1. Sequence numbers:
(a) In Stop-and-Wait protocols, how many sequence numbers are needed? How many bits are needed to represent the sequence numbers?
Since only one packet can be unacknowledged in Stop-and-Wait protocols, there only needs to be 2 sequence numbers since the values can be alternated for consecutive packets. The sequence numbers can be represented with only one bit.

(b) In pipelined protocols, using a sequence number field of 7 bits, how many different sequence numbers can be represented? What is their range?
7 bits can be used to represent 128 values. The range is from 0 to 127.

2. Suppose that 2 hosts A and B are using rdt 3.0 to communicate reliably. During the communication, host A sends segment with sequence number = "x" to host B. Host B receives segment "x" and sends an acknowledgement for it (ACK "x"). Suppose that ACK "x" is lost; host A times out and retransmits segment "x". What does host B do when it receives segment "x"?
Host B will drop segment “x” since it is a duplicate and will retransmit the ACK for it (ACK “x”).

3. The statement: "In TCP, a connection does not have to be closed by both sides at the same time." is true or false? Explain. True. Either side can initiate a connection close with a FIN packet, and wait for the other side to reply with an ACK packet. Once the other side is also done with the connection, the other side will send a FIN packet and wait until it receives an ACK before closing the connection.

4. During connection establishment, the connection initiator sends a "SYN segment" and a sequence number.
   i. What is the purpose of this sequence number? Explain.
   The transmitted sequence number, known as an “initial sequence number”, tells the other end of the connection (e.g., host B) what will be the first sequence number used for the data sent from the initiator (e.g., host A) to host B.

   ii. When replying to the SYN segment, the other side of the connection sends a "SYN-ACK" and a sequence number. What's the purpose of this sequence number? Is its value the same as the sequence number sent by the connection initiator? This sequence number is the “initial sequence number” for the data transmitted from host B to host A. This sequence number is not the same as the initial sequence number sent by A, the connection initiator.

5. TCP uses different types of acknowledgements. List them and explain their function and how they work.
   - Cumulative ACKs are used by TCP to limit the amount of control traffic generated. The basic idea is to use a single acknowledgement to confirm receipt of multiple segments. So, when ACK “x” is received, it indicates that segments up to “x-1” have been received and that segment “x” is next segment being expected by the connection endpoint that generated the ACK.
   - Duplicate ACKs in TCP are used to indicate that out-of-order segments have been received; out-of-order segments may be caused by data loss. If the TCP sender receives 3 duplicate ACKs, it will retransmit the segment assuming that it was lost.
• Delayed ACKs, similarly to cumulative ACKs, are also used by TCP as a way to limit the amount of feedback generated by the operation of the protocol. The way Delayed ACKs work is as follows: upon receiving a segment, the TCP receiver does not immediately generate an ACK; it tries to “piggyback” the ACK on a data segment going in the reverse direction. It uses the “delayed ACK” timer to make sure it will send the ACK in the case there is no segment to be transmitted in the other direction.

6. TCP assumes that packet loss is an indication of congestion. Is that a reasonable assumption? Why or why not? It’s a reasonable assumption in networks where transmission losses are not likely to occur. This is usually the case of “wired” networks, where packet losses usually result from congestion at routers. Thus, the TCP sender, when it detects that a segment was lost, it will not only retransmit that segment but also perform congestion control by backing off its sending rate.

7. In the computation of TCP’s RTT, how can we make TCP be more responsive to the current conditions of the network? Using the book’s notation, the formula for calculating the RTT is: 
   
   \[ \text{EstimatedRTT} = (1 - \alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT} \]
   
   By choosing a larger alpha, current network conditions (as indicated by SampleRTT) will have higher weight and consequently greater influence on the value of EstimatedRTT.

8. TCP congestion control switches between two different "modes". Explain how each of them works and how they work together.
   
   The two different “modes” are slow start and congestion avoidance. When a connection first starts, the TCP sender uses slow start to increase its congestion window which starts at 1 Maximum Segment Size (MSS) and increases by 1 MSS for every acknowledgement received. This means that, under slow start, the congestion window will increase exponentially. Slow start’s main purpose is to for the TCP sender to quickly find the available bandwidth, resulting in efficient utilization of the network. Slow start ends when the congestion window is greater than the ssthresh value or congestion is detected, i.e., a packet loss occurs; at that point, congestion avoidance begins.
   
   Under congestion avoidance, the TCP sender increases its congestion window by 1 MSS per RTT. This results in increasing the congestion window linearly. When a timeout occurs, the value of the congestion window is set back to 1 MSS, and the value of ssthresh is updated to half the value of the congestion window when the loss event occurred. Then, congestion avoidance stops and slow start begins again. If instead of a timeout, the TCP sender receives 3 duplicate ACKs, it will retransmit the lost segment, and, instead of setting its congestion window to 1MSS and going through slow start, it cuts its window in half.