A Congestion Control

1. At time $t$, a TCP connection has a congestion window of 4000 bytes. The maximum segment size used by the connection is 1000 bytes. What is the congestion window after it sends out 4 packets and receives ACKs for all of them...

(a) If the connection is in slow-start?

(b) If the connection is in congestion avoidance (AIMD mode)?
2. In congestion avoidance (CA) mode, a TCP sender increases the size of its congestion window by one 
maximum segment size (MSS) each RTT. Suppose a TCP implementation does this by increasing its 
congestion window by $\Delta$ every time it receives an ACK, where 

$$\Delta = \frac{\text{MSS}}{\text{current window size}}$$

Knowing this, how can a greedy TCP receiver get more than its fair share of the link bandwidth? (*Hint:* 
Remember that the sequence number acknowledged by an ACK doesn’t refer to a *packet*, but rather to 
a *byte* in the data stream.)

3. David is implementing XCP. After reading that XCP rarely drops packets (less than one in a million), 
he decides that ACKs are less critical than they are in TCP. Rather than acknowledging each correctly 
received packet, he decides an XCP receiver should use *negative acknowledgements* (NACKs) to explicitly 
inform the sender when it is missing a particular range of bytes. What is he overlooking? 
(Assume that the last packet in a stream is never lost — that is, the problem is not that the receiver 
doesn’t know it should NACK the last packet if it is lost.)

4. True or False.

   T  F  TFRC measures loss event rates using the last $N$ loss interval periods.
   T  F  TFRC supports multimedia better than TCP since it adapts to available bandwidth changes 
more quickly than TCP.
   T  F  TFRC transmission rate is based solely on the loss rate and RTT of the connection.
   T  F  TFRC uses a sliding window and ACK clocking to ensure that packets are sent out into the 
network at the computed rate.
5. RED gateways can optionally be run in “byte mode,” where the average queue size is measured in bytes instead of packets. If David is using SSH, would he prefer routers to operate in byte mode or not? Why?

B Quality of Service

6. A network uses routers with fair queueing. This is not weighted — all weights are the same.

(a) Two connections share the same congested gateway and have no other congested gateways. Connection A has RTT 5ms, connection B has RTT 10ms. Express the throughput of connection B ($tput_B$) in terms of the throughput achieved by connection A ($tput_A$), or indicate if there is no relationship between the two.

(b) Two connections traverse the same congested gateway, but also traverse some other unshared congested gateways. Express the throughput of connection B ($tput_B$) in terms of the throughput achieved by connection A ($tput_A$), or indicate if there is no relationship between the two.

7. Matt thinks that fair queueing is a great idea! Every flow gets a fair share of bandwidth. He is sad to find out it isn’t widely implemented. Why isn’t fair queueing used across the whole Internet?
C Routing Protocols

8. In link state routing protocols, routers broadcast link state announcements to the rest of the network containing information about which routers they are directly connected to. Each router collects these announcements from every other router and uses them to build a graph of the network.

(a) Routers store these link state announcements with a time to live (TTL); an announcement is deleted when 1) a more recent announcement from the same router arrives and replaces it or 2) the TTL expires. Why is this TTL timeout necessary?

(b) Link state announcements have sequence numbers to ensure that a router never replaces newer information with stale information from an older message still circulating in the network. Unlike other protocols, these sequence numbers are not expected to wrap around (and so they must be fairly long, e.g., 64 bits). A problem can arise when a router crashes, reboots, and begins sending link state announcements starting at sequence # 0. How does the network deal with this situation (assuming that routers do not store the current sequence number in non-volatile storage so that it is available after a reboot)?
9. The *count-to-infinity* is a common problem with distance vector routing protocols. One partial solution is a heuristic known as *split horizon with poison reverse* (See Wikipedia).

(a) Draw a simple network and give a scenario in which that network changes and split horizon with poison reverse prevents count to infinity. (*Note:* This requires neither a complicated network nor a complicated scenario.)

(b) Draw a simple network and give a scenario in which that network changes and split horizon with poison reverse *does not* prevent count to infinity. (Again, you can do this with a simple network and straightforward scenario.)
D  Router Design

10. Consider the iSLIP crossbar scheduling algorithm.
   (a) For a router with \( N \) ports, what is the maximum number of iterations iSLIP could take to complete?

   (b) Place packets in the virtual output queues (VOQs) below such that during the next time slot the iSLIP algorithm takes 2 iterations to complete and each output is given a packet to transmit. Assume that input 1 is the next input in both output 1’s and output 2’s round robin schedule.

   (c) Place packets in the virtual output queues (VOQs) below such that during the next time slot the iSLIP algorithm takes 1 iteration to complete and each output is given a packet to transmit. Assume that input 1 is the next input in both output 1’s and output 2’s round robin schedule.