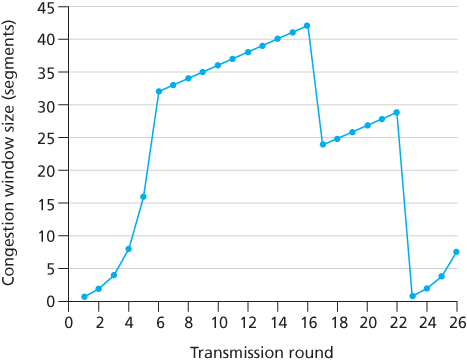
Read chapter 3 of your textbook. The following review questions should help guide your reading. Points possible 30

For each question, you should give a correct answer (as best you know it) or provide an intelligent question regarding the reading that applies to this question and explains why you could not answer the question. (Please note that "I didn't understand any of this" isn't a question, isn't intelligent, and has spelling and grammar errors. It will receive 0 points.)

Make sure to show your work. Answer the questions (best in different color**) using your own words** and **understanding**s on the chapter materials.

1. R6. Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how? **(2 Points)**
   1. **It is possible to have reliable data transfer when the application developer has included the necessity for the reliable data transfer protocol.**
2. R8. Suppose that a Web server runs in Host C on port 80. Suppose this Web server uses persistent connections, and is currently receiving requests from two different Hosts, A and B. Are all of the requests being sent through the same socket at Host C? If they are being passed through different sockets, do both of the sockets have port 80? Discuss and explain. **(Make sure to include diagram of your explanation) (3 Points)**
   1. **They are all being passed through port 80.**
3. R10. In our **rdt** protocols, why did we need to introduce timers? **(2 Points)**
   1. **We need a timer for this protocol because it is a stop and wait protocol. We need to know for how long data is taking to travel to see if we need to resend packets before we slow it down**
4. R12. Visit the Go-Back-N Java applet at the companion Web site. [Link is here](https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/go-back-n-protocol/index.html) **(5 Points)**
   1. Have the source send five packets, and then pause the animation before any of the five packets reach the destination. Then kill the first packet and resume the animation. Describe what happens.
      1. **If the first packet is killed off, none of the following packets are received at this destination, they all need to be sent again.**
   2. Repeat the experiment, but now let the first packet reach the destination and kill the first acknowledgment. Describe again what happens.
      1. **The packets are all acknowledged including the one that was killed off as long as a following ack is eventually received by the sender.**
   3. Finally, try sending six packets. What happens?
      1. **It will not let us send six packets as the window size is limited to 5.**
5. R13. Repeat R12, but now with the [Selective Repeat Java applet](https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/selective-repeat-protocol/index.html). How are Selective Repeat and Go-Back-N different? **(2 Points)**
   1. **Selective repeat still accepts all ack and recognizes that if there is a packet loss, that the packets are received out of order.**
6. R15. Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110. **(2 Points)**
   1. How much data is in the first segment?
      1. **20 bytes because the first sequence is 90 and the second is 110.**
7. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number? **(2 Points)**
   1. **The ack number will be the same as the sequence number, 110.**
8. P7. In protocol rdt3.0, the ACK packets flowing from the receiver to the sender do not have sequence numbers (although they do have an ACK field that contains the sequence number of the packet they are acknowledging). Why is it that our ACK packets do not require sequence numbers? **To best answer this question, consider why we needed sequence numbers in the first place. (2 Points)**
   1. **We have sequence numbers to recognize duplicate packets. In rdt3.0 we do not care if we have duplicate packets so it is not needed.**
9. 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. **(5 Points)**
   1. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?
      1. **Sequence: 127+80=207**
      2. **Source port number: 302**
      3. **Destination port number: 80**
   2. 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?
      1. **Acknowledgment number: 207**
      2. **Source port number: 80**
      3. **Destination port number: 302**
   3. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?
      1. **It is still 127**
   4. 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.
10. 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. **(5 Points)**



1. Identify the intervals of time when TCP slow start is operating.
   1. **Between 1-33, 23-26**
2. Identify the intervals of time when TCP congestion avoidance is operating.
   1. **Between 33-42, 17-22.**
3. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
   1. **Triple duplicate ACK as the cwnd was not dropped to 1. And the new threshold is dropped to 43/2**
4. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
   1. **It was by a timeout as the cwnd was dropped to 1.**
5. What is the initial value of ssthresh at the first transmission round?
6. What is the value of ssthresh at the 18th transmission round?
   1. **25**
7. What is the value of ssthresh at the 24th transmission round?
   1. **3**