

第三章作业

1.

P4.

- Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1s complement of the sum of these 2 bytes?
- Suppose you have the following 2 bytes: 11011010 and 01100101. What is the 1s complement of the sum of these 2 bytes?
- For the bytes in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1s complement doesn't change.

2.

P24. Answer true or false to the following questions and briefly justify your answer:

- With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.
- With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.
- The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.
- The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.

3.

P26. Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 536 bytes.

- What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.
- For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

4.

P27. Host A and B are communicating over a TCP connection, and Host B has already received from 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 number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

- In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?
- 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?
- If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?
- Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.

5.

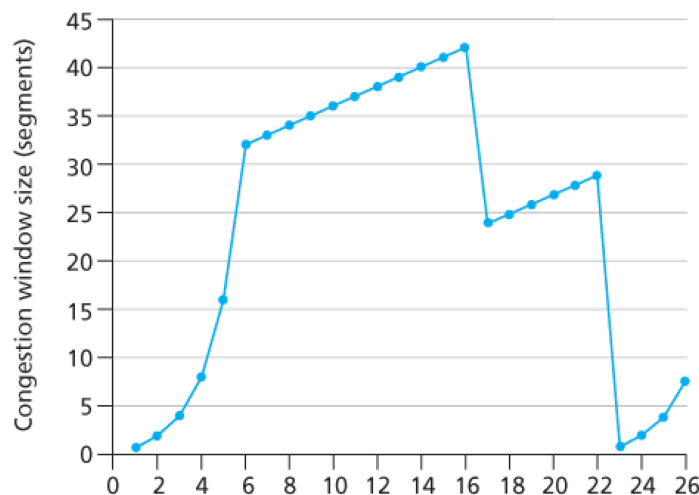
P37. Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host A sends 5 data segments to Host B, and the 2nd segment (sent from A) is lost. In the end, all 5 data segments have been correctly received by Host B.

- How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.
- If the timeout values for all three protocol are much longer than 5 RTT, then which protocol successfully delivers all five data segments in shortest time interval?

6.

P40. Consider [Figure 3.58](#). 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.

- Identify the intervals of time when TCP slow start is operating.
- Identify the intervals of time when TCP congestion avoidance is operating.
- After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
- After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?



- What is the initial value of `ssthresh` at the first transmission round?
- What is the value of `ssthresh` at the 18th transmission round?
- What is the value of `ssthresh` at the 24th transmission round?
- During what transmission round is the 70th segment sent?
- 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`?
- 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?
- 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?

6.

P44. Consider sending a large file from a host to another over a TCP connection that has no loss.

- a. Suppose TCP uses AIMD for its congestion control without slow start. Assuming *cwnd* increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant round-trip times, how long does it take for *cwnd* increase from 6 MSS to 12 MSS (assuming no loss events)?
- b. What is the average throughput (in terms of MSS and RTT) for this connection up through time=6 RTT?

7.

P51. Consider the network described in the previous problem. Now suppose that the two TCP connections, C1 and C2, have the same RTT of 100 msec. Suppose that at time t_0 , C1's congestion window size is 15 segments but C2's congestion window size is 10 segments.

- a. What are their congestion window sizes after 2200 msec?
- b. In the long run, will these two connections get about the same share of the bandwidth of the congested link?
- c. We say that two connections are synchronized, if both connections reach their maximum window sizes at the same time and reach their minimum window sizes at the same time. In the long run, will these two connections get synchronized eventually? If so, what are their maximum window sizes?
- d. Will this synchronization help to improve the utilization of the shared link? Why? Sketch some idea to break this synchronization.