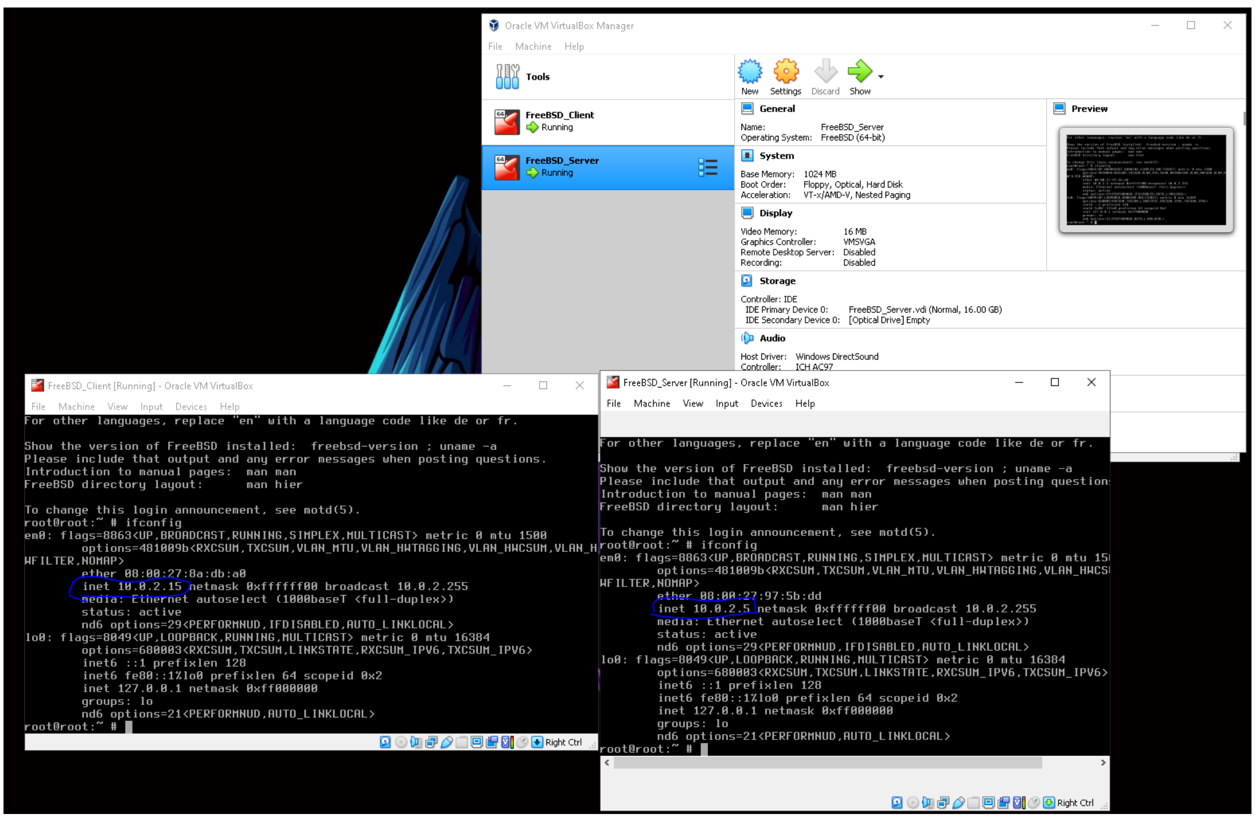
Assignment 2

Tuguldur Tserenbaljir (鐵特德) 109006271

Part 1: Environment Setup)

To set up my environment, I downloaded the Oracle virtual machine and the two virtual machines from google drive. After successfully installing the Oracle virtual machine, the two files are added and configured on their network where their IP address is shown below by following the exact steps from Appendix A **(Figure 1.1).**

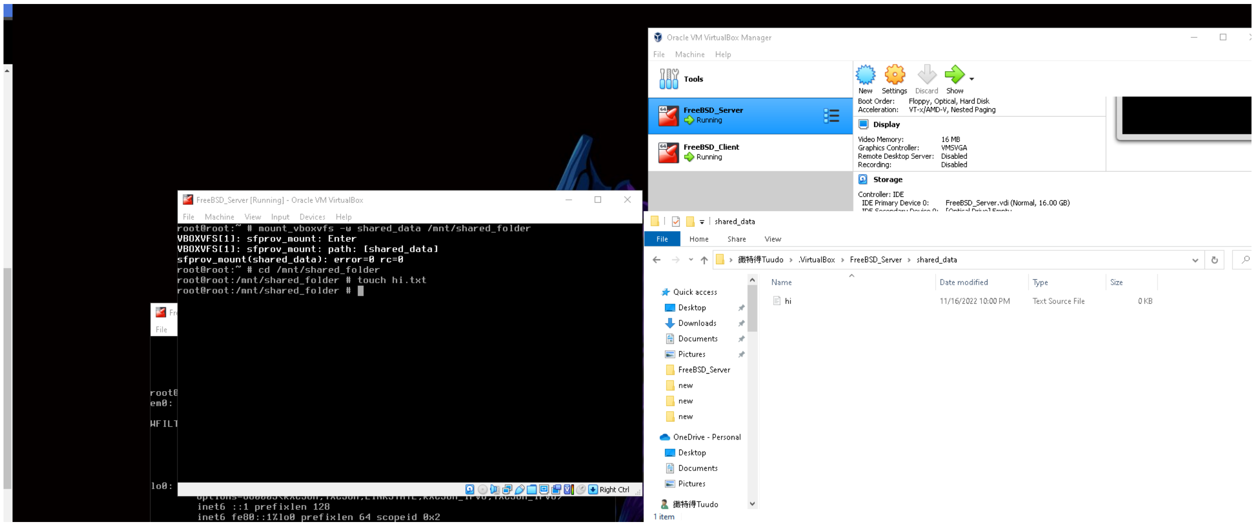
Once I run the two files, I create a NAT network and set up their network. Thus, when I run the command “ifconfig” the following information shows up. This shows that their network is set up successfully.



(Figure 1.1)

The next step was to create a shared folder. I create a **“shared_data”** folder on the host inside the Server folder. After creating a shared folder, the setup on the Oracle Virtual machine is configured. Once I am done with all of the steps on the host, I make a folder on the Server by the following command “**mkdir /mnt/shared_folder**”. Next, mounted the two files by using the command “**mount_vboxvfs -w shared_data /mnt/shared_folder**”. Once I was done mounting, I go to the directory

“**/mnt/shared_folder**” and use the command **“touch hi.txt”** to test if it is done successfully **(Figure 1.2)**.



(Figure 1.2)

The **“hi.txt”** in our host shows that it was done successfully.

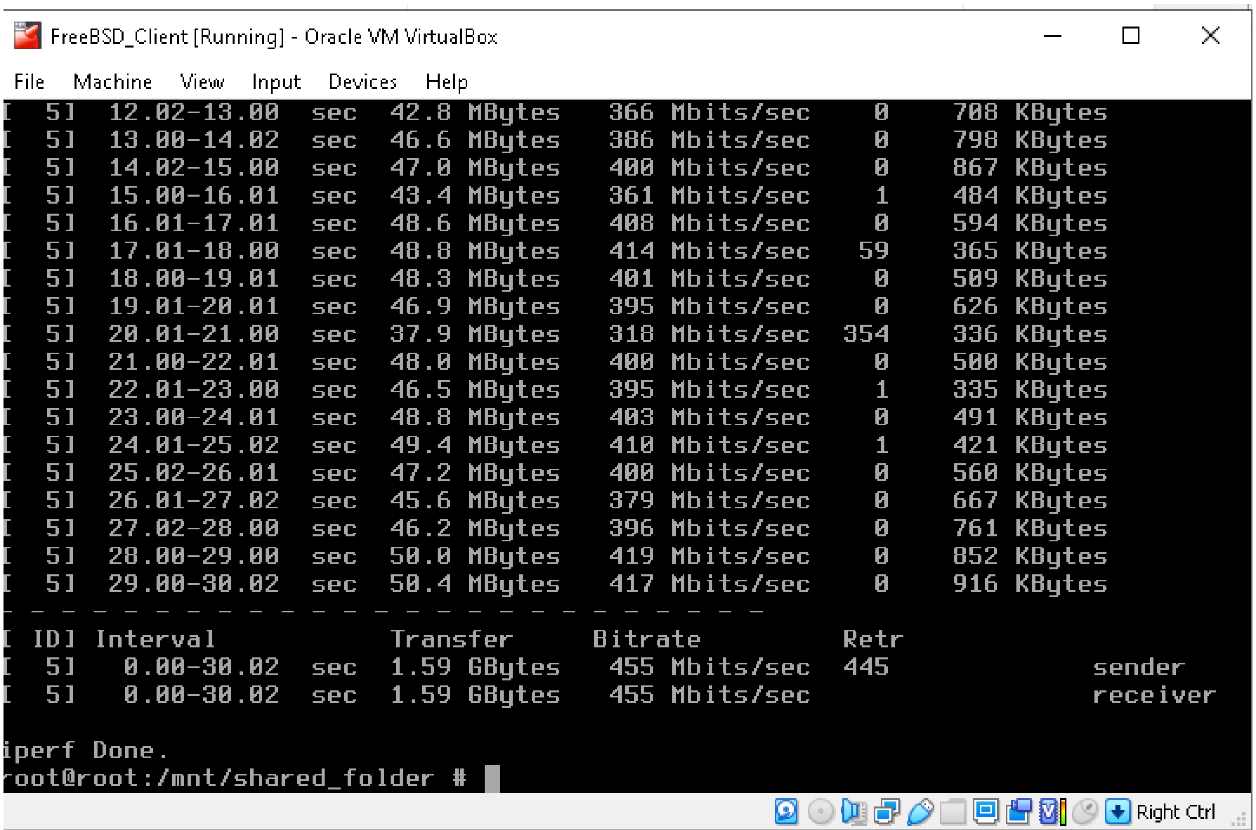
Part 2: TCP Data Transmission)

1) To run the iperf server, “**iperf3 -s &**” is entered in the FreeBSD_Server. When we get the **“Server is listening on …”** message, we can start step 2.
2) Step 2 is to start tcpdump and run the iperf client right after it. So on the server side, the command “**timeout 35 tcpdump -i em0 -w /mnt/shared_folder/tcp_dump.pcap**”.
3) On the client side, I enter the command **iperf3 -c 10.0.2.15 -t30 -i1 -w 1M -C newreno**

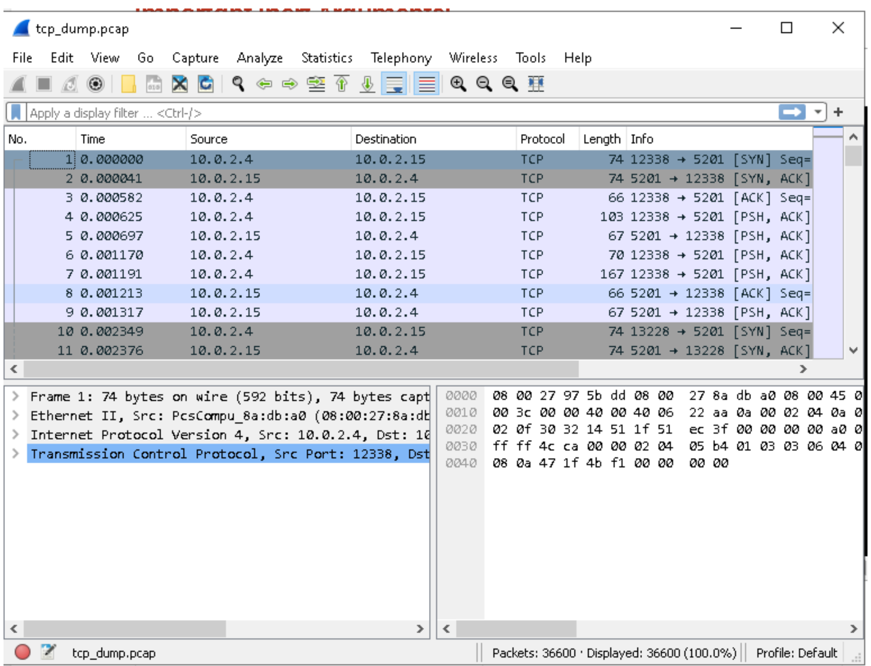
   -c <10.0.2.15> is for running client at the IP address of FreeBSD_Server

   -t30 the timeout in seconds. In our case, it is 30 seconds.

-i1 Output of each interval in seconds. In our case, it is 1 second.



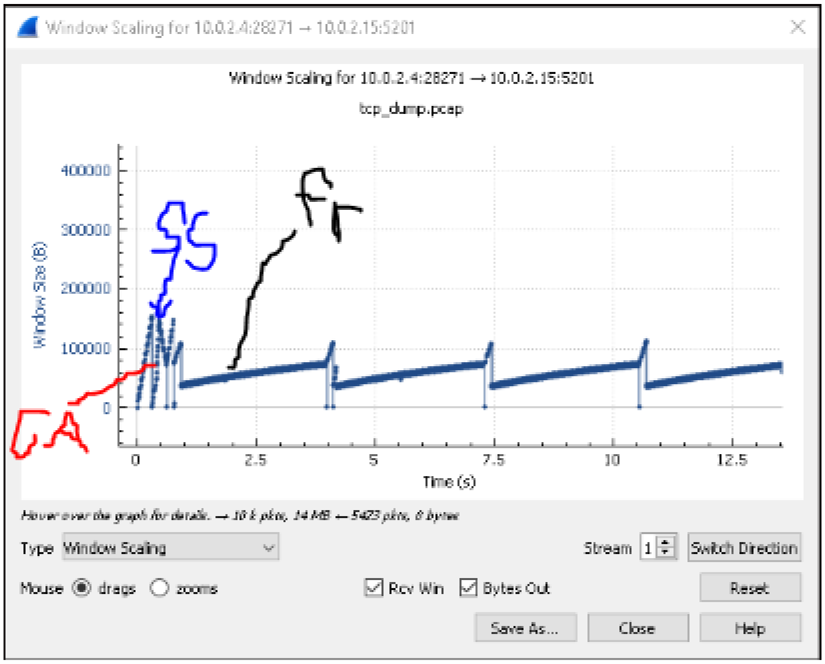
In addition, we can open our **"tcp_dump.pcap"** file on our host with Wireshark.



Part 3: TCP Congestion Control)

Scenario 1) Delay, packet loss default

4Mbit/s

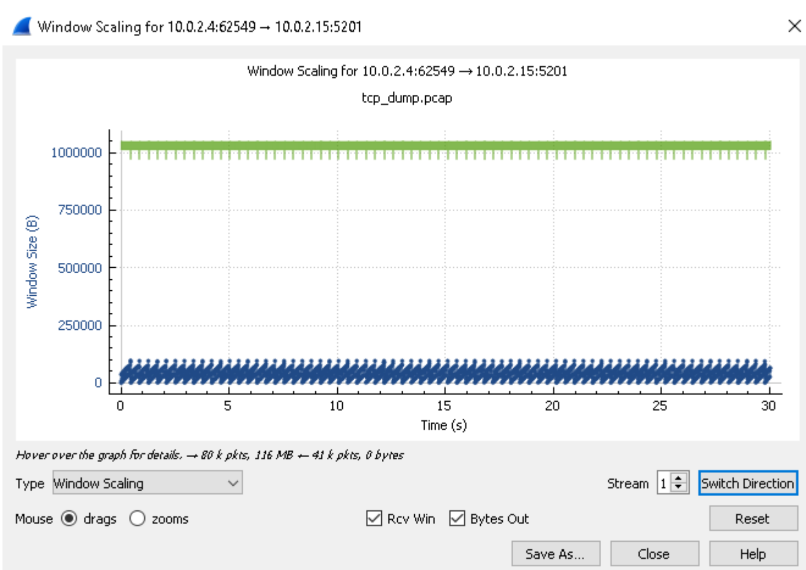


Slow start- it grows until around 16000 bits and 1 second (blue).

Congestion avoidance- where it drops gradually drops from the very top(red).

Fast recovery- cut the bit in half and increase linearly (black).

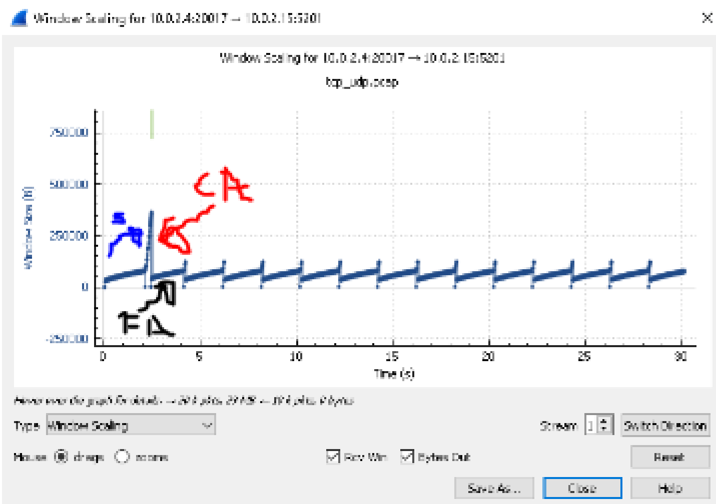
32Mbit/s



Slow start should start from the beginning where it should grow exponentially, but since the system is transmitting too fast, it has to be congestion avoidance. Thus, it cuts half and grows linearly. The sequence was it slows and starts too fast that it falls into the congestion avoidance then fast starts every time and that is why the graph is like this for 32Mbit/s.

Scenario 2) 8Mbit/s, packet loss default

Delay 10ms

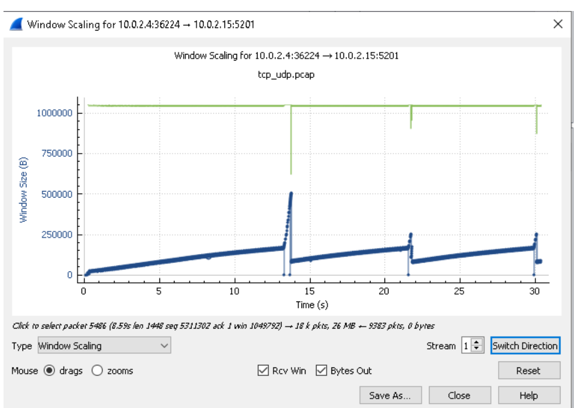


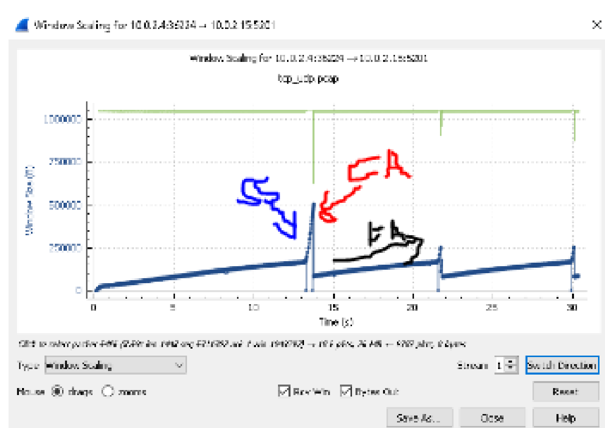
Slow start- it grows until around 35000 bits and 3 seconds (blue).

Congestion avoidance- where it drops gradually drops from the very top(red).

Fast recovery- cut the bit in half and increase linearly (black).

Delay 100ms





Slow start- it grows until around 15000-50000 bits and 13-14 seconds (blue).

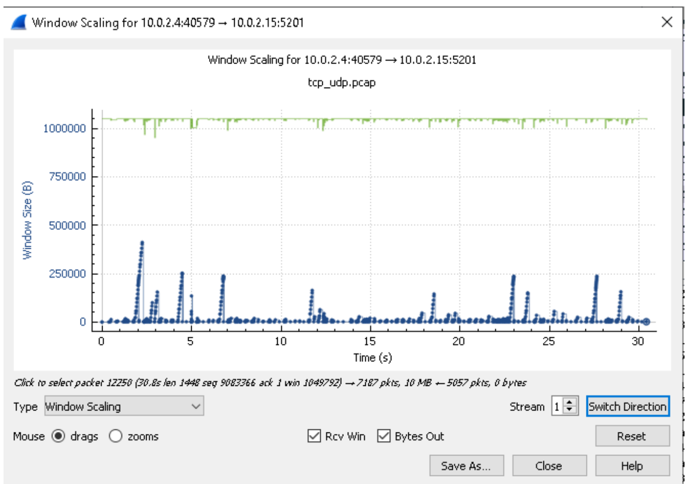
Congestion avoidance- where it drops gradually drops from the very top(red).

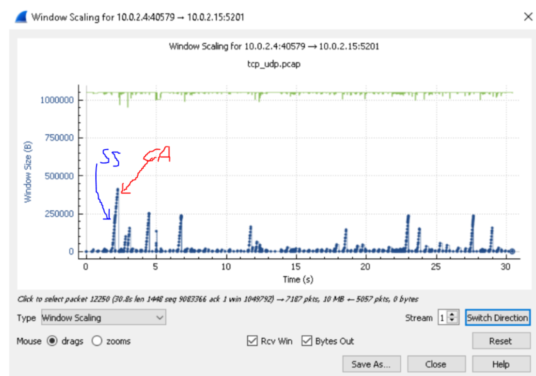
Fast recovery- cut the bit in half and increase linearly (black).

The longer the delay, it will have the longest period of Fast recovery as we see from the 2 graphs. Both graphs are delayed to their respective delay parameter before going into the slow start where both gradually follow the congestion avoidance. Next, it also delayed where it grows linearly respective to the Fast recovery.

Scenario 3)

Packet loss 0.1



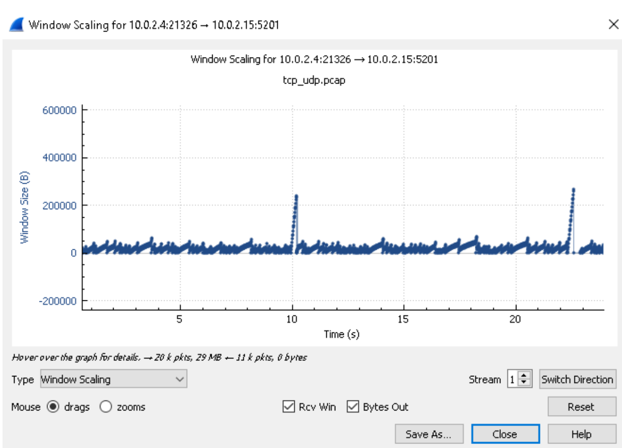


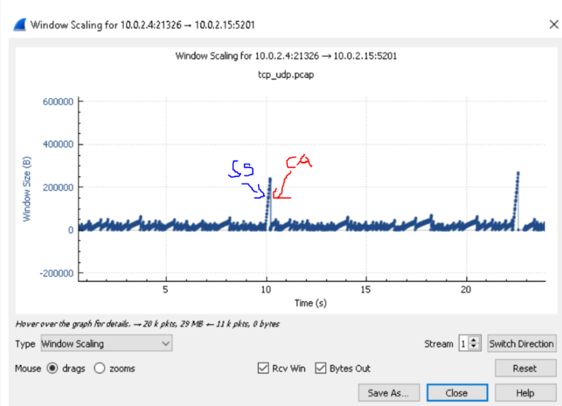
Slow start- Around the 3-second mark it grows exponentially(blue).

Congestion avoidance- where it drops gradually drops from the very top(red).

Right after the congestion avoidance, we should see the graph growing linearly for the Fast Recovery, but since we have a packet loss rate of 0.1, the graph looks so. Every packet has a 10 percent chance of getting lost which is very high.

Packet loss 1 %





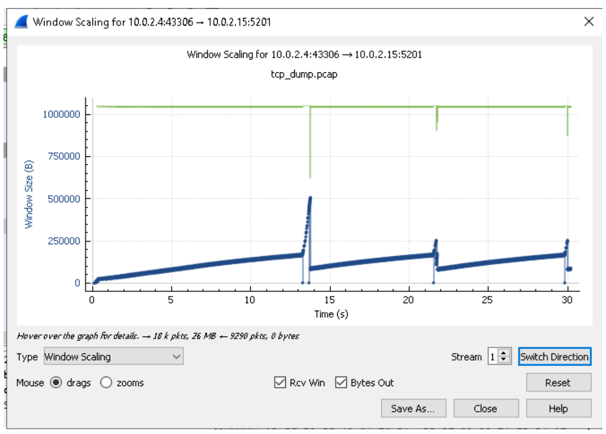
Slow start- Around the 10-second mark it grows exponentially(blue).

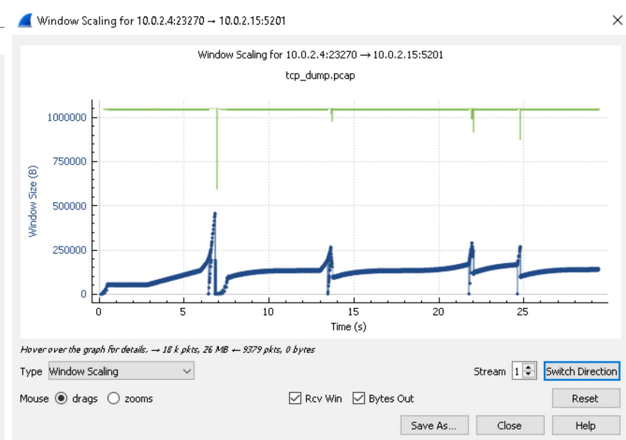
Congestion avoidance- where it drops gradually drops from the very top(red). Since we do not have a high packet loss rate and delay, the graph will look similar to Scenario 1's 32Mbit/s. Because almost every packet is going through at a respective high speed, the congestion avoidance will try to make sure it will not go over the limit. After the congestion avoidance is activated, it will linearly grow and fall into a congestion avoidance.

Scenario 4)

8Mbit/s, delay 100ms and pack loss rate default

Newreno **Cubic**



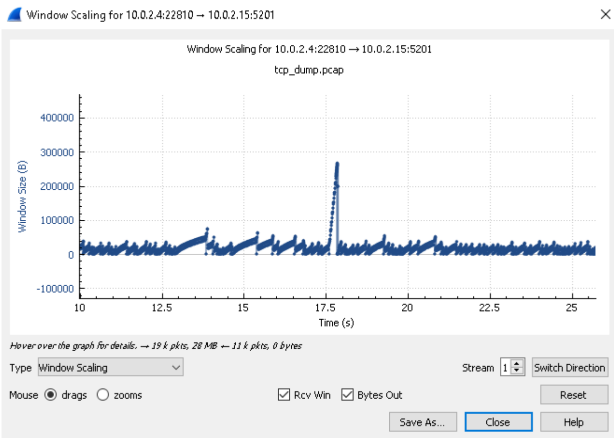


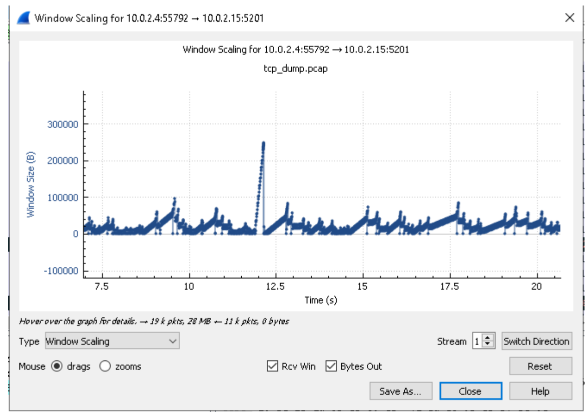
The slow start, congestion avoidance, and fast recovery from our previous experiments of 8Mbit/s, 100ms and pack loss rate being default did not change much. However, if we set the "cc_algo" to cubic, we see a change right away. First, of all the lines seem to look like it is curving instead of the linear one we saw before. Thus, around the 6-7 second mark we see that the graph is exponentially growing and having more slow starts than the newreno graph. This is because we can optimize the long fat network where we can achieve high bandwidth connections than the newreno one.

Scenario 5)

8Mbit/s, delay default, and pack loss rate of 1%

Newreno **Cubic**





Similar to Scenario 4, we can see that the bandwidth is increased in the graph and it is certain that we can optimize it to be faster than the newreno graph. One of the main differences is that after each one goes into congestion avoidance where both graphs fall to 0 bit, the Cubic graph does a fast recovery whereas the Newreno graph follows the congestion which follows a pattern.

Problems I faced:

-Did not start the command "timeout 35 tcpdump -i em0 -w /mnt/shared_folder/{filename}.pcap" every time I deleted and made a new pipeline in the server. It kept showing the same graph and outputs and it took me a while to find out.

-It was my first time using Wireshark and it sounded overwhelming because from everyone I hear it is an advanced program for OS.

- Did not permanently mount my folder where my Wireshark was not getting any pcap files.