Justin Wong
CMPE 150/L
February 21, 2016

Lab 3: TCP

1. What would increasing the MTU do for throughput on a lossless link?
Increasing the MTU could potentially break the ability to transfer through that link, if the amount of data sent through would be greater than the link can handle, this could also stop other smaller packets from getting through to the receiver.

2. How would decreasing the TCP Window size affect throughput?
With a smaller TCP window size less data would be able to be transferred in one message from sender to receiver, as such the throughput would be slower since there would need to be more messages sent to compensate for the smaller amount each time.

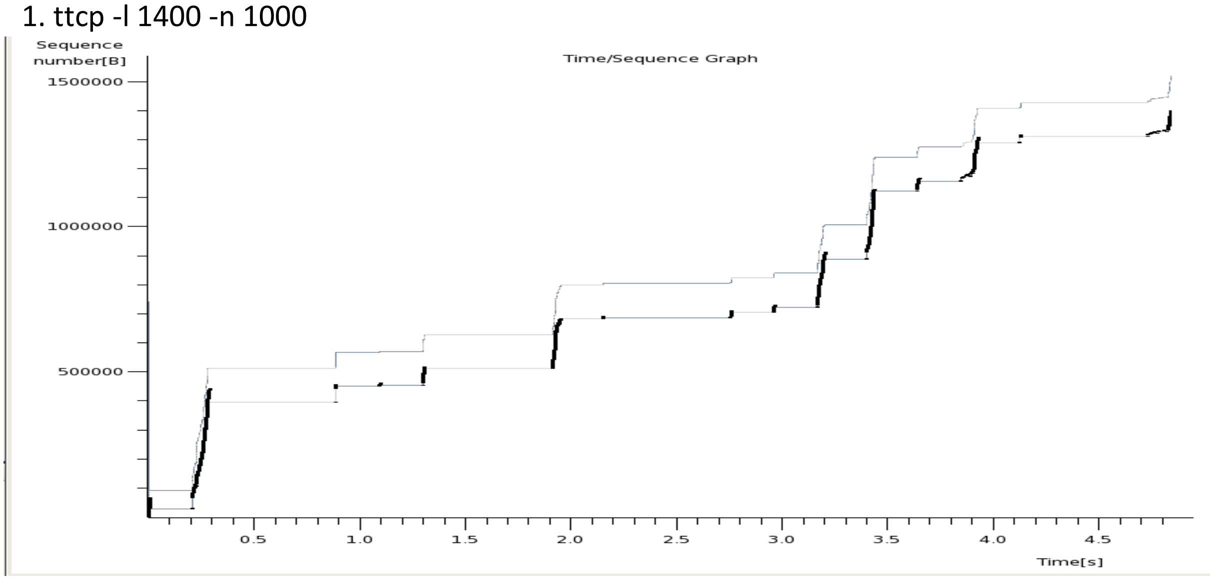
3. What would happen to your application latency if you had an infinite size buffer, and you were receiving more data than you could process?
The application latency would be very high, and could cause partial or complete failures from your network, network applications and network services.

4. What assumption does TCP make about packet loss?
If TCP sees packet loss then it will lower the amount of packets sent since it believes that there is simply congestion in the network.

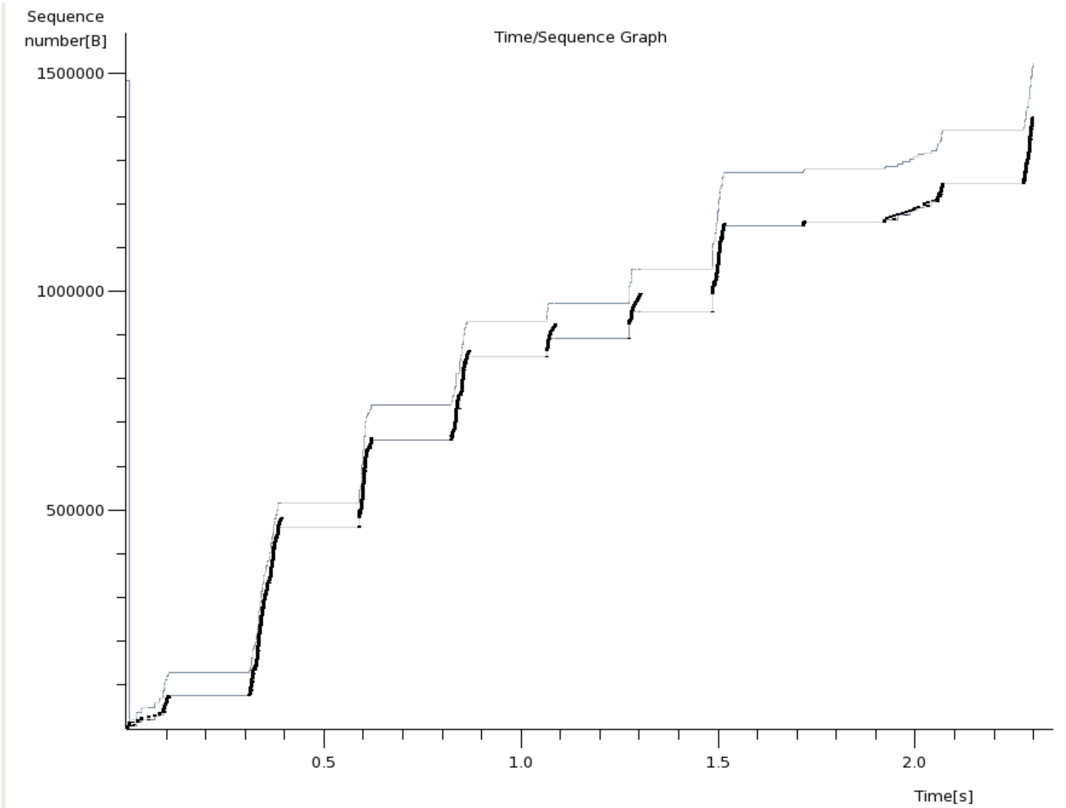
5. Briefly explain the following mechanisms of TCP:
5a. Slow Start: The window from the sender starts off small, it will wait for an acknowledgement from the receiver when the sender receives the ACK the window size will increase this will continue until the sender does not receive an ACK from the receiver at which point it will decrease the window size to accommodate for the perceived congestion.
5b. Congestion Avoidance: As long as non-duplicate ACKs are received then the congestion window (maximum segment size) is increased. When duplicate ACKs are received then the size of the congestion window will be halved and continue to increase again.
5c. Fast Recovery/Fast Retransmit: If there is a time out during a TCP then the sender will resend the segment that is presumed to be lost, in addition if the sender receives 3 duplicate ACKs for the same packet then it will resend that packet.

Lab

1. ttcp -l 1400 -n 1000



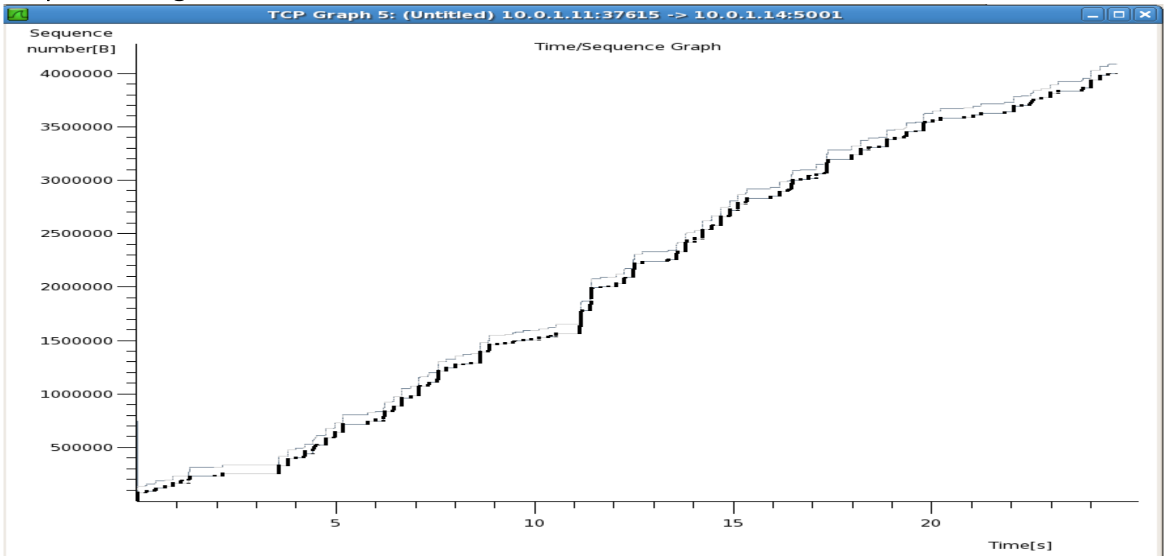
2. sysctl -w net.ipv4.tcp_rmem 10000 75000 12500000
sysctl -w net.ipv4.tcp_wmem 10000 75000 12500000



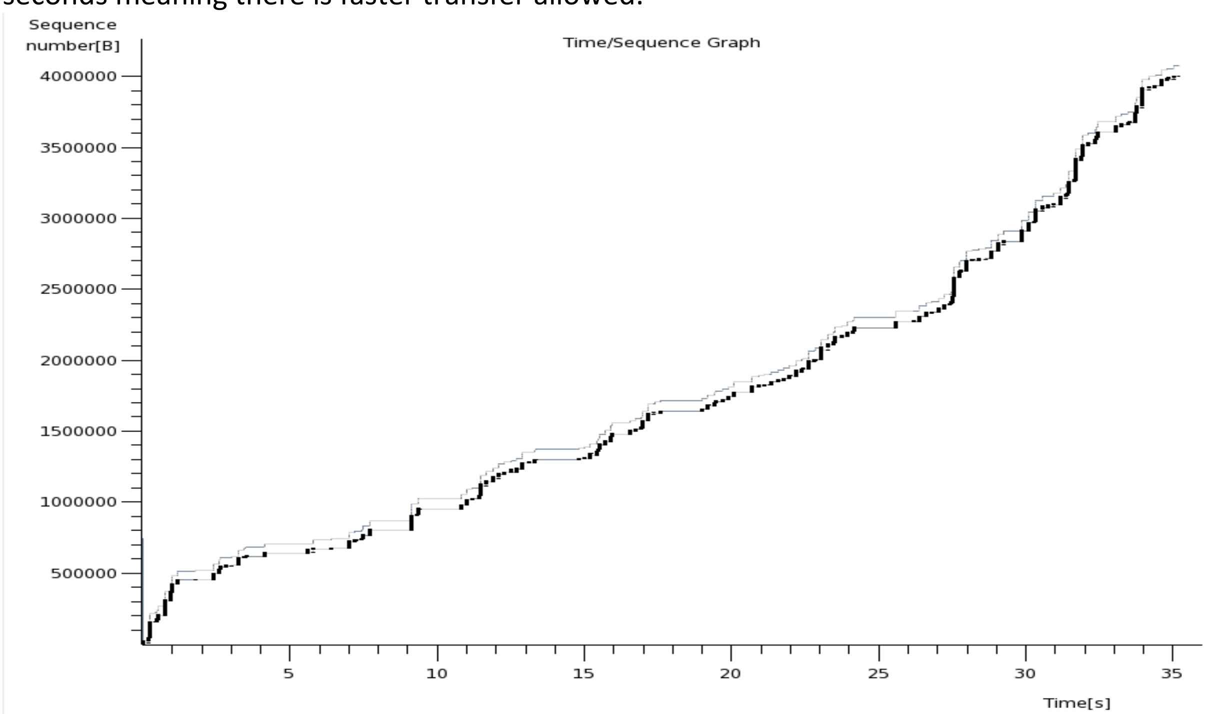
The main difference between #1's graph and #2's graph is that #2 is about 2 seconds shorter in time meaning that the connection was much faster to transfer packets. There is also more vertical lines in the graph of number 2 where packets are actually transferred, in graph 2 it is able to send more before having to lower the transmission rate.

3. Try to get 5-15% loss to show Congestion Control and Slow Start

tc qdisc change dev eth0 root netem loss 15%

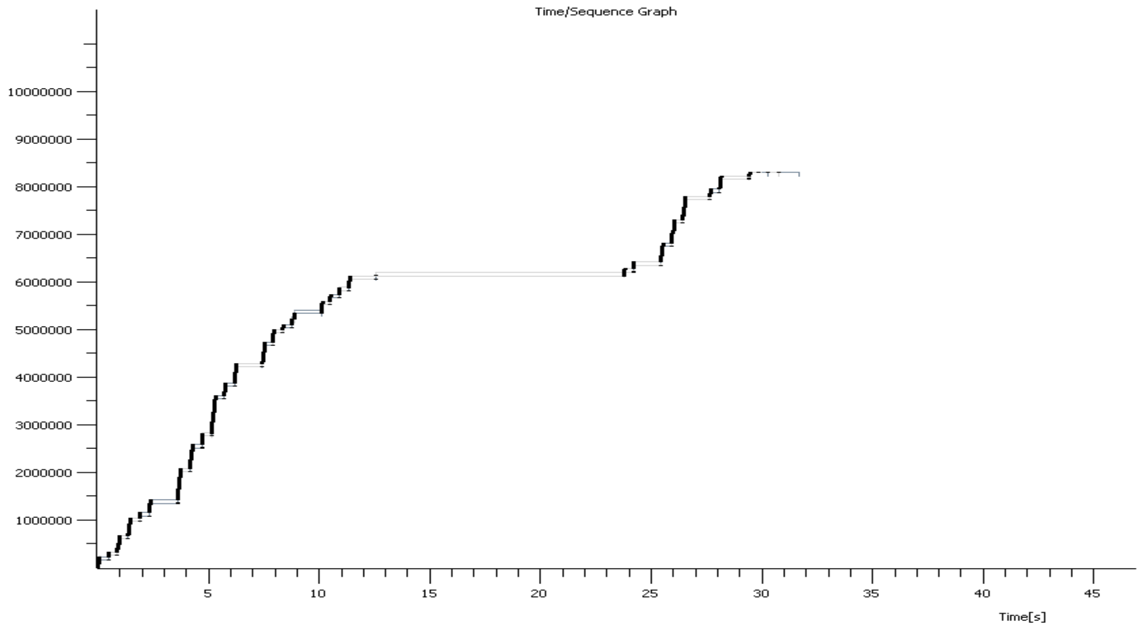


Slow start: the rate of transfer time is slow at the very beginning, and gradually builds up. It is clear that this is happening because the black lines are where packets are being transferred over, in the beginning up until around 2 there is some transfer, then at 3.5 it picks up until 11 seconds meaning there is faster transfer allowed.



Congestion Control: There are multiple times when the rate of transfer needs to be reset to prevent it from going too fast. As seen at roughly 11 seconds to 14 seconds there is transfer but then at 14-16 seconds there is no transfer going on to reset the sped and at 15-18 seconds it resumes. This happens a couple of times during this TCP connection.

4. 100% -> 0% (10 seconds) -> 100%
tc qdisc change dev eth0 root netem loss 0%
tc qdisc change dev eth0 root netem loss 10



0%

The TCP is a normal connection at first. Then the 100% packet loss starts at around 12 seconds and goes until about 23 seconds. Then it finishes.