1. (5 pts) Show a general expression of the maximum number of bits in the link at any given time for the distance \( d \), the propagation speed \( s \), and the transmission rate \( R \). Use this expression, and compute the maximum number of bits in the link at any given time.

2. (5 pts) What is the maximum window size at the sender A, and why? Also, explain how A computes this size in detail.

3. (8 pts) Packets may get lost or delayed due to network congestion as they travel from A to B. Answer the following questions.
   1) How can the TCP sender at A know packet losses?
   2) TCP retransmits a packet when it is lost instead of correction. Would you do the same if you design the protocol? Why or why not?
   3) How can the TCP sender at A match its sending rate to the network bottleneck link rate?
   4) TCP congestion control has two phases: slow start and congestion avoidance. Why the two phases? Why not just slow start or congestion avoidance?

4. (8 pts) Suppose that A and B start communication using TCP three-way handshake.
   1) Give an example with specific sequence numbers.
   2) Why not two-way handshake?
   3) Explain what kind of denial-of-service attack is possible, and how, if two-way handshake is used.
   4) How can we defend such a denial-of-service attack?

5. (4 pts) Study the TCP and UDP packet header, and discuss what each field does.

6. (10 pts) Suppose that A sends packets to B starting from pkt 1, following the go-back-n protocol with a sender window size of 3. After A sent pkt 99, the following is the sequence of actions that have happened so far:
   
   A: send pkt 100
   B: receive pkt 100, send ack 101
   A: send pkt 101
   B: receive pkt 101, send ack102

   1) After sufficient time (longer than delay from B to A), acks 101 and 102 from B may or may not arrive at A. What is the set of sequence numbers possible at A’s cwnd at this point (when B awaits pkt 102)? Justify your answer.
2) When B waits for pkt 102, what are all possible ack values sending back to A from B? Justify your answer.

7. (15 pts) Show TCP NewReno’s execution, when the initial cwnd = 5, the sequence number begins at 0, and segment 4 is lost in transit, until the exit of fast recovery.

8. (15 pts) Show TCP Reno’s execution for the same as Q 7 except that segments 1, 2, and 3 are lost.

9. (15 pts) In multimedia networking, providing QoS is critical, and one of the QoS techniques is policing that regulates the rate of network flows. First, study a policing technique called the leaky bucket, and then solve the following problem:

Packets arrive: 1, 2, 3 at t=0
4 at t=1
5 at t=2
6 at t=3
7, 8 at t=6
9, 10 at t=7

In the leaky bucket, one new token is generated per slot. Assume that link speed is infinite, and the bucket capacity is unlimited. Assume that delays except queueing are negligible. Initially no token is in the bucket.

Q: Packet 1 can leave immediately after arrival using a token just generated at t=0. Compute the time slot that each packet departs the system.

10. (15 pts) Assuming that we run a number of TCP flows with RED at routers, we will consider the following two cases with a total of 1000 packets. In the first case, we have a certain queue length caused by 500 flows, each with a window of 2 packets. Dropping one packet in RED decreases the congestion window of only one TCP flow. All the other 499 TCP flows will increase their congestion windows. In the second case, we have only one flow with a window of 1000 packets. Dropping one packet would reduce that window to 500, and it would take 500 round trip times for the window to grow back to 1000 packets. The discrepancy between these two cases arises mainly because the number of active flows is an important entity in active queue management. Modify the packet drop probability function in RED so that this discrepancy can be mitigated. (Hint: the number of active flows should be a parameter in computing the packet drop probability in RED).