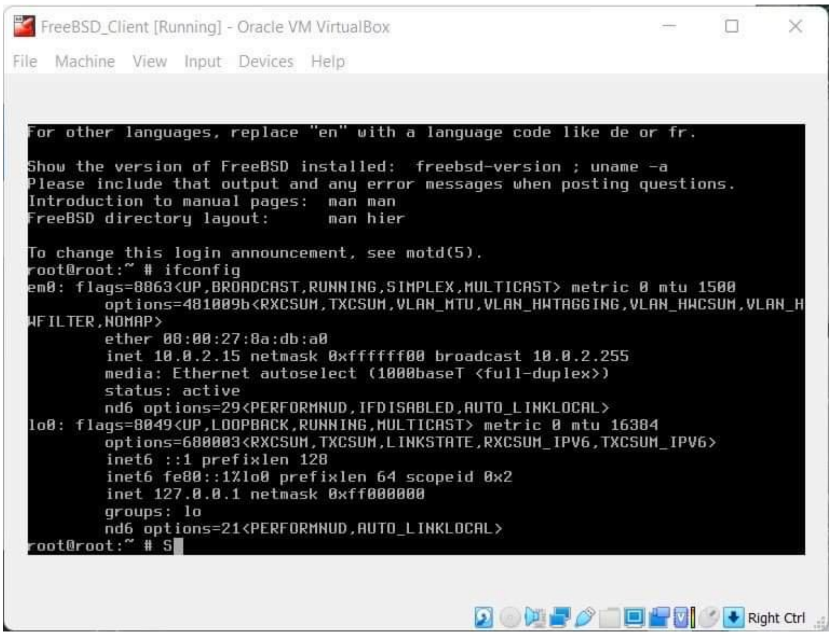
Introduction to Computer Networks - Programming Assignment 2

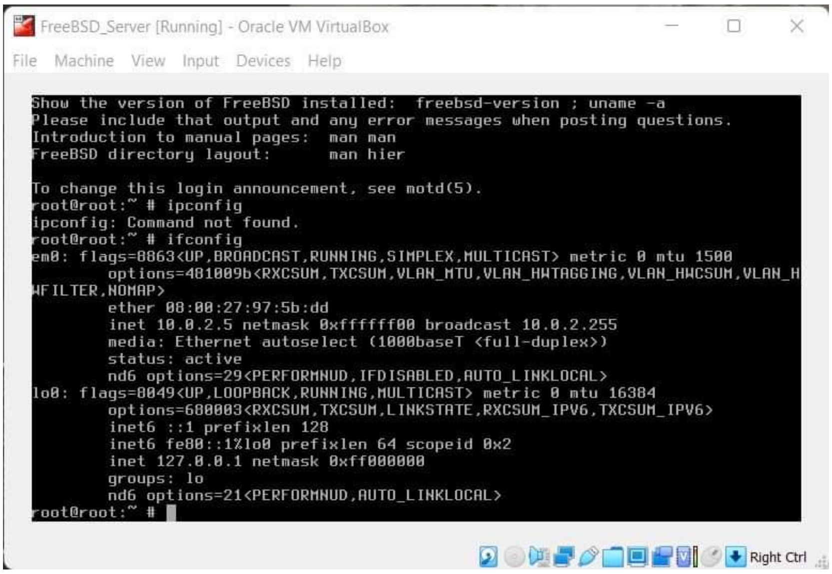
Anastasia Riana 邱慧莉 - 109006234

Part 1. Environment Setup

We can see from these figures that the FreeBSD client and server has been successfully installed. By using the **ifconfig** command, we are able to check both the server and client's IP address, which is 10.0.2.15 and 10.0.2.5 for client and server respectively.

Result:





Part 2. TCP Data Transmission

The command used to run the **iperf3** command is:

iperf3 {a} {b} {c}

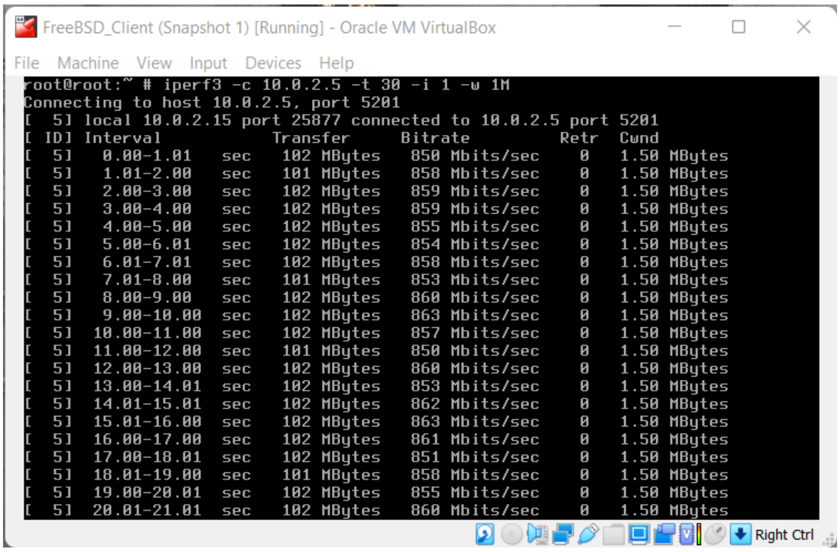
iperf3 {-c 10.0.2.5} {-t 30} {-i 1} -w 1M

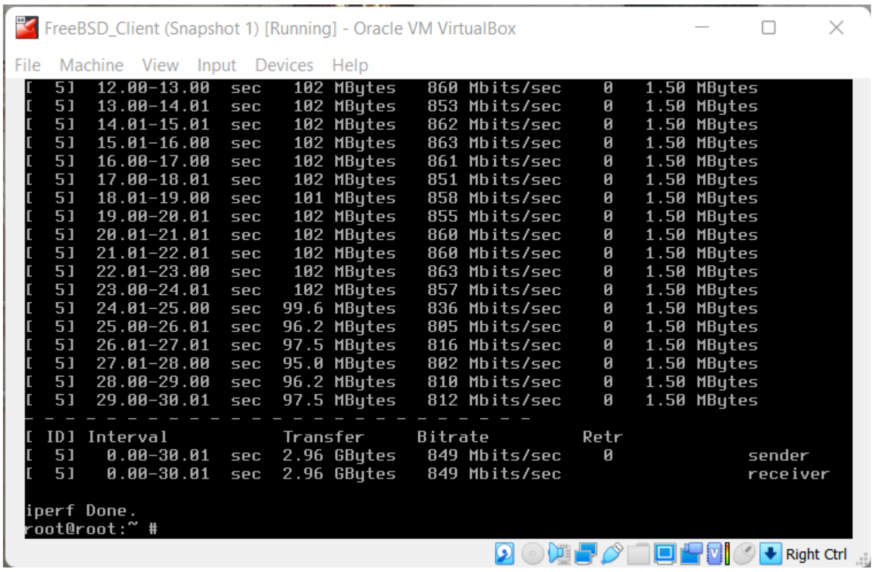
{-c 10.0.2.5} -c 10.0.2.5: indicates that run iperf3 to be run in client mode and connecting to an iperf3 server that is running on host

{-t 30}: indicates the time to transmit the data in seconds.

{-i 1} : set the interval time in seconds between the periodic bandwidth, jitter and loss report as requested by the homework question.

Result:





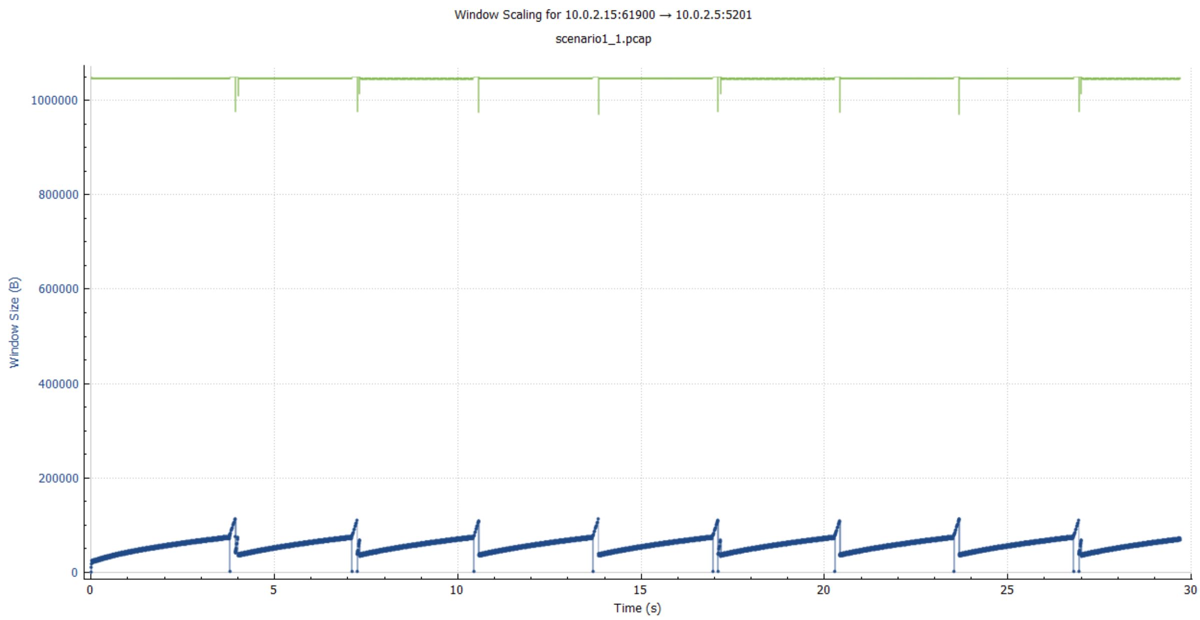
Part 3. TCP Congestion Control

Scenarios

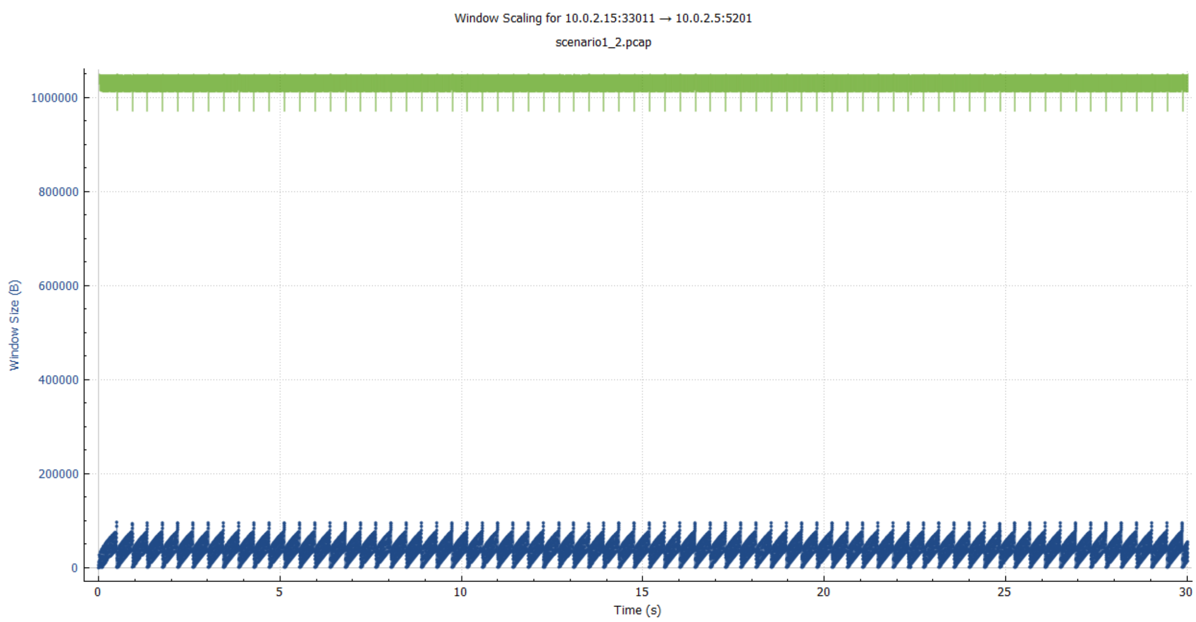
	Bandwidth	Delay	Packet Loss Rate	algorithm
Scenario 1	4 and 32 Mbit/s	default	default	newreno
Scenario 2	8 Mbit/s	10 and 100 ms	default	newreno
Scenario 3	8 Mbit/s	default	0.1 and 1%	newreno
Scenario 4	8 Mbit/s	100 ms	default	newreno and cubic
Scenario 5	8 Mbit/s	default	1%	newreno and cubic

Note: violet - slow start ; blue - congestion avoidance ; maroon - fast recovery

Scenario 1_1: 4Mbit/s Bandwidth



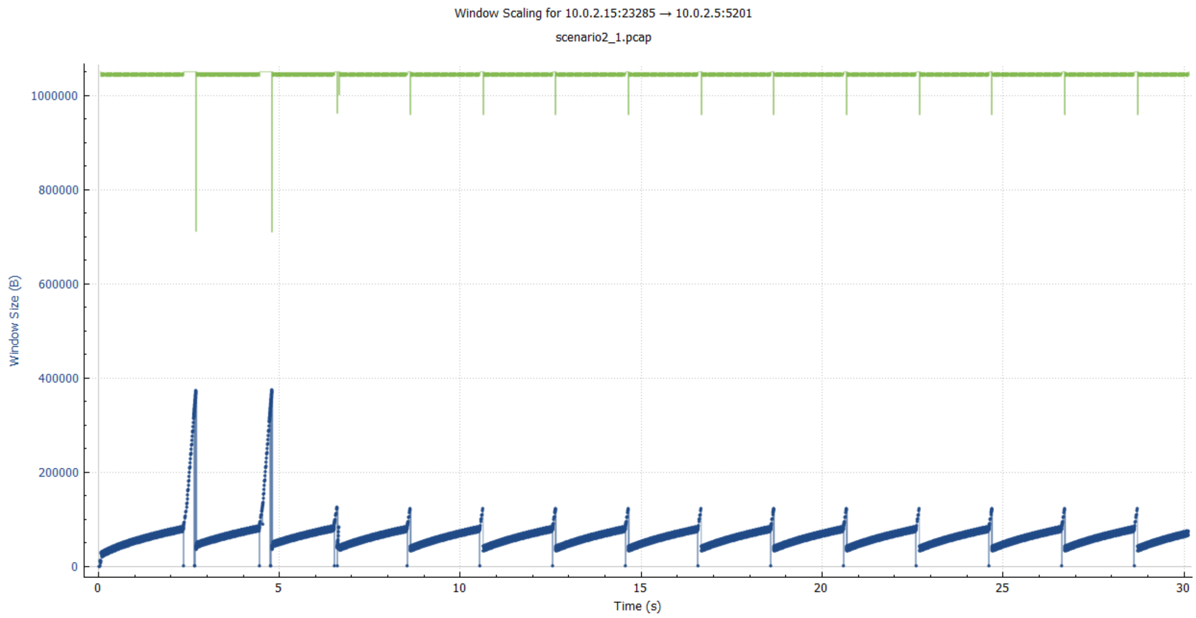
Scenario 1_2: 32Mbit/s Bandwidth



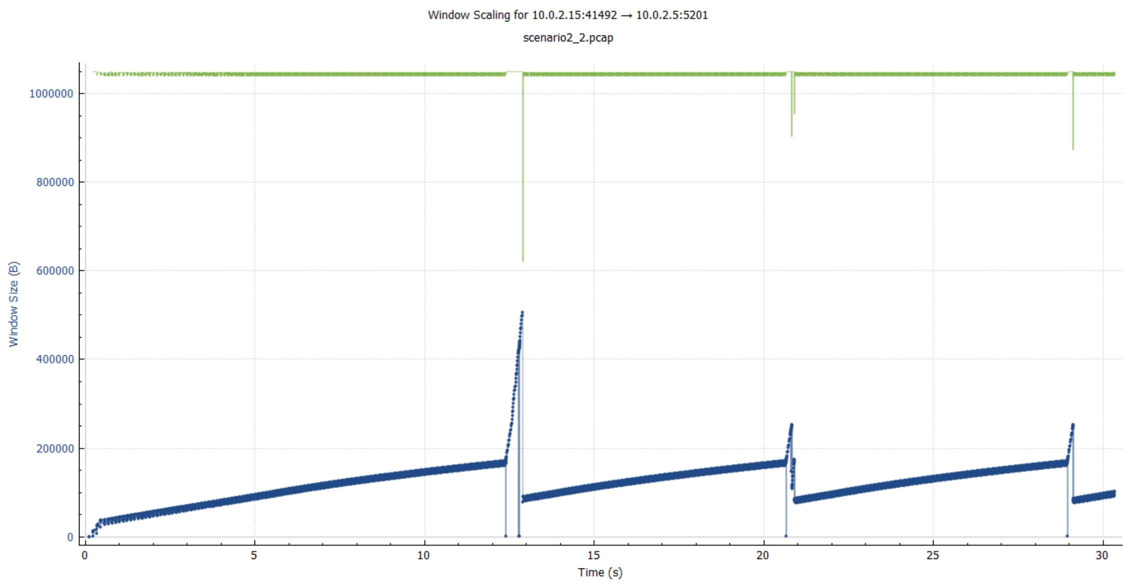
Analysis:

From the first scenario, we can see one slow start in the beginning. We see multiple congestion avoidance, indicated by the rising spike. Lastly, we see multiple fast recovery, indicated by the break of the spike drop. From the second scenario, we can see one slow start in the beginning. We see multiple congestion avoidance, indicated by the rising spike. Lastly, we don't really see any fast recovery. Then, we can conclude that with higher bandwidth, packets are sent more efficiently and can be sent faster from server to client and vice versa.

Scenario 2_1: 8Mbit/s Bandwidth, 10 msec delay



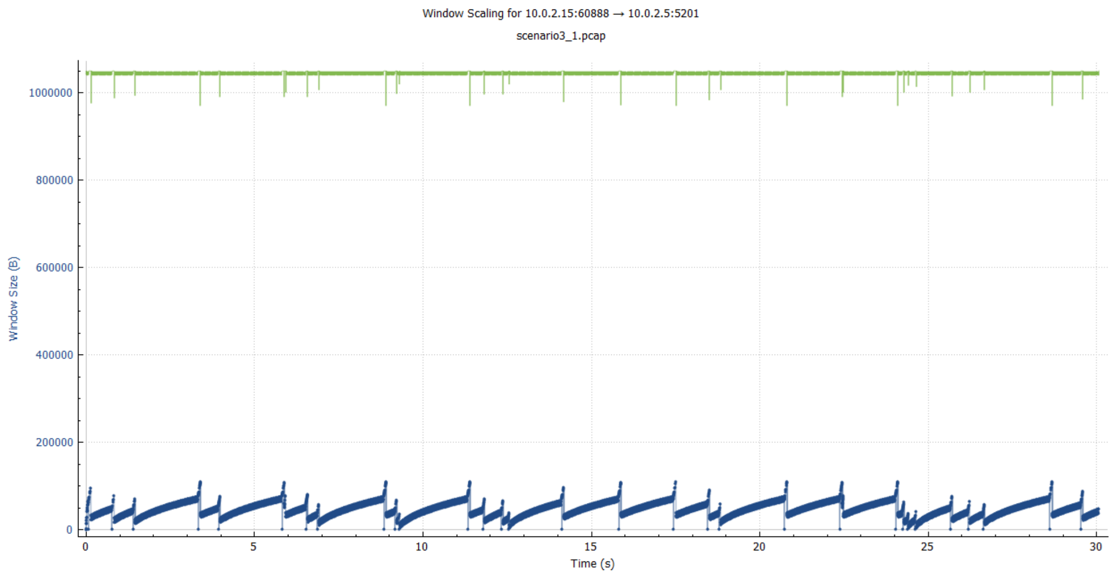
Scenario 2_2: 8Mbit/s Bandwidth, 100 msec delay



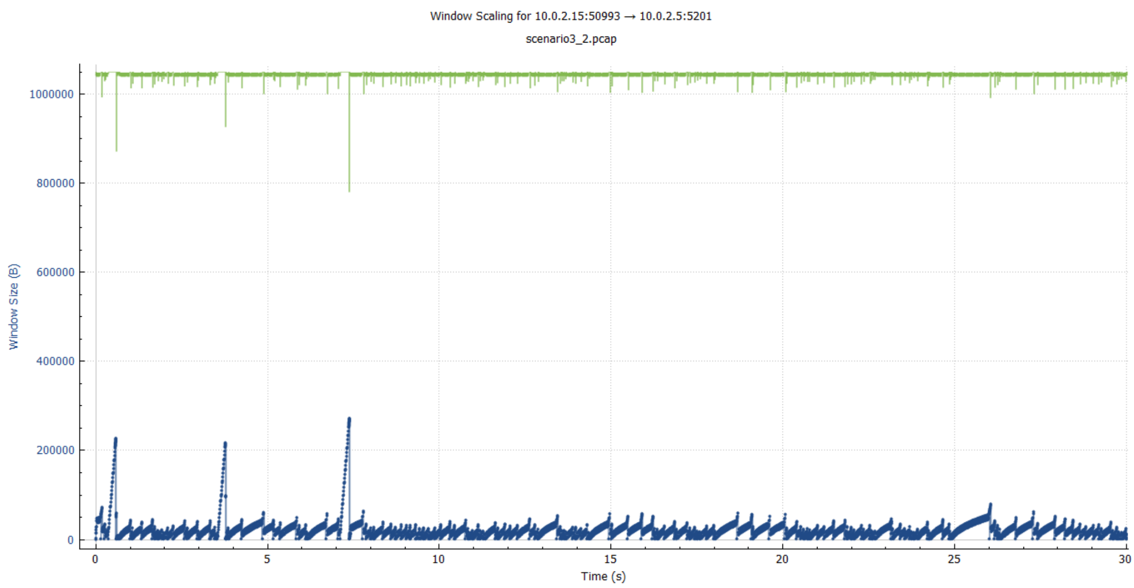
Analysis:

From the first scenario, we can see one slow start in the beginning. We see multiple congestion avoidance, indicated by the rising spike. Lastly, we see multiple fast recovery, indicated by the break of the spike drop. From the second scenario, we can see one slow start in the beginning. We see multiple congestion avoidance, indicated by the rising spike, however it is less than the first scenario. Lastly, we also see many fast recoveries. Then, we can conclude that the greater the delay, the sender must spend more time being idle which reduces how fast throughput grows. Hence, the smaller delay is preferred, which is the first scenario.

Scenario 3_1: 8Mbit/s Bandwidth, 0.1% Packet Loss Rate



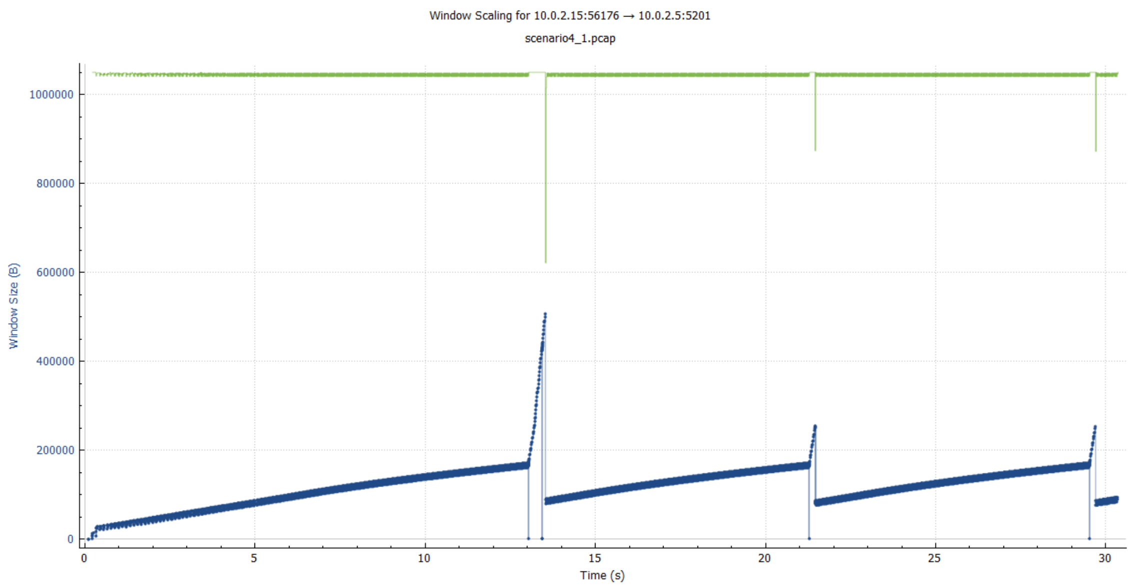
Scenario 3_2: 8Mbit/s Bandwidth, 1% Packet Loss Rate



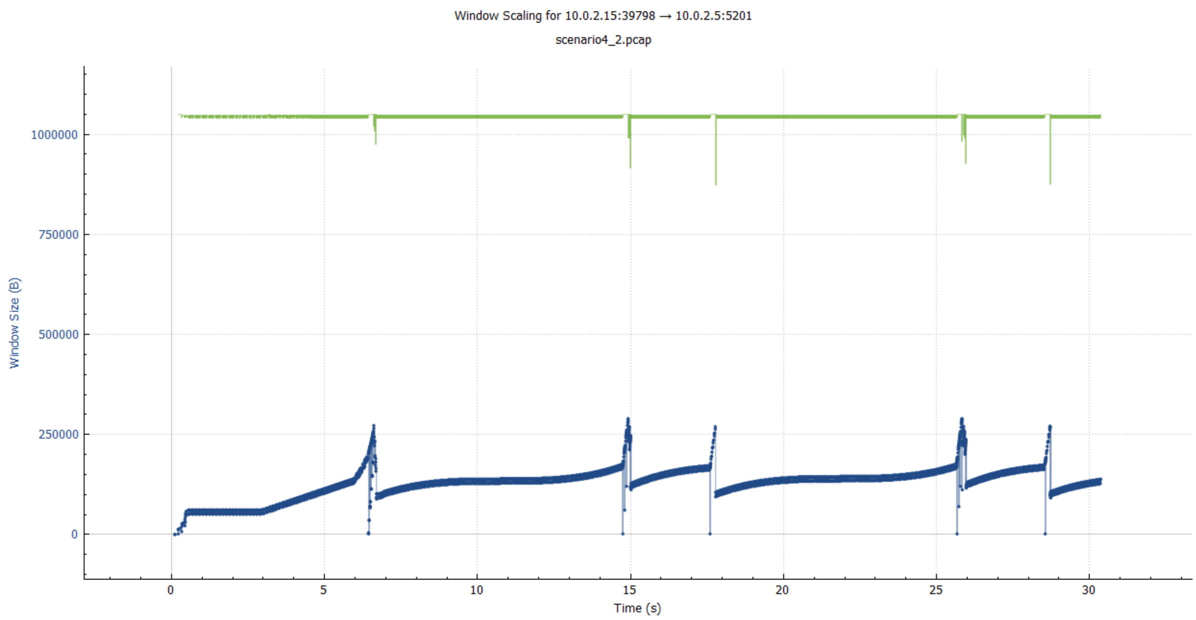
Analysis:

From the first scenario, we can see one slow start in the beginning. We see multiple congestion avoidance, indicated by the rising spike. Lastly, we see multiple fast recovery, indicated by the break of the spike drop. From the second scenario, we can see one slow start in the beginning. We see multiple congestion avoidance, indicated by the rising spike, compared to the first scenario. Lastly, we also see a lot of fast recoveries, compared to the first scenario. Then, we can conclude that the lower the packet loss rate, the better.

Scenario 4_1: 8Mbit/s Bandwidth, Delay 100 msec, Algorithm newreno



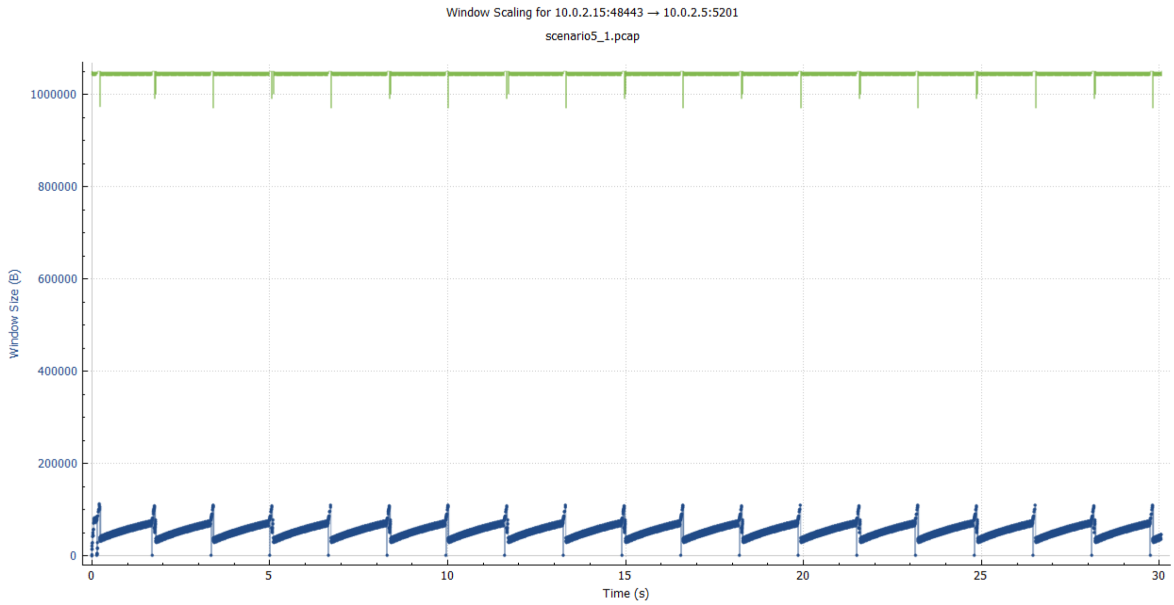
Scenario 4_2: 8Mbit/s Bandwidth, Delay 100 msec, Algorithm cubic



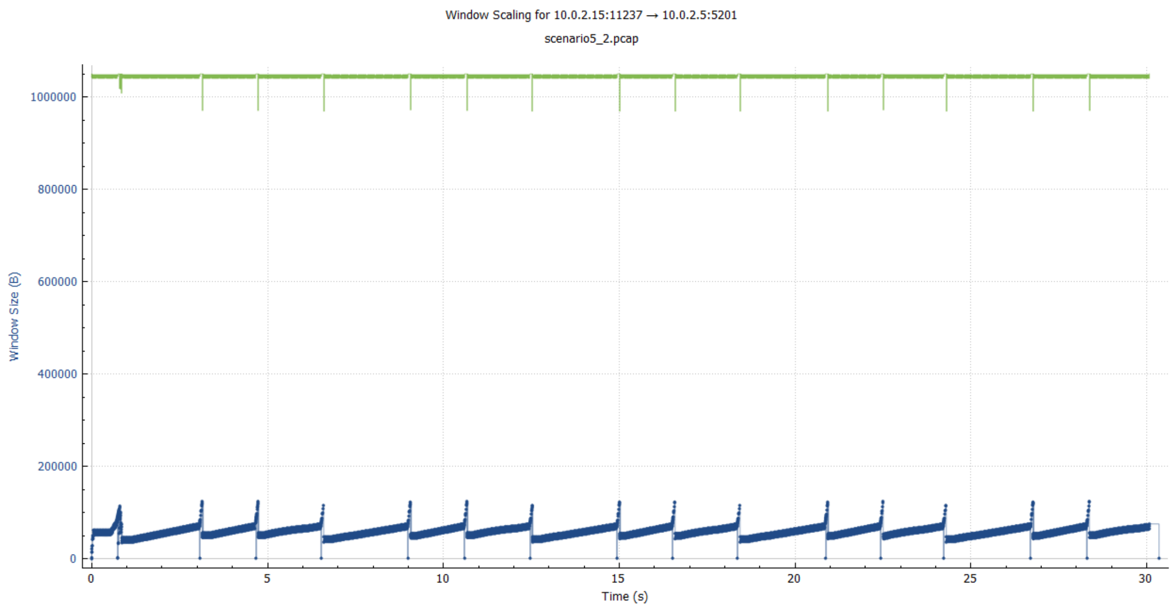
Analysis:

From the first scenario, we can see one slow start in the beginning. We see about three congestion avoidance, indicated by the rising spike. Lastly, we see three fast recovery, indicated by the break of the spike drop. From the second scenario, we can see one slow start in the beginning. We see much more congestion avoidance, indicated by the rising spike, compared to the first scenario. Lastly, we also see a lot of fast recoveries, compared to the first scenario. We can conclude that both algorithms are optimized, however the cubic algorithm performs better as it can reach the available bandwidth much faster than newreno.

Scenario 5_1: 8Mbit/s Bandwidth, 1% Packet Loss Rate, Algorithm newreno



Scenario 5_2: 8Mbit/s Bandwidth, 1% Packet Loss Rate, Algorithm cubic



Analysis:

From the first scenario, we can see one slow start in the beginning. We see multiple congestion avoidance, indicated by the rising spike. Lastly, we see multiple fast recovery, indicated by the break of the spike drop. From the second scenario, we can see one slow start in the beginning. We see less congestion avoidance, indicated by the rising spike, compared to the first scenario. Lastly, we also see less fast recoveries, compared to the first scenario. We can conclude that both algorithms are optimized, however in this scenario, the newreno is performing better.

Problem Faced

During the environment setup process, I struggled a little bit as I have no experience with using virtual machines because I didn't use virtual machines to do Assignment 1. I happened to have a spare harddrive, hence for Assignment 1, I installed ubuntu to that spare harddrive. Due to those reasons, I had to spend more time reading documentation on the internet to make sure that everything would be installed correctly.

During the process of saving captured packets to a .pcap file, I also had a few issues as I can’t seem to save the file to the shared_data folder. Then, I realized that I have to do the mount command everytime I reload the VM server. After realizing that, I had no more problems about saving the .pcap file.