Sidharth Gilela
ID #: 1428033
sgilela@ucsc.edu
CMPE 150/150L

Lab 3

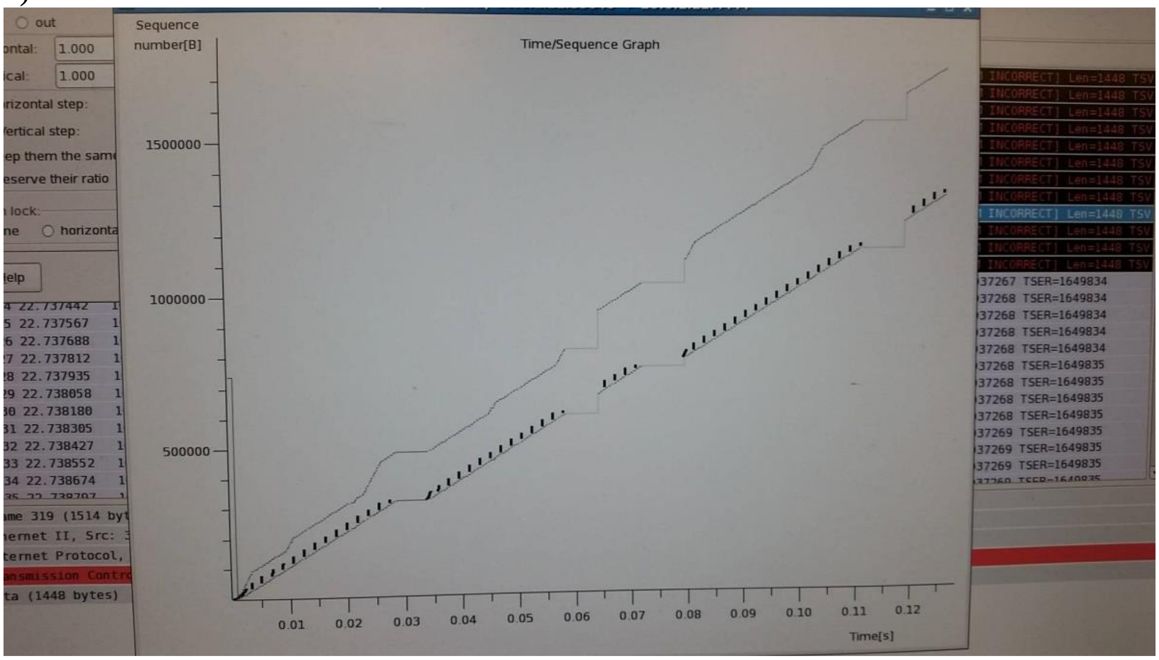
Theory Questions

1) Increasing the MTU on a lossless link will increase the throughput. This is because since increasing the MTU will increase the size of each packet, now we have a lesser amount of packets being sent each carrying a lot of data. However, if we were to decrease the MTU, we would have a lot of smaller packets where each of these packets has a header. All of these headers will give overhead to the throughput making it so that the throughout in reduced. Thus, with less packets that are huge (increasing the MTU), there is less overhead and we are maximizing the amount of information that could travel in a single RTT and hence increasing the throughput over a lossless link.
2) Decreasing the TCP Window size would decrease the throughput. Since there is less of a window size, less packets could be sent at once in an RTT and thus, the throughput in decreased.
3) The answer to this question could be interpreted from a definition of latency. Latency is defined as "the delay from input into a system to desired outcome" (techtarget.com). If I have an infinite size buffer and was receiving more data that I could process, then that would mean that the data that could not be processed will be dropped. Since this data from packets are dropped, our "desired outcome" of reaching the end host is delayed for a longer amount of time since a retransmission would have to be done. Thus, since the delay would increase, the latency will as well, since latency is defined as a form of delay.
4) The assumption that TCP makes about packet loss is that the packet was never reached to the host it was supposed to get to.
5) Here are the parts:
   A) Slow start is the process of increasing the window size by 2^n every time the packets have successfully been received by the host that they were supposed to get to. Once there is a packet drop though, this slow start will shift to congestion avoidance.
   B) Congestion avoidance is the process of increasing the window size by 1 every time the packets have been received successfully by the host they were supposed to get to. This is implemented after a packet lost from Slow Start and when there is packet loss from Slow Start, first the window size is cut in half and then the window size is incremented linearly as I have explained above.
   C) Fast Recovery/Fast Retransmit is a congestion control algorithm. This algorithm makes it possible to quickly recover lost data packets, avoiding a timeout. During a regular timeout period, no new or duplicate packets could be sent. However what this algorithm does, is that when the sender receives three duplicate acknowledgements, it will immediately resend the

lost segment without waiting for the timeout. This algorithm basically bypasses a regular timeout in order to improve efficiency (techtarget.com)

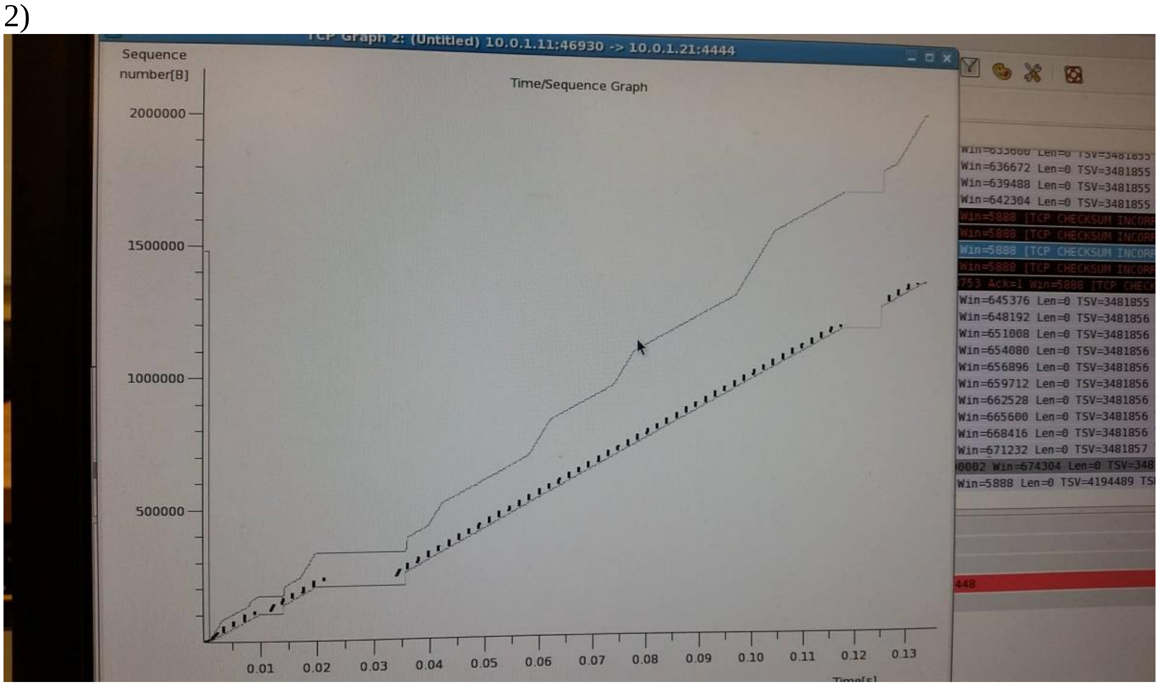
Lab Questions

1)



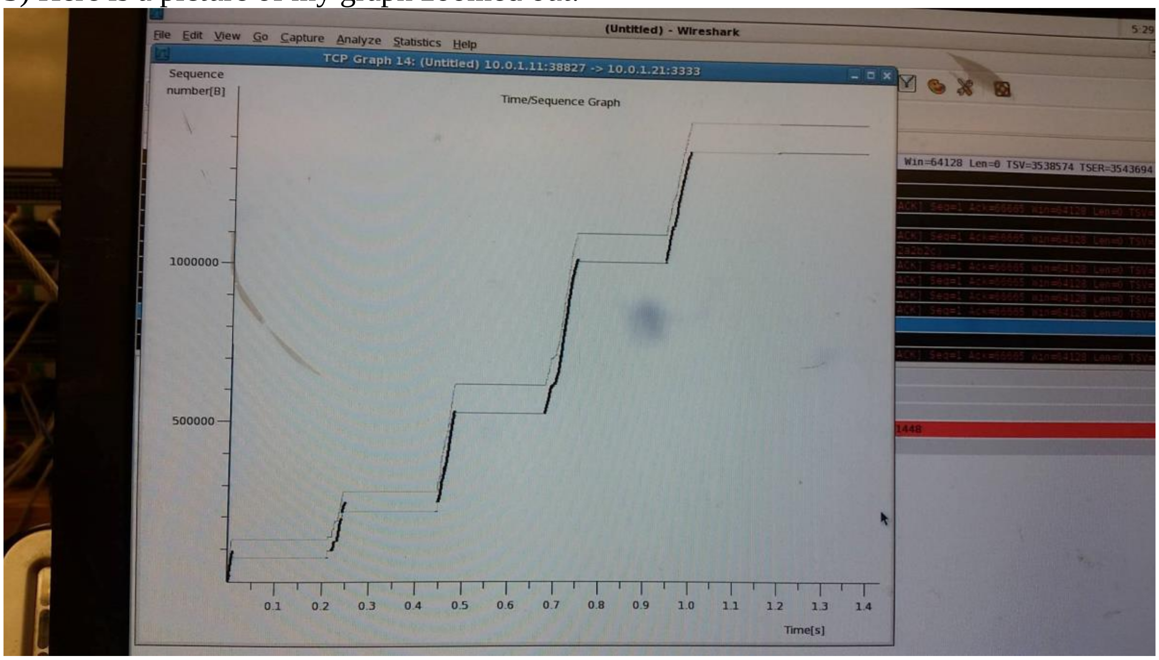
Here is a picture of the Tcptrace Time Sequence Graph

2)

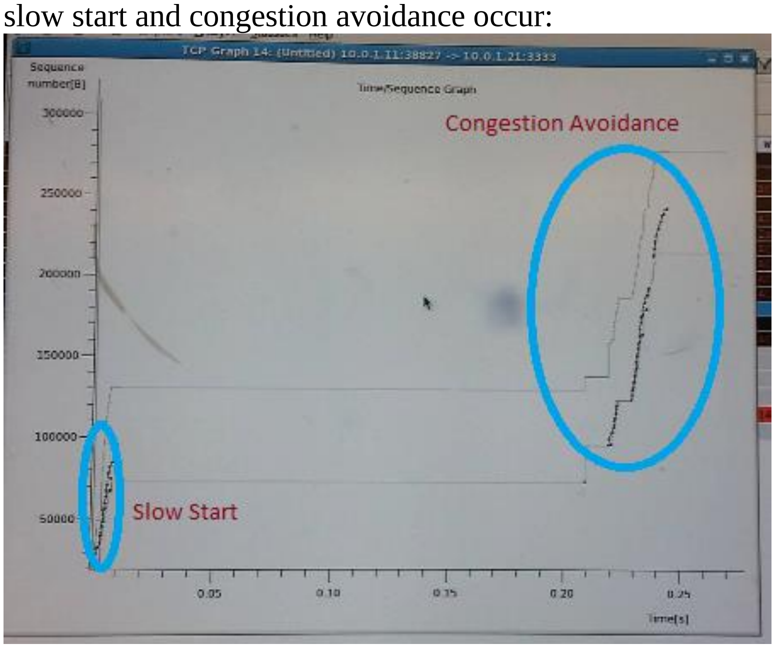


This is a picture of Tcptrace Sequence Graph after modifying those values. The main difference that can clearly be seen is the how the window size line looks. In our picture after making the changes the window size line seems to grow at more linear pace that that of the previous picture. This may be so because it appears that there are lesser packet drops as well causing the line to seem more linear. However, one of the things that I do not observe in both these graphs is the window size being cut in half after slow start is completed and congestion avoidance starts. After talking to the ta's, I have come to the conclusion that I probably cannot see this due to how old these PC's are and maybe some hardware faults. We also notice how when there are packet drops, and three more packets are sent afterwards, fast retransmit occurs because TCP recognizes that there is a lost packet.

3) Here is a picture of my graph zoomed out:



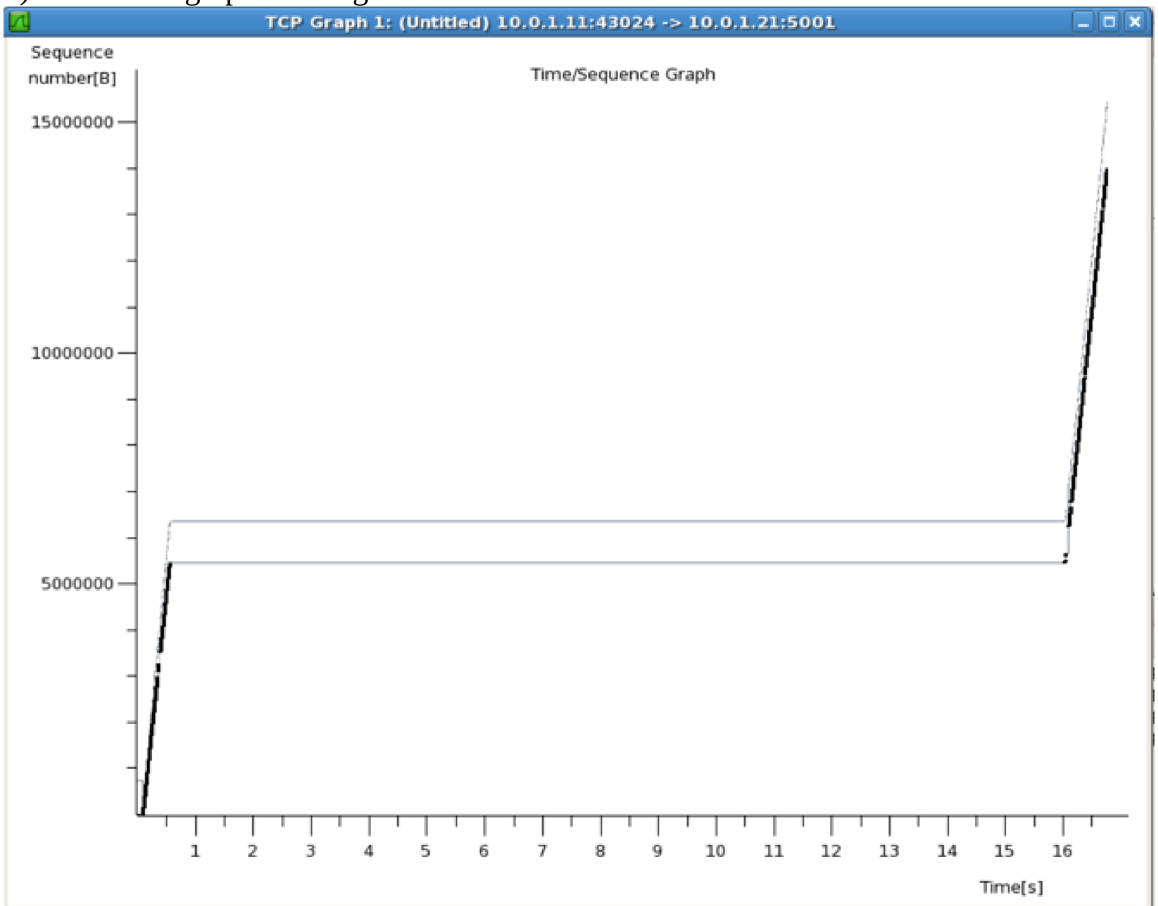
Here is a picture of my graph zoomed into the beginning section in order to see where slow start and congestion avoidance occur:



As I have showed above, the first circle indicates when slow start starts. In this mode we can see how the window size line which is the top line grows at an exponential rate.

However, after the first packet loss which can seen by the flat horizontal line, congestion avoidance starts, where the top line grows at a linear rate. This can be seen by the other circle. I have not shown this for the entire graph but for the rest of the increases in window size, they all grow at a linear rate. This is pretty easy to see because of how curved the top line, which is the window size line, grows.

4) Here is the graph I have gotten:



As it can be seen in the above picture, when the ttcp starts everything works fine and looks fine in the beginning and then a little while later when the link loss for the receiver is set to 100%, there is huge horizontal gap. This gap is there because the packets are not being received and thus the communication with the sender and receiver is that the sender keeps resending the same packet while the receiver is not being able to receive it. However, after a while when the link loss is set back to 0% the communication with the sender and receiver continues from where it left off as it can be seen by the linear line after the gap.

Sources Used:

1) http://whatis.techtarget.com/definition/latency
2) http://searchnetworking.techtarget.com/definition/fast-retransmit-and-recovery