1 sec + (800 kb / 28 kbps)] = 28.77 sec
   W $\geq$ 0.1 sec * 28 kbps / 536 byte + 1 = 1.65
   The minimum window size is 2.

b. Latency = 0.1 sec + 0.1 sec + (800 kb / 100 kbps) = 8.2 sec
   W $\geq$ 0.1 sec * 100 kbps / 536 byte + 1 = 3.33
   The minimum window size is 4.

c. Latency = 0.1 sec + 0.1 sec + (800 kb / 10000 kbps) = 0.28 sec
   W $\geq$ 0.1 sec * 10000 kbps / 536 byte + 1 = 234.20
   The minimum window size is 187 (235 > 187).

Usually the window size is expressed in *number of segments*.
Here we allow you to express it in bytes: 2*536 bytes, 4*536 bytes, 187*536 bytes.

Problem 4

This problem is about estimating round-trip times (RTTs). Given $\alpha = 0.1$, let $SampleRTT_n$ be the most recent sample RTT, let $SampleRTT_{n-1}$ be the next most recent sample RTT, and so on.

a. In a TCP connection, suppose four acknowledgments have been returned with corresponding sample RTTs $SampleRTT_1$, $SampleRTT_2$, $SampleRTT_3$, and $SampleRTT_4$. Express $EstimatedRTT$, the estimated RTT after all four ACKs have been received, in terms of the four sample RTTs.

b. Express this as a general formula for n sample RTTs.

a. If $EstimatedRTT_n$ is the estimate after the nth sample:

$$
\begin{aligned}
EstimatedRTT_1 &= SampleRTT_1 \\
EstimatedRTT_2 &= x \cdot SampleRTT_2 + (1-x) \cdot SampleRTT_1 \\
EstimatedRTT_3 &= x \cdot SampleRTT_3 + (1-x) \cdot EstimatedRTT_2 \\
&= x \cdot SampleRTT_3 + (1-x)[x \cdot SampleRTT_2 + (1-x) \cdot SampleRTT_1] \\
&= x \cdot SampleRTT_3 + (1-x) \cdot x \cdot SampleRTT_2 + (1-x)^2 \cdot SampleRTT_1 \\
EstimatedRTT_4 &= x \cdot SampleRTT_4 + (1-x) \cdot EstimatedRTT_3 \\
&= x \cdot SampleRTT_4 + (1-x) \cdot x \cdot SampleRTT_3 \\
&\quad + (1-x)^2 \cdot x \cdot SampleRTT_2 + (1-x)^3 \cdot SampleRTT_1
\end{aligned}
$$

b. $EstimatedRTT_n = x \cdot \sum_{j=2}^{n} (1-x)^{(n-j)} \cdot SampleRTT_j + (1-x)^{(n-1)} \cdot SampleRTT_1$