MPCS 54001, Winter 2021

Homework 3 - Solutions

Due: February 4 at 5:30 pm

Submit your assignment to Gradescope. This work must be entirely your own. If you need help, post questions to Ed Discussion and/or visit the staff during office hours. As a reminder, if you make a public post on Ed Discussion, please don't give away the answer!

---

1. UDP and TCP use 1s complement for their checksums. (*Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.*)

    a. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the 1s complement sum of these 8-bit bytes? Show all work. (10 points)

    1 1 0 1 0 0 0 1

    b. Why is it that UDP takes the 1s complement of the sum to detect errors at the receiver; that is, why not just use the sum? (10 points)

    This simplifies error detection at the receiver. To detect errors, the receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error.

    c. Is it possible that a 1-bit error will go undetected? What about a 2-bit error? (5 points)

    All one-bit errors will be detected, but two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1).

2. Host *A* and Host *B* are communicating over a TCP connection, and Host *B* has already received from Host *A* all bytes up through byte 126. Suppose Host *A* then sends two

segments to Host *B* back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port is 302, and the destination port number is 80. Host *B* sends an acknowledgment whenever it receives a segment from Host *A*.

a. In the first segment sent from Host *A* to Host *B*, what are the sequence number, source port number, and destination port number? (5 points)

   In the first segment from Host A to B, the sequence number is 127, source port number is 302 and destination port number is 80.

b. In the second segment sent from Host *A* to Host *B*, what are the sequence number, source port number, and destination port number? (5 points)

   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.

c. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number? (5 points)

   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.

d. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number? (5 points)

   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.

e. If the second segment arrives before the first segment, in the acknowledgment of the second arriving segment, what is the acknowledgment number? (5 points)

If the second segment arrives before the first segment, in the acknowledgement of the second arriving segment, the acknowledgement number is 247, as a cumulative ACK, indicating that it is waiting for bytes 247 and onwards.

3. Consider the following sequence of events (in chronological order) at a sender following the specified ARQ protocol with a sliding window size equal to 4 and an infinite amount of data to send. Write the corresponding sliding window range at the sender immediately after each event.

a. Go-Back-N sender (5 points)

Event *E*	Sliding Window Range after *E*
`SEND(0)`	{0, 1, 2, 3}
`SEND(1)`	{0, 1, 2, 3}
`ACK(0)`	{1, 2, 3, 4}
`SEND(2)`	{1, 2, 3, 4}

b. Go-Back-N sender (10 points)

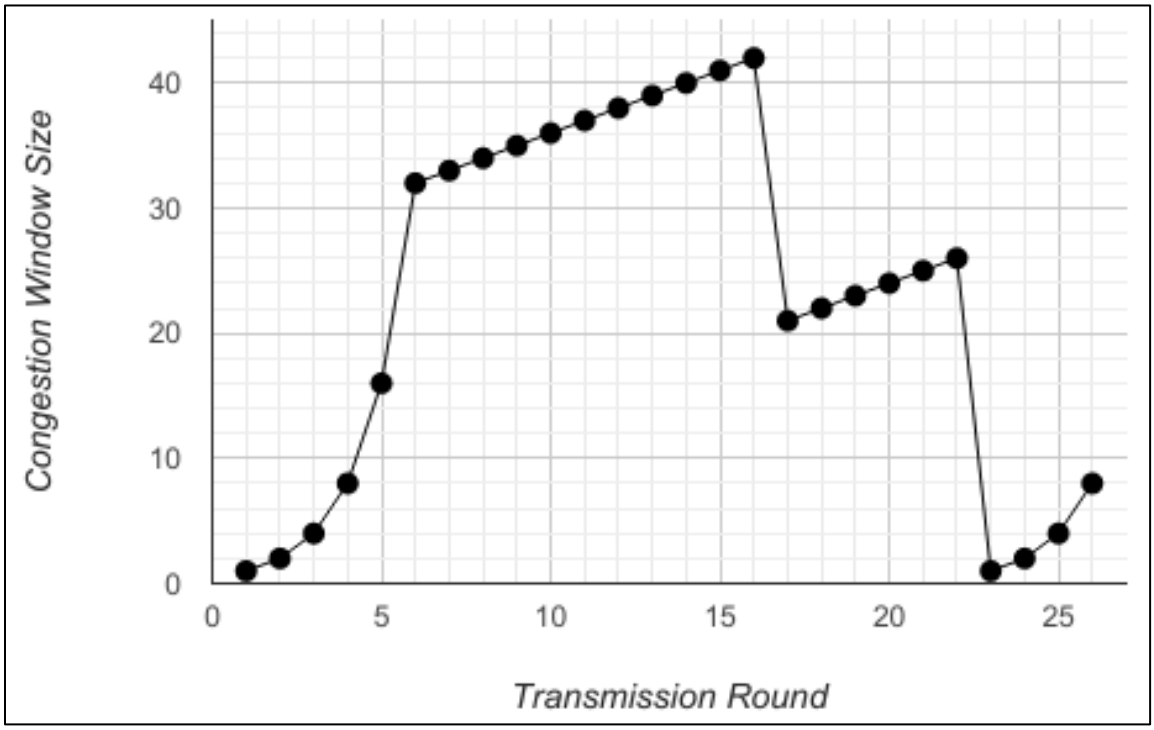
Event *E*	Sliding Window Range after *E*
`SEND(0)`	{0, 1, 2, 3}
`SEND(1)`	{0, 1, 2, 3}
`SEND(2)`	{0, 1, 2, 3}
`SEND(3)`	{0, 1, 2, 3}
`ACK(0)`	{1, 2, 3, 4}
`ACK(2)`	{3, 4, 5, 6}
`TIMEOUT(3)`	{3, 4, 5, 6}
`SEND(3)`	{3, 4, 5, 6}

c. Selective Repeat sender (10 points)

Event *E*	Sliding Window Range after *E*
`SEND(0)`	{0, 1, 2, 3}

SEND(1)	{0, 1, 2, 3}
SEND(2)	{0, 1, 2, 3}
SEND(3)	{0, 1, 2, 3}
ACK(0)	{1, 2, 3, 4}
ACK(2)	{1, 2, 3, 4}
TIMEOUT(1)	{1, 2, 3, 4}
SEND(1)	{1, 2, 3, 4}

4. Consider the following figure. Assuming that TCP Reno is being used for congestion control, answer the following questions. *Be sure to include explanations for your answers.*



a. Which interval(s) correspond(s) to TCP slow start? (5 points)

Slow start (i.e., exponential growth) occurs during (1,6) and (23,26).

b. Which interval(s) correspond(s) to TCP congestion avoidance? (5 points)

Congestion avoidance (i.e., linear growth) occurs during (6,16) and (17,22).

c. What is the initial value of `ssthresh` at the first transmission round? (5 points)

The initial value of `ssthresh` is 32 since that is where slow start ends and congestion avoidance begins.

d. Did a triple duplicate ACK or a timeout lead to the segment loss detection after the 22nd transmission round? (5 points)

Since the congestion window size dropped to 1, the segment loss was detected via a timeout.

e. What is the value of `ssthresh` at the 24th transmission round? (5 points)

The value of `ssthresh` would be 13 since the congestion window was 26 when the previous segment loss detection occurred.