Introduction to Computer Networks

Homework #3 Oct. 22 (10 points)

[1]

(a*) (2 points)* Suppose you have the following 2 bytes: 01011100 and 01010110. What is the 1s complement of the sum of these 2 bytes?

(b) Suppose you have the following 2 bytes: 11011101 and 00110110. What is the 1s complement of the sum of these 2 bytes?

(c) For the byte in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1a complement doesn't change.

[2] Consider the Stop-and-Wait protocol rdt3.0 and the sender side simply ignores (that is, takes no action on) all received packets that are either in error or have the wrong value in the ACK seq field of an acknowledgement packet. Suppose that the protocol is modified so that each time a frame is found in error, at either the sender or the receiver, the last transmitted frame is immediately resent.

Does the protocol still operate correctly? Explain why or why not. (Hint: consider what would happen if there were only bit errors; there are no packet losses but premature timeouts can occur. Consider how many times the nth packet is sent, in the limit as n approaches infinity.)

[3*] (2 points)* A channel has a data rate of 8 Kbps (bandwidth) and a propagation delay of 20ms. For what frame size does stop-and-wait given an efficiency of at least 75%? Ignore processing time at the receiver.

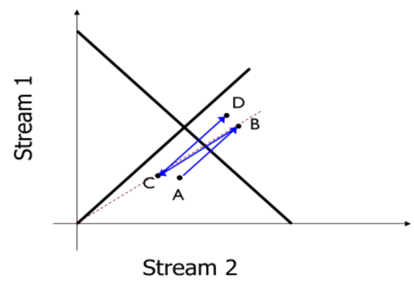
[4]. (2 points) Consider the GBN protocol with a sender window size of 3. Suppose that at time *t*, the next in-order packet that the receiver is expecting has a sequence number of *k*. Assume that the medium does not reorder messages. Answer the following questions:

(a). What are the possible sets of sequence number inside the sender's window at time t? Justify your answer

(b) 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.

Continued on reverse side

[5]. (2 points) Referring to the TCP's algorithm of linear increase and multiple decrease, suppose that instead of a multiplicative decrease, TCP decreased the window size by a constant amount. Would the resulting algorithm converge to an equal share algorithm? Support your answer using diagrams similar to the one show below. (Hint, consider two cases: (1) the constant amount for both connections is the same, (2) the ratio of the linear decrease on loss between connection 1 and connection 2 is 2:1)



[6] (2 points) Host A and Host B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 248. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 40 and 60 bytes of data, respectively. In the first segment, the sequence number is 249, the source port number 503, and the destination port number is 80. Host B sends an ACK whenever it receives a segment from Host A.

(a) In the second segment sent from A to B, what are the sequence number, source port number, and destination port number?

(b) If the first segment arrives before the second segment, in the ACK of the first arriving segment, what is the ACK number, the source port number, and the destination port number?

(c) If the second segment arrives before the first segment, in the ACK of the first arriving segment, what is the ACK number?

(d) Suppose the two segments sent by A arrive in order at B. The first ACK is lost and the second ACK Arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and ACKs sent. Assume there is no additional packet loss. For each segment in your diagram, show the sequence number and the number of bytes of data. For each ACK that you add, provide the ACK number.