P21. Suppose we have two network entities, A and B. B has a supply of data messages that will be sent to A according to the following conventions. When A gets a request from the layer above to get the next data (D) message from B, A must send a request (R) message to B on the A-to-B channel. Only when B receives an R message can it send a data (D) message back to A on the B-toA channel. A should deliver exactly one copy of each D message to the layer above. R messages can be lost (but not corrupted) in the A-to-B channel; D messages, once sent, are always delivered correctly. The delay along both channels is unknown and variable. Design (give an FSM description of) a protocol that incorporates the appropriate mechanisms to compensate for the loss-prone A-to-B channel and implements message passing to the layer above at entity A, as discussed above. Use only those mechanisms that are absolutely necessary.

P22. Consider the GBN protocol with a sender window size of 4 and a sequence number range of 1,024. 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 numbers 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.

//P23. Give one example where buffering out-of-order segments would significantly improve the throughput of a GBN protocol.

//P24. Consider a scenario where Host A, Host B, and Host C are connected as a ring (i.e., Host A to Host B, Host B to Host C, and Host C to Host A). Assume that Host A and Host C run protocol rdt3.0, while Host B simply relays all messages received from Host A to Host C. Does this arrangement enable reliable delivery of messages from Host A to Host C? Can Host B tell if a certain message has been correctly received by Host A?

//P25. Consider the Telnet case study in Section 3.5.2. Assume a Telnet session is already active between Host A and Server S. The user at Host A then types the word “Hello.”   
a. How many TCP segments will be created at the transport layer of Host A?   
b. Is there any guarantee that each segment will be sent into the TCP connection as soon as it is created?   
c. Does TCP provide any mechanism that can be useful for an interactive Telnet session?   
d. Would UDP offer a viable alternative to TCP for Telnet sessions over a reliable channel?

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

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.   
a. In the second segment sent from Host 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 acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?   
c. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number? d. 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.

P28. Host A and B are directly connected with a 100 Mbps link. There is one TCP connection between the two hosts, and Host A is sending to Host B an enormous file over this connection. Host A can send its application data into its TCP socket at a rate as high as 120 Mbps but Host B can read out of its TCP receive buffer at a maximum rate of 50 Mbps. Describe the effect of TCP flow control.

P29. SYN cookies were discussed in Section 3.5.6. a. Why is it necessary for the server to use a special initial sequence number in the SYNACK? b. Suppose an attacker knows that a target host uses SYN cookies. Can the attacker create half-open or fully open connections by simply sending an ACK packet to the target? Why or why not? c. Suppose an attacker collects a large amount of initial sequence numbers sent by the server. Can the attacker cause the server to create many fully open connections by sending ACKs with those initial sequence numbers? Why?

P30. Consider the network shown in Scenario 2 in Section 3.6.1. Suppose both sending hosts A and B have some fixed timeout values. a. Argue that increasing the size of the finite buffer of the router might possibly decrease the throughput (lout). b. Now suppose both hosts dynamically adjust their timeout values (like what TCP does) based on the buffering delay at the router. Would increasing the buffer size help to increase the throughput? Why?

P31. Suppose that the five measured SampleRTT values (see Section 3.5.3) are 106 ms, 120 ms, 140 ms, 90 ms, and 115 ms. Compute the EstimatedRTT after each of these SampleRTT values is obtained, using a value of α = 0.125 and assuming that the value of EstimatedRTT was 100 ms just before the first of these five samples were obtained. Compute also the DevRTT after each sample is obtained, assuming a value of β = 0.25 and assuming the value of DevRTT was 5 ms just before the first of these five samples was obtained. Last, compute the TCP TimeoutInterval after each of these samples is obtained.

P32. Consider the TCP procedure for estimating RTT. Suppose that α = 0.1. Let SampleRTT1 be the most recent sample RTT, let SampleRTT2 be the next most recent sample RTT, and so on.   
a. For a given TCP connection, suppose four acknowledgments have been returned with corresponding sample RTTs: SampleRTT4, SampleRTT3, SampleRTT2, and SampleRTT1. Express EstimatedRTT in terms of the four sample RTTs.  
b. Generalize your formula for n sample RTTs.   
c. For the formula in part (b) let n approach infinity. Comment on why this averaging procedure is called an exponential moving average.

P33. In Section 3.5.3, we discussed TCP’s estimation of RTT. Why do you think TCP avoids measuring the SampleRTT for retransmitted segments?

P34. What is the relationship between the variable SendBase in Section 3.5.4 and the variable LastByteRcvd in Section 3.5.5?

P35. What is the relationship between the variable LastByteRcvd in Section 3.5.5 and the variable y in Section 3.5.4?

P36. In Section 3.5.4, we saw that TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first duplicate ACK for a segment is received?

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.   
a. 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.  
b. 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?

P38. In our description of TCP in Figure 3.53, the value of the threshold, ssthresh, is set as ssthresh=cwnd/2 in several places and ssthresh value is referred to as being set to half the window size when a loss event occurred. Must the rate at which the sender is sending when the loss event occurred be approximately equal to cwnd segments per RTT? Explain your answer. If your answer is no, can you suggest a different manner in which ssthresh should be set?

P39. Consider Figure 3.46(b). If l′in increases beyond R/2, can lout increase beyond R/3? Explain. Now consider Figure 3.46(c). If l′in increases beyond R/2, can lout increase beyond R/4 under the assumption that a packet will be forwarded twice on average from the router to the receiver? Explain.

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