Name: Alden Rivera / 明文

Student ID: 0416329

Date: 2017/10/13

**Lab 1 Report**

Output1

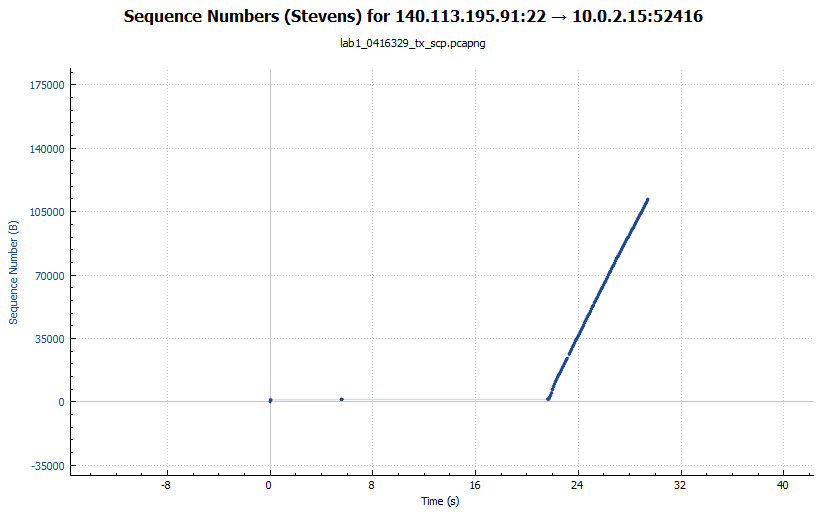
For the first part of our lab assignment we first need to generate an “upload file”

By using the command “dd if=/dev/zero of=test bs=1M count=300”. Then we open wireshark and start sniffing for traffic. We upload our file to the server with the command “scp <filename> [user@140.113.195.91:/tmp/XXXXXXX](mailto:user@140.113.195.91:/tmp/XXXXXXX)” (XXXXXXX = student id) and we save our sniffing session.

With our saved file we start filtering our packets, in my case I used the ip.src filter instruction, more specifically ip.src==140.113.195.91 because that is the IP address of the computer we are currently using.

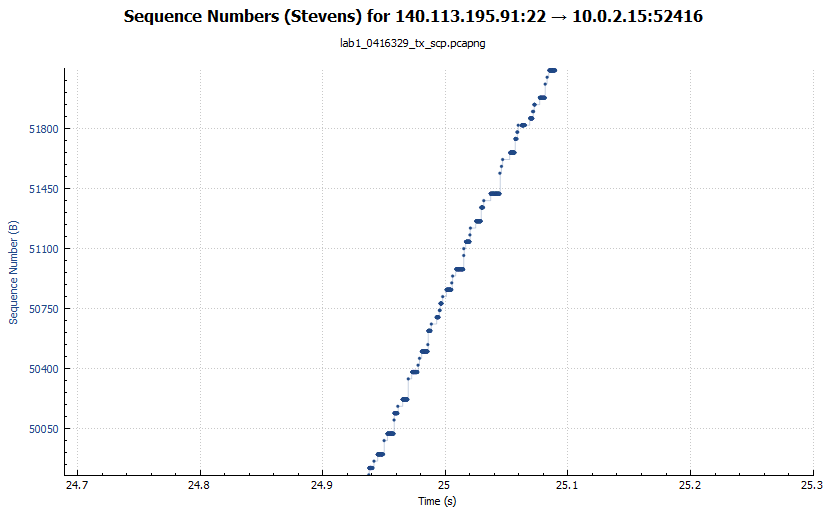
Finally to generate our graph we to the tool bar, choose statistics then TCP Stream Graphs and finally sequence numbers(Stevens). Thus we get the following graphs (one is zoomed out and one zoomed in).

Figure 1.a



For this first image we can see a stable upload rate, one minor gap at 23 seconds and the whole upload took 30 seconds.

Figure 1.b (Zoomed in version)



In this zoomed in version we can see a more accurate number for the bits and a certain time.

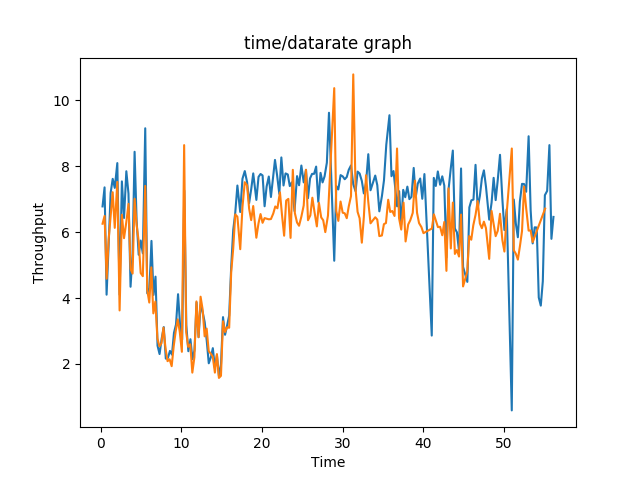
**Output 2**

The task to accomplish here is to make a graph for the average throughput of two different concurrent file downloads.

The first step is to download the “two\_connection.sh” file from e3 which if opened we can see two GET requests inside of it. We then start capturing our traffic and use the two\_connection.sh file in conjunction to the address of the file we want to target as following “./two\_connection.sh <http://140.113.195.91/testfile>”. Again, we save our captured traffic but this time in a “.pcap” format so we are able to use with our script to calculate the throughput and plot our graphs.

To calculate the throughput we first go through all of our packets checking to which connection it belongs to by using their ips and destination ports and with the given formula we calculate and plot each transfer sequence.

This was the end result:



Both connections have mostly a similar throughput, when one connection is low the other is as well but we can also see the spikes from the orange connection normally occur when there is a dip in the throughput and vice versa meaning in my opinion that it is because of the shared bandwidth and if they were to cancel out we’d have a stable throughput.