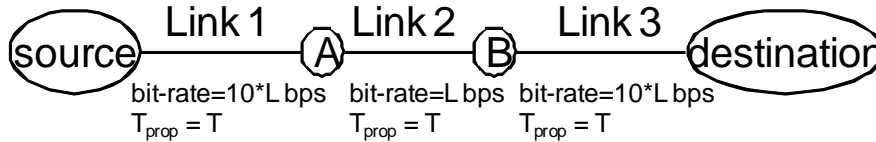


CPEG 419

1. page 293: 3, 4, 5,6, 7,9,10,14
 2. Page 295: 5, 14, 23, 31, 33, 35, 39
 3. Make a state diagram of TCP that shows the size of the congestion window and includes timeout, ACK arrivals. Neglect the receiver window.
 4. Draw a detailed timing diagram that shows the beginning of the file transfer where the 4th packet is lost. Make the diagram until the 11th packet is transmitted.
 5. Suppose that a host sends a TCP packets and each packet has exactly one byte of data. Now suppose that the host sent a packet with sequence number equal to 10. When this packet reaches the destination, the destination generates an ACK. Which of the following ACK numbers might the destination generate. Make a timing diagram for each case.
 - (a) Ack Number = 9
 - (b) Ack Number = 10
 - (c) Ack Number = 11
 - (d) Ack Number = 12
 6. Assume that a TCP connection has a window size of 8 and there are no unacknowledged packets. Show the timing diagram for TCP-RENO when the fourth packet send is lost. Show the timing until the window reaches 5. By timing diagram, we mean a diagram with two vertical lines showing time going down with one line for the source and one line for the destination.
 7. We can define the average data rate to be the file of the file transfered divided by the time it takes to transfer the file. Of course, this data rate will depend on the file of the file. Suppose $MSS=1500B$, the initial window size is $1MSS$, the maximum receiver window size is $20KB$, RTT is $100ms$, $SSThresh=40KB$, and there is a single $100Mbps$ link between the source and destination.
 - (a) What is the data rate when transferring a file of size $1500B$
 - (b) What is the data rate when transferring a file of size $15000B$
 - (c) What is the data rate when transferring a file of size $150000B$
 - (d) a, b, and c, but for $RTT = 500ms$.
Don't forget connection establishment. However, the closing of the connection should not be included, since the file transfer is complete before the connection is closed.
- Extra Credit**
8. SYN Attack
 - (a) Suppose that the initial value of RTO is 3 sec., and RTO is multiplied by two if a packet loss occurs.
 - (b) Suppose that a TCP server will only resend a SYN-ACK four times if an ACK does not arrive before RTO after the previous SYN-ACK was sent.
 - (c) Suppose that the server reserves $100KB$ of memory when a SYN arrives, and frees the memory after the connection ends, the connection is reset, or the connection fails (e.g., if no response for the SYN-ACK arrives).
 - (d) Finally, suppose that the server is connected via $10 Mbps$ link and SYN packets are $40B$.
 - (e) How large must be server's memory be to withstand a SYN attack?

9. Consider the topology below



Suppose that there is a single TCP connection from the source to the destination and that this TCP connection is being used to send an infinite number of bytes to the destination (i.e., the connection never stops).

- Show that if $cwnd = \text{bandwidth} \cdot \text{delay}$ product, then there are no packets in any buffer. To do this, account for all packets would be located.
- Show that if $cwnd = 2 \times \text{bandwidth} \cdot \text{delay}$ product, then the number of packets in the buffer of the bottleneck link is exactly equal to the bandwidth-delay product.
- Show that if the size of the queue at in node A for the interface of link 2 is equal to the bandwidth-delay product, then link 2 is fully utilized.

10. Wireshark

Download and install wireshark. Start wireshark. Press the "Capture" menu and select "interfaces." In the list interfaces, find the interface where packets are being received. the interface might have a name that includes ethernet or something similar. Ignore names like VPN. The active interface should show an IP address and non-zero packets. For this interface, select "start." The dialog box should close. Next in the filter box (in the upper left), enter the expression `tcp.port eq 80`. This display filter will filter all packets except those with a TCP port or 80. Now, with wireshark running, download a web page like `nytimes.com` (which has IP address = `199.239.136.200`). Wireshark should show a large number of packets. Select the capture menu and select stop.

There are three screens in wireshark. The top one shows high-level information about the packet. It shows the time, the source and destination of the packet, and the protocol. the middle screen shows some decoded information about the packet. and the bottom shows the a raw dump of the packet.

In the upper window, find the TCP-SYN packet sent from your machine to `199.239.136.200` (if you downloaded `nytime.com`). Click on the line that has the syn packet (it should even say that is it a syn packet somewhere in the line). Note that in the middle window, you can expand the "transmission control protocol" so see details about the packet. For example, you can see that the syn flag is set. Record the time when this syn packet was sent. (To make things easier, you might want to right click on the packet in the upper window and select colorize conversion -> TCP and pick a color. this way all packets in this connection will have the same color.

Now, record the time when packets and ACKs are sent and received. Try to determine the window size. Also, try to decode what is going on in the transport layer. Do this for at least 10 packets.