Homework – 4

Rahul Chowdary Garigipati – 800765549.

1) What do we mean when we say that TCP is self-clocking? Why is TCP self-clocking?

Ans.) TCP is 'self-clocking,' i.e., the source sends a new packet only when it receives an ack for an old one and the rate at which the source receives acks is the same rate at which the destination receives packets. So, the rate at which the source sends matches the rate of transmission over the slowest part of the connection.

In other words, TCP is self-clocking because it never sends more data than the network can handle.

2) Why do TCP and UDP messages use ports? Describe a situation where two computers might need to have two separate TCP conversations running at the same time.

Ans.) A computer may have many network processes that they all need to have network conversations at the same time.

An Operating system needs to find a way to keep track of all the network conversations, so, they give them each a number between 0 to 2^{16}. And hence, the Operating Systems have two separate lists – UDP ports and TCP ports.

When the two computers are connected to the LAN, then when computer A wants to access the network attached storage and when computer B wants to access the network devices. Then the two computers need to have two separate TCP conversations running at the same time.

3) Consider the diagram in Figure 1. Answer with at least a full sentence.
   a) Which TCP flavor is being used for each of these TCP connections? How can you tell?

   Ans.) The orange line is “TCP Reno”. We can identify by the Additive increase and the Multiplicative decrease of the line.

   The blue line is also “TCP Reno”. We can identify by the Additive increase and the Multiplicative decrease of the line.

   b) At what times do these TCP conversations detect congestion?

   Ans.) The blue line detects the congestions at 15:04:11, 15:04:15, 15:04:21.

   The orange line detects the congestions at 15:04:11, 15:04:15.

   We can detect a TCP congestion when there`s a change in the transmission rate.

   c) How did these two TCP conversations know there was congestion? Reference the self-clocking property of TCP in your answer.

   Ans.) the source sends a new packet only when it receives an ack for an old one and the rate at which the source receives acks is the same rate at which the destination receives packets. So, the rate at which the source sends matches the rate of transmission over the slowest part of the connection. Hence the TCP conversations know that there was congestion when the rate of transmission speed reduces.

4) Consider the diagram in Figure 1. Answer with at least a full sentence.
   a) The transmission speeds of these two conversations are getting closer and closer together as time goes on. Why is that happening?

   Ans.) That means that there is a problem at one of the destinations network as the both the conversations are gradually decreasing their transmission speeds. That means that there is a congestion, It is happening as I am sending more packets than the ACK`s received.

   b) If a third TCP conversation started up using the exact same TCP flavor with the same rules, would the same thing happen? Why or why not?

Ans.) It depends, if the problem at the destinations network is resolved then the transmission speed will not decrease as there is no congestion anymore. If not then the same thing happens.

5) Consider the diagram in Figure 2.
   a) What is happening in the section labelled “A"?

   Ans.) A 3-way handshake is going on in the section labelled "A”.

   b) From looking at the section labelled “B", what would you guess that Computer B's sending window size is in Kb? Why?

   Ans.) The sending window size of computer B is 40 Kb, as computer B has sent its maximum buffer of data and awaiting for the reply from computer A, tells us that the sending window size of computer B is 40 Kb.

   c) From looking at the section labelled “B", what would you guess that Computer A's receiving window size is in Kb? Why?

   Ans.) The receiving window size of computer A is 40 Kb, because in TCP computer A`s receive window is computer B`s sending window, and vice versa.

6) Consider the diagram in Figure 2.
   a) What goes wrong in the section labelled “C"?

   Ans.) In section C, it shows that when the computer B is sending data to the computer A, then some part of the transmitting data gets lost and then the computer A sends the same Acknowledge back to the computer B, then again computer B has to start over from which transmission window has gone wrong and retransmit the data again, this is called as TCP Retransmission.

   b) How does Computer B know something has gone wrong?

   Ans.) When computer B transmits the data to the computer A and computer A sends back the ACK`s to computer B, if the ACK`s are different to what the computer B sends then the computer B knows that something went wrong and starts retransmitting the data that computer A couldn`t receive.

   When the computer B sends the data and while transmitting, the data gets lost then the computer A sends back the same ACK to the computer

B, then also the computer B knows that something went wrong and starts over the retransmission of the window data that got wrong according to the ACK`s.

c) What should Computer B do next to fix the problem?

Ans.) Computer B can address this problem by retransmitting the data from which transmission window has gone wrong according to the last ACK received by computer B.

7) Imagine you are working for NMSU ICT's networking team and you notice that suddenly there are more TCP retransmissions than transmissions flowing through the campus net-work. What is happening and why is this bad?

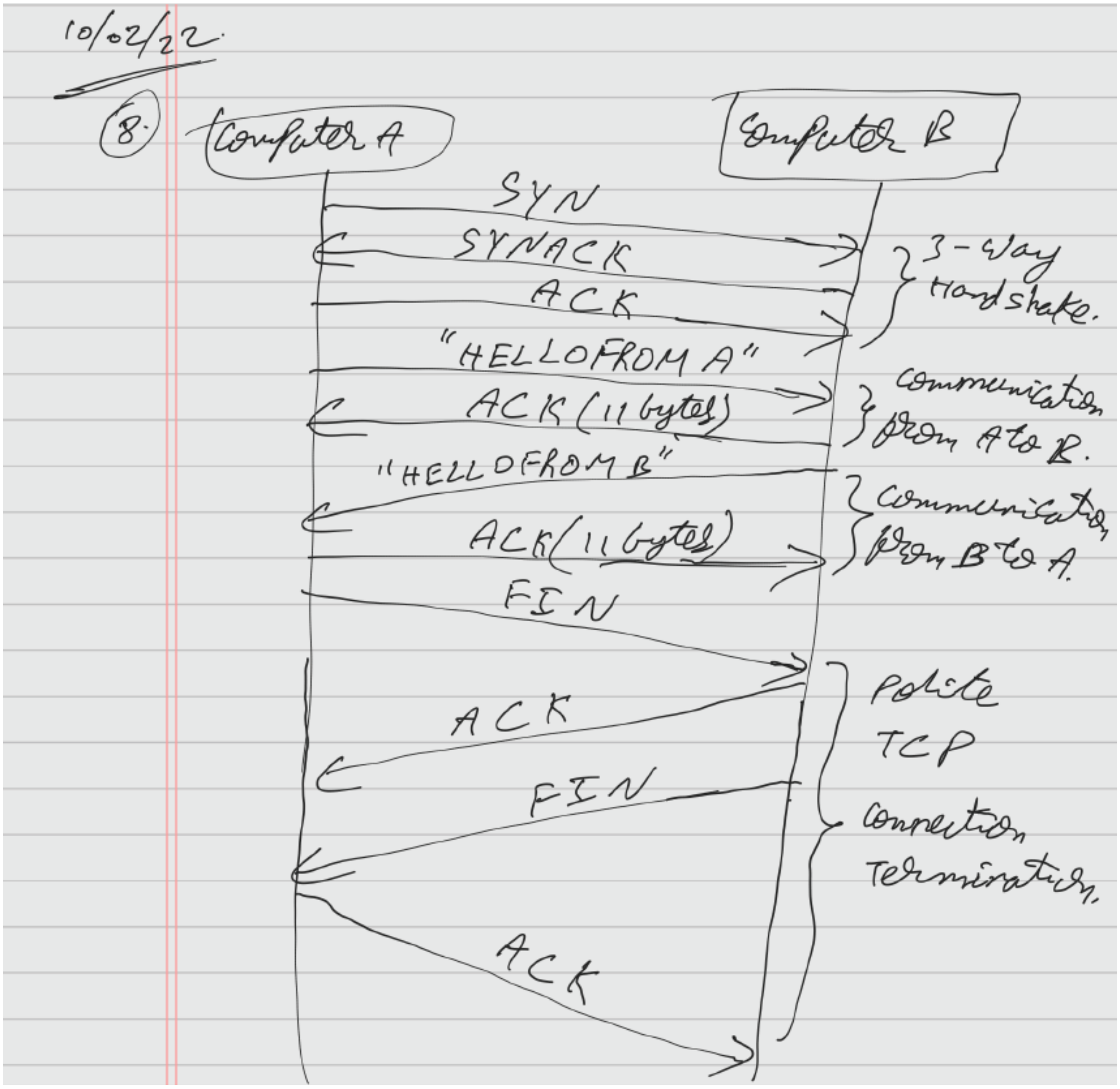
Ans.) When there`s a gap in sequence numbers in the capture, packet loss can lead to duplicate ACK`s, which leads to retransmissions.

Common reasons for retransmissions include network congestion, where packets are dropped. And another reason is Router QoS(Quality of Service) rules that gives preferential treatment to certain protocols.

A few retransmissions are ok, but excessive retransmissions are bad, because this shows up as slow application performance and/or packet loss to the user. This is because when the data is sent and there`s no Acknowledgement before the TCP`s wait time is expired, then the data is retransmitted. The retransmission rate of traffic from and to the internet shouldn`t exceed 2%, if the rate is higher than that then the user experience of the network may be affected.

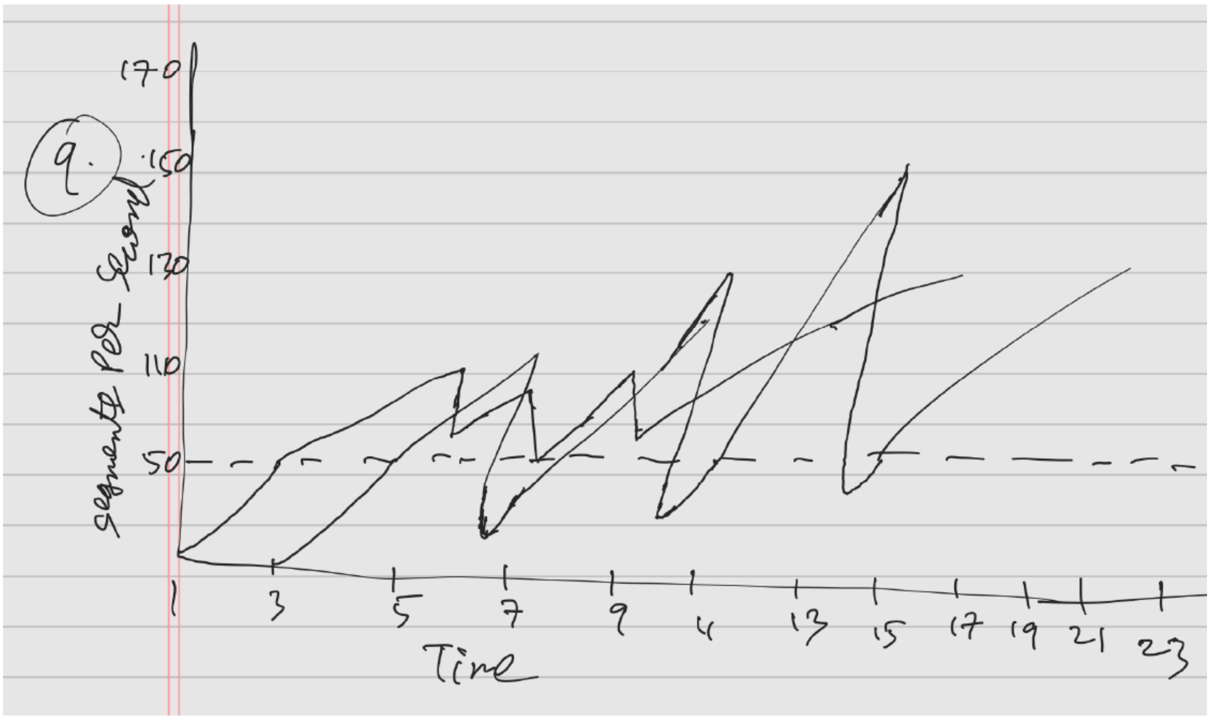
8) Draw a TCP packet diagram like Figure 2 where the following things happen:
    - A three-way handshake starts the TCP session
    - Computer A sends data and it is acknowledged
    - Computer B sends data and it is acknowledged
    - A polite TCP session termination

Ans.)



9) Create a diagram similar to Figure 1 where:
    - Two TCP conversations start within two seconds of each other and experience congestion every 2-6 seconds.
    - Your diagram must follow these two conversations for at least 30 seconds.
    - The first machine must use TCP Reno with a Slow Start threshold of 50 segments per second. The additive increase is 10 segments per second. The multiplicative decrease is a division by 2.
    - The second machine must also use TCP Reno with a Slow Start threshold of 50 segments per second. The additive increase is 20 segments per second. The multiplicative decrease is a division by 4.

Ans.)



10) Return to your diagram from the previous question. There is more to the difference between TCP implementations than just its flavor.

a) Will these two TCP conversations eventually converge on a fair sharing of the available bandwidth? Why or why not?

Ans.) As the first machine gets interrupted due to congestion with a multiplicative decrease and the second machine with an additive increase to the segment gets converged. This is due to the network congestion; the two TCP conversations converge on a fair sharing of the available bandwidth.

b) You can determine how many segments of data were sent in your diagram by integrating (as in calculus area under-the-curve) each of these lines. How many segments of data did each of your TCP conversations communicate? A rough geometric approximation is okay here.

Ans.) The first TCP conversation communicated roughly about 290 segments of data and the second TCP conversation communicated roughly about 390 segments of data.

c) Which conversation transmitted more segments of data? Why?

Ans.) The second TCP conversation transmitted more segments of data, because the second TCP conversation can carry more segments of data per second compared to the first TCP conversation.

d) Now do you believe that the tools you learned in calculus are actually useful for computer scientists.

Ans.) Yes, I guess so.