LAB 3 – ELEC3500

Gavin Austin (3279166) & Ayden Khairis (3282229)

Introduction:

The goal of this lab was to simulate the basic TCP Congestion control techniques. This experiment provided the necessary data to study the effect of the AIMD Technique for a range of different traffic load conditions, as this is what has been identified as the process to throttle TCP Connection rates and therefore reduce congestion on a transmission link.

The work detailed below is based on a modified version of the standard TCP/IP Model, as to support the experiment. Each run of the simulation was set to 200 seconds, with values such as segment size and client count being changed for each run. These changes will be noted during the results section below.

Simulation Model:

Figure 1 shows the simulation model architecture used in the experiment. There were 8 simulation variables which were set to default values including client number, simulation time, advertised window, maximum transmission unit, maximum segment size, router queue size, client file size, and transmission link speed. In Part I of this simulation, the variables were unchanged. In Part II of this simulation, the maximum segment size value was changed from 1000B to 2000B. The amount of clients were changed from 1 - 3 throughout both parts of the simulation.

The Omnet IDE version needed to be downgraded in order for the results to come out correct. Even with this downgrade, the simulations would have an error but still produce the correct result and correct plots.

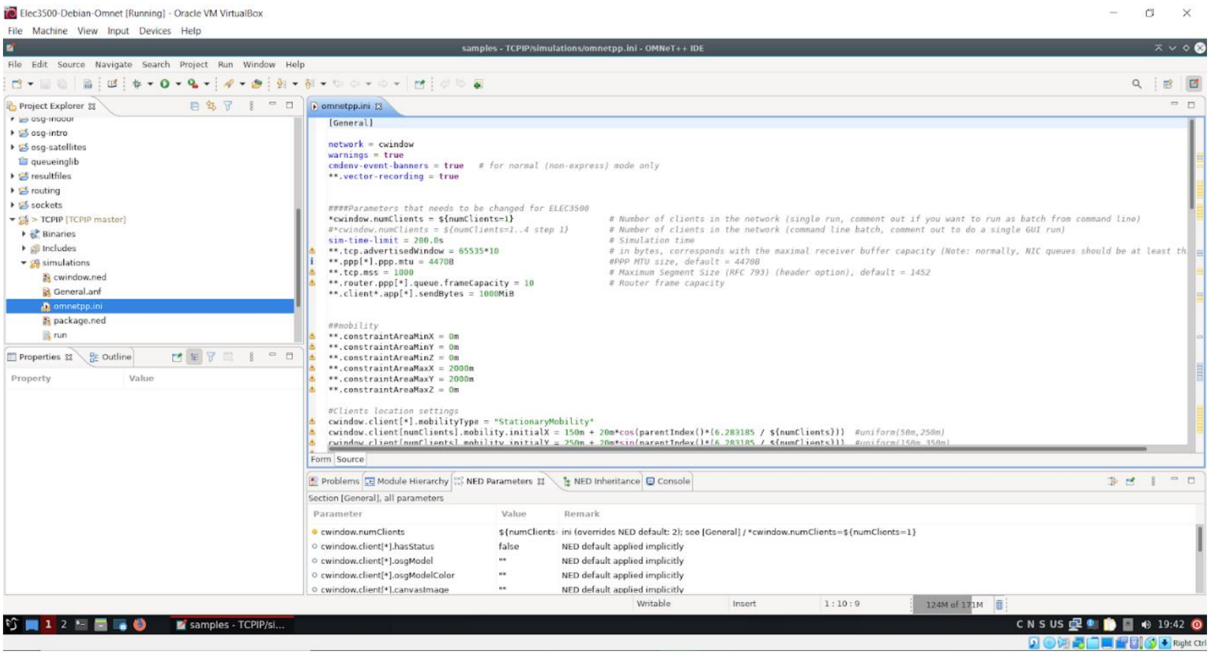


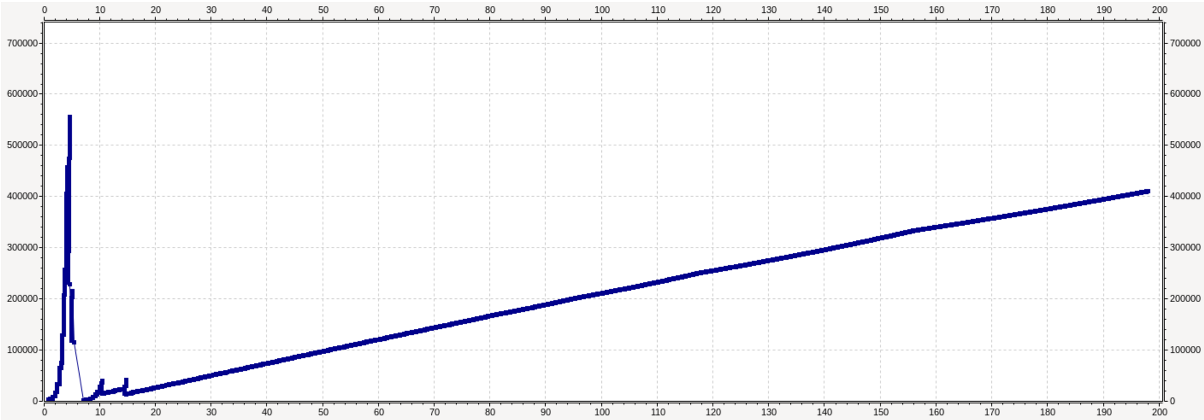
Figure 1

Results:

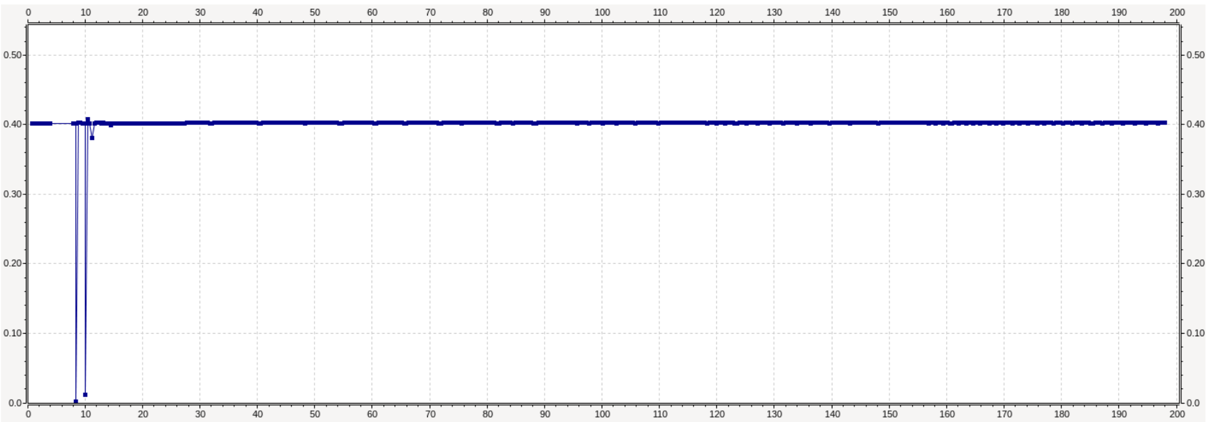
Part 1

Simulation #1:

* Sim time vs congestion window size for client #0

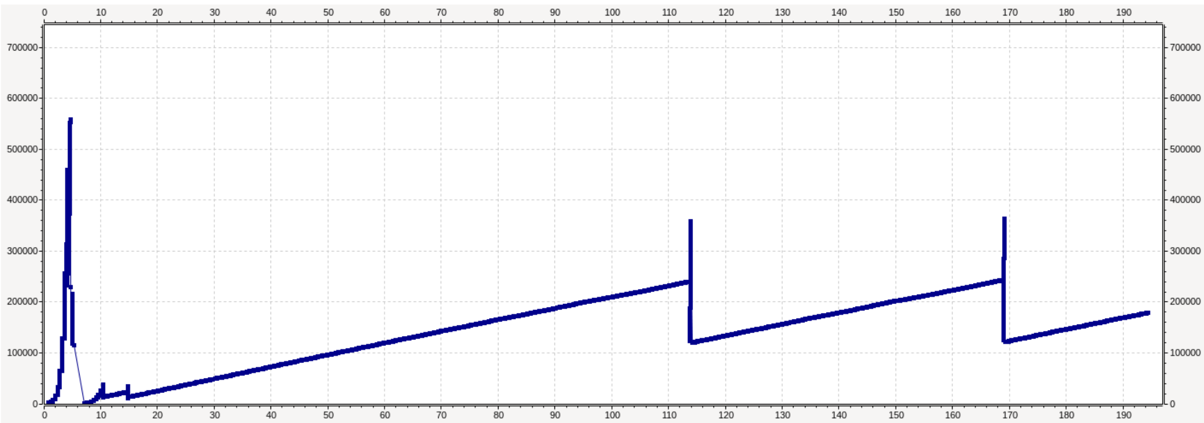


* sim time vs RTT (measured RTT) for client #0

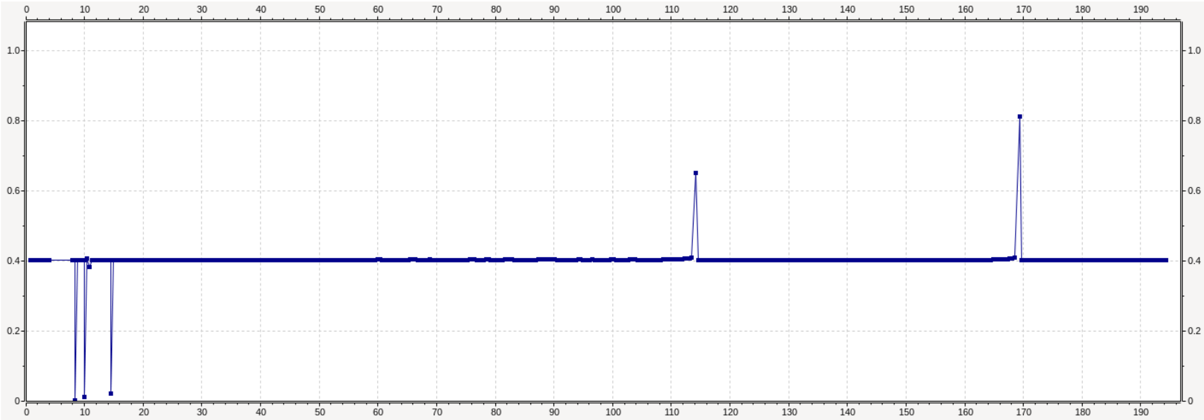


Simulation #2

* sim time vs congestion window size for client #0

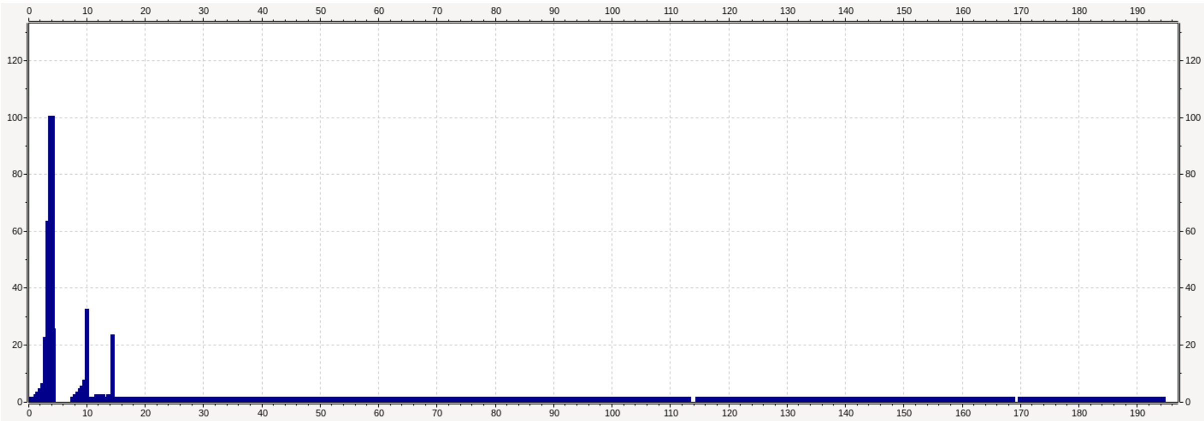


* sim time vs RTT (measured RTT) for client #0

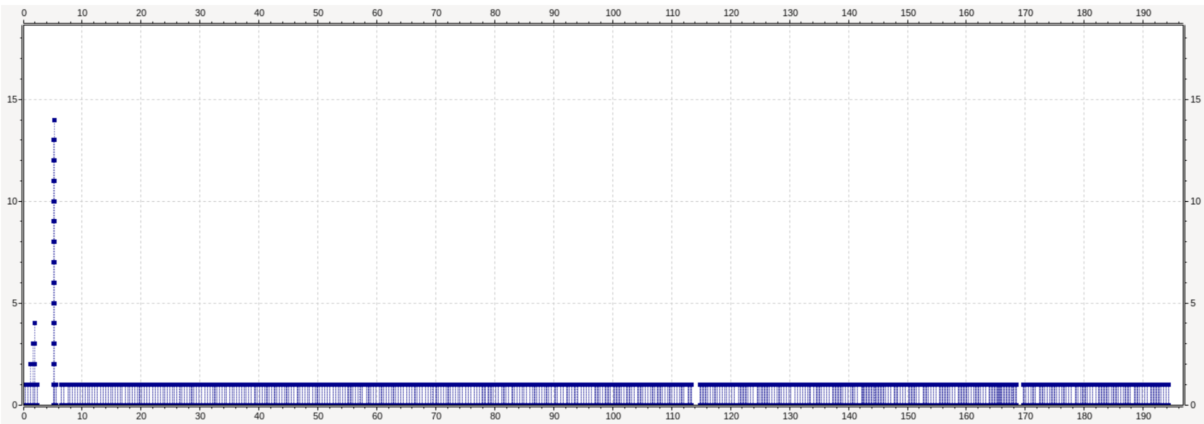


* sim time vs router queue lengths for clients #0 and #1

clients #0

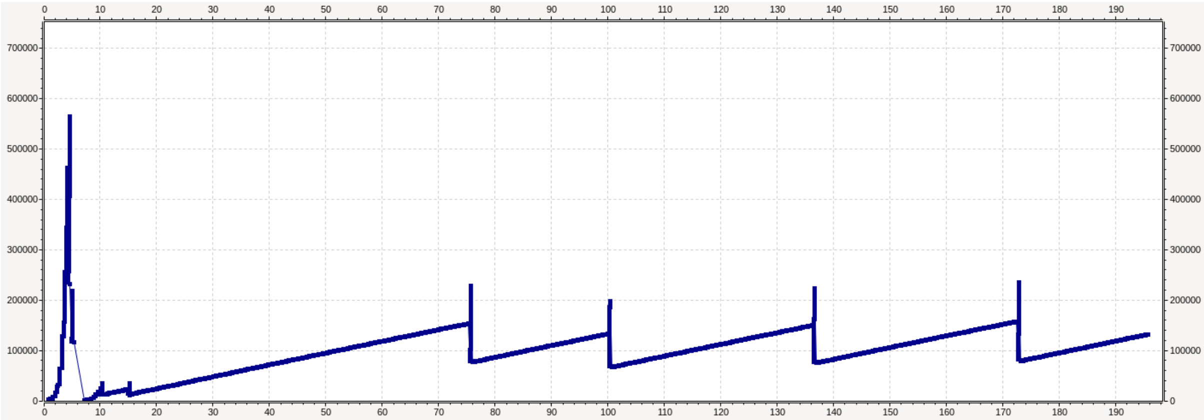


clients #1

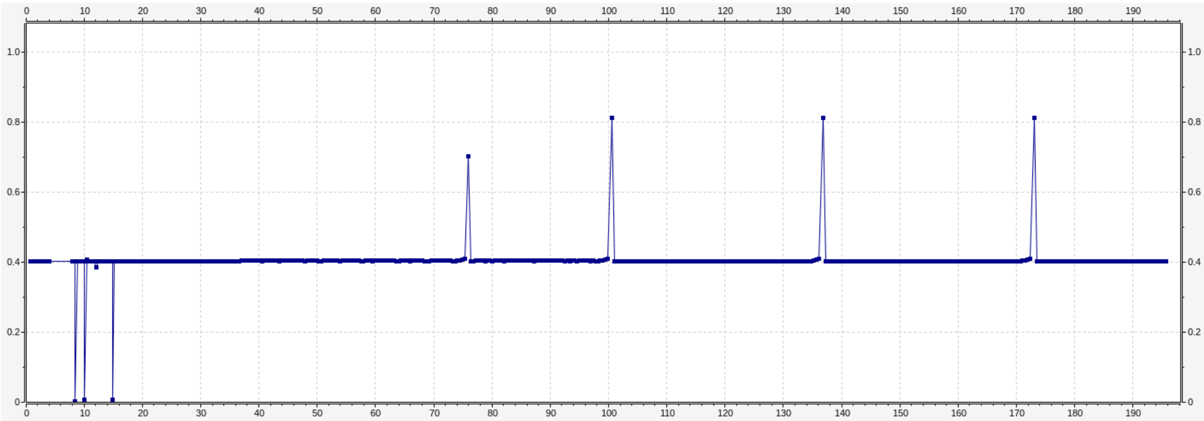


Simulation #3

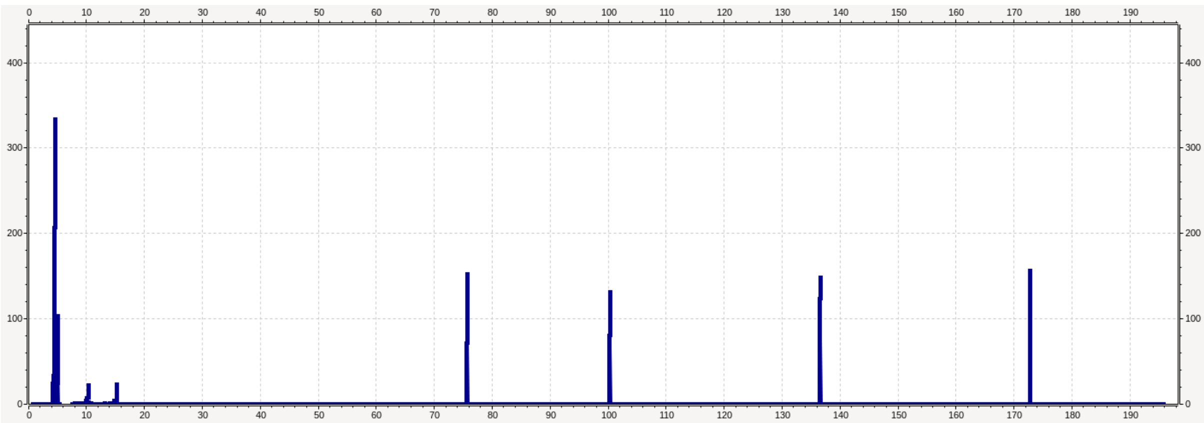
* sim time vs congestion window size for client #0



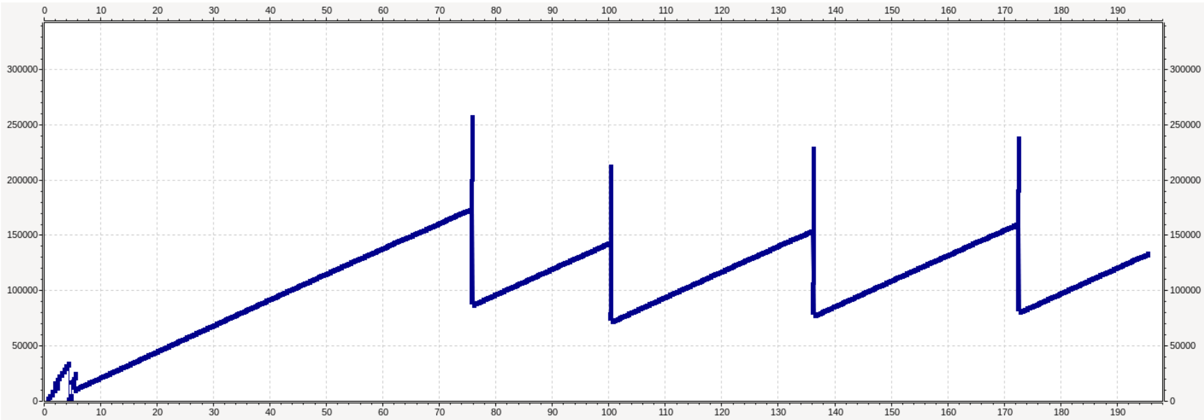
* sim time vs RTT (measured RTT) for client #0



* sim time vs number of duplicated ACK received by client #0



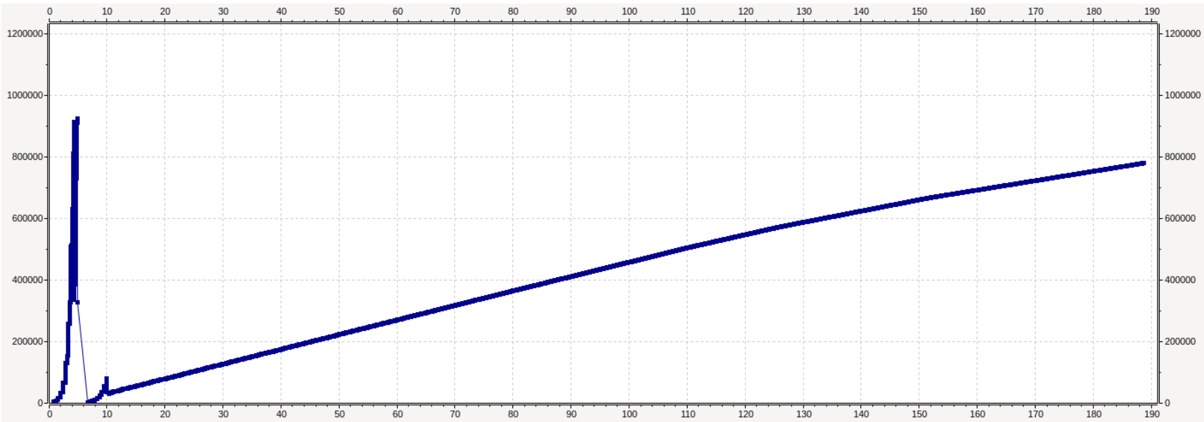
* sim time vs congestion window size for client #2



Part 2

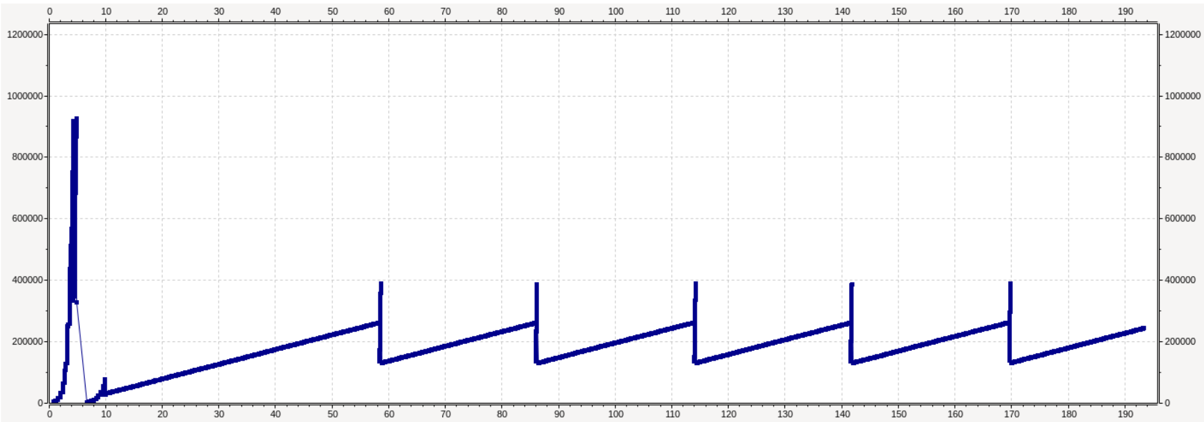
Simulation 1: (Clients #0)

Simulation time vs the congestion window size for Client #0

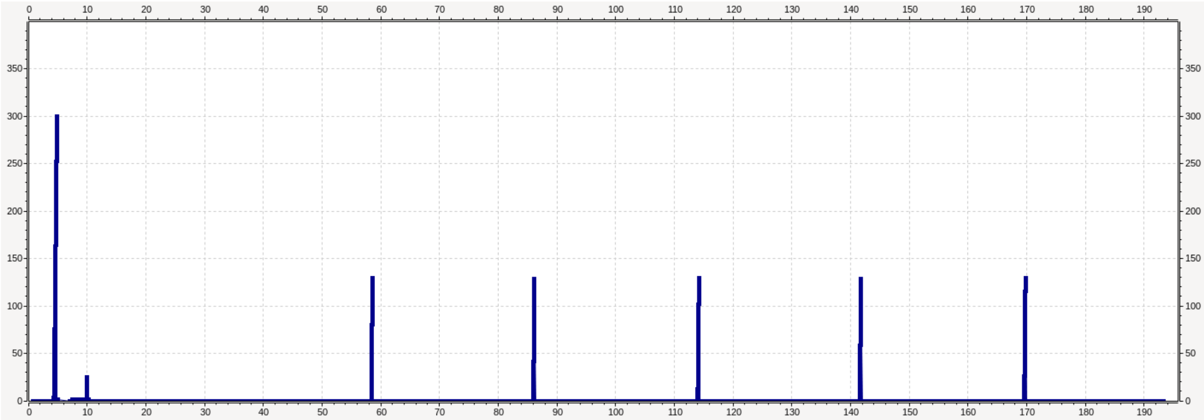


Simulation 2: (Clients #1)

Simulation time vs the congestion window size for Client #0

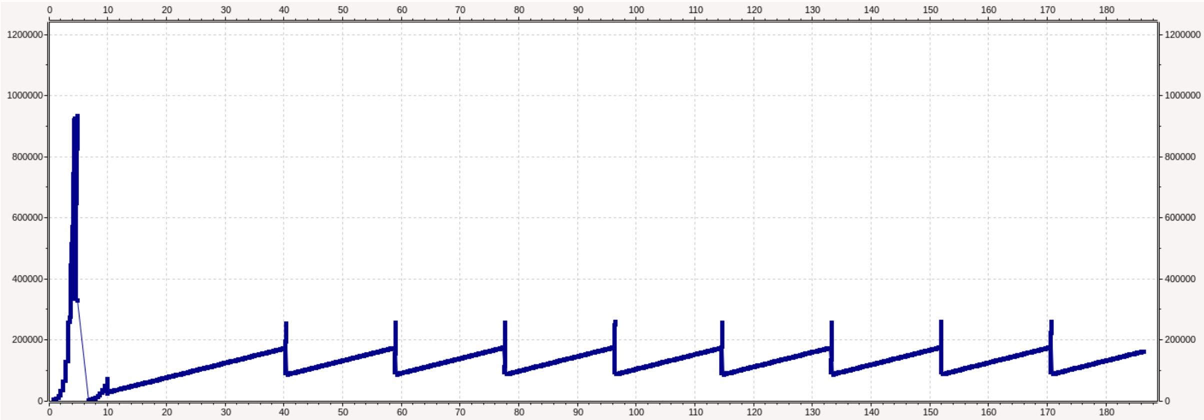


Simulation time vs number of duplicated ACKs for Client #0

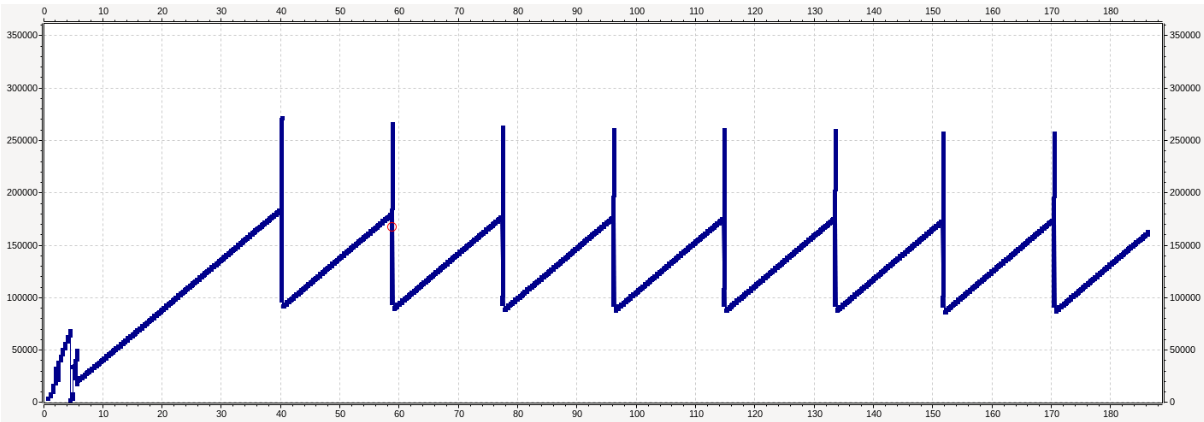


Simulation 3: (Clients #2)

Simulation time vs the congestion window size for Client #0



Simulation time vs congestion window size for Client #2



Analysis

In the results above, we can see a series of changes/effects on the window size due to the modification of traffic load.

In the simulation we can see Traffic load as a function of time and data, this means that for traffic load to increase, either the time to send data is reduced, or in this case, the amount of data being sent is increased. The congestion window size mirrors this increase, up until the point of packet loss, where it is reset to half of the value that was achieved before loss. From here, both the congestion window size & traffic load can be seen to modulate between being reset and being in a state of increase. This is mirrored equally with Round Trip Time, as the traffic load is built up, the time to be processed & returned adds up. As the traffic load is reset by the congestion window, RTT will decrease.

Question 1:

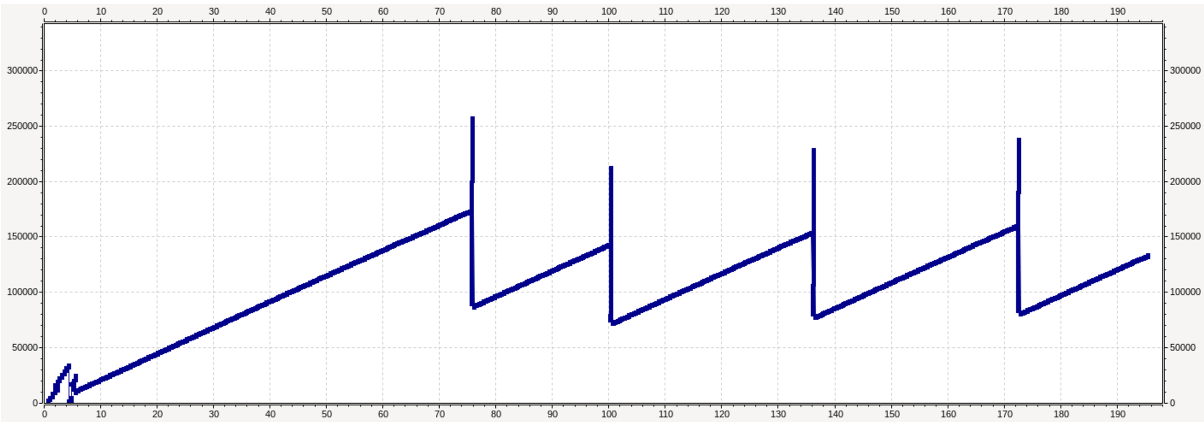
The congestion window size varies differently for different load conditions because the protocol needs to adjust the window size based on the load conditions. When the load conditions increase, the congestion window size also increases, and more timeouts occur. This is due to the congestion window resetting at 1 MSS during a timeout, MSS being the maximum segment size. Again, the higher the load, the higher the congestion window.

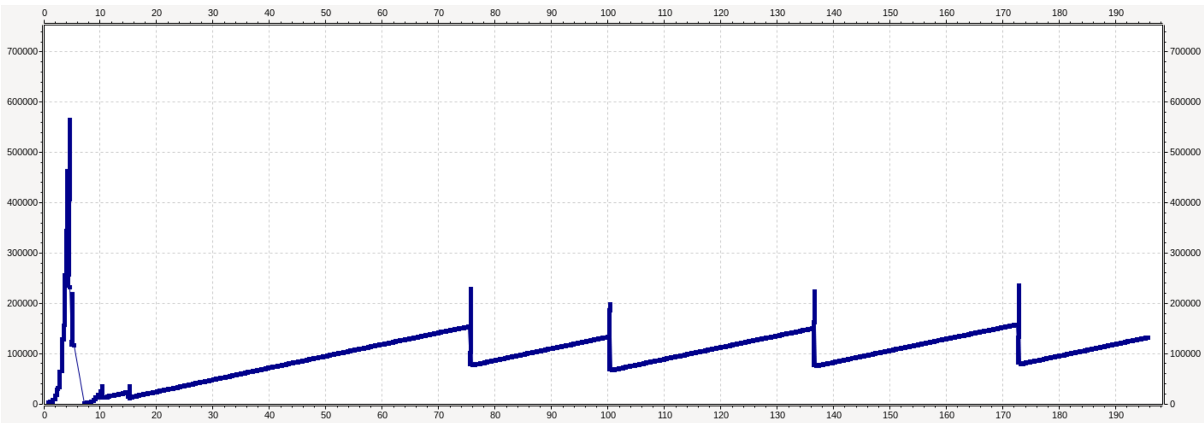
Question 2:

Yes, there is a connection between the two events. We can see that when the congestion window size is reset to half its value, we see a peak in the DupACK Plot. As a duplicate acknowledgement packet indicates a lost packet in the stream, we can see that these are being produced when the packets are dropped, causing there to be a resize in the congestion window.

Question 3:

The average throughput of Client #0 and Client #2 are relatively similar. When the graphs below are compared, we can see that each have the same number of dips and a similar max congestion window size. However, Client #0 has a section of very large throughput. When testing the initial packet loss, this skews the average, meaning Client #0 has an overall higher average throughput.





Question 4:

The change from 1000B to 2000B in the simulation affected client 0's throughput mostly at the beginning of the simulation. Looking at simulation 1 in both parts of the lab, you can see there is a 40000 difference at the beginning of the simulation where the highest traffic loads are. The lowest traffic loads (besides the beginning of the simulation) occur at the same time despite the segment size differing.

Question 5:

When viewing the graphs above for DupACK vs Simulation Time, we can see a reduction in dropped packets in the system when MSS is lower. As MSS increases, it takes less packets to reach a traffic load that triggers a cwnd reset/halving. We see less duplicate ACK's being received by the client when less of these events occur.