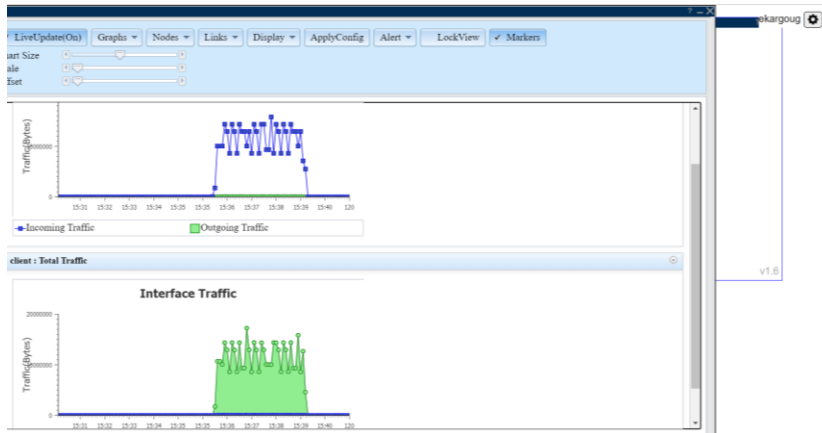


LAB 2

Exercise 1.1

1)

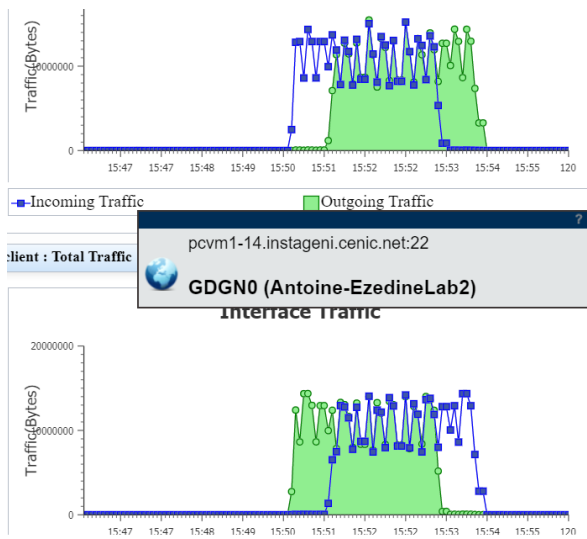


Max:15791324.7199999999

Min:1769966.7600000002

Average: 8780645.74

2)



```

ekargoug@server:~$ -----
Client connecting to client, TCP port 5001
TCP window size: 45.0 KByte (default)
-----
[ 3] local 10.1.1.2 port 33966 connected with 10.1.1.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec  99.1 MBytes 83.2 Mbits/sec
[ 3] 10.0-20.0 sec  104 MBytes 87.1 Mbits/sec
[ 3] 20.0-30.0 sec  98.8 MBytes 82.8 Mbits/sec
[ 3] 30.0-40.0 sec  105 MBytes 87.8 Mbits/sec
[ 3] 40.0-50.0 sec  98.6 MBytes 82.7 Mbits/sec
[ 3] 50.0-60.0 sec  101 MBytes 84.4 Mbits/sec
[ 3] 60.0-70.0 sec  107 MBytes 89.7 Mbits/sec
[ 3] 70.0-80.0 sec  102 MBytes 85.7 Mbits/sec
[ 3] 80.0-90.0 sec  103 MBytes 86.5 Mbits/sec
[ 3] 90.0-100.0 sec 99.4 MBytes 83.4 Mbits/sec
[ 3] 100.0-110.0 sec 105 MBytes 87.9 Mbits/sec
[ 3] 110.0-120.0 sec 106 MBytes 88.8 Mbits/sec
[ 3] 120.0-130.0 sec 106 MBytes 88.7 Mbits/sec
[ 4]  0.0-180.1 sec 1.87 GBytes 89.0 Mbits/sec
[ 3] 130.0-140.0 sec 110 MBytes 92.0 Mbits/sec
[ 3] 140.0-150.0 sec 114 MBytes 95.4 Mbits/sec
[ 3] 150.0-160.0 sec 114 MBytes 95.2 Mbits/sec
[ 3] 160.0-170.0 sec 113 MBytes 95.1 Mbits/sec
[ 3] 170.0-180.0 sec 114 MBytes 95.5 Mbits/sec
[ 3]  0.0-180.0 sec 1.85 GBytes 88.4 Mbits/sec

Client connecting to server, TCP port 5001
TCP window size: 45.0 KByte (default)
-----
[ 3] local 10.1.1.1 port 50478 connected with 10.1.1.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec  115 MBytes 96.3 Mbits/sec
[ 3] 10.0-20.0 sec  113 MBytes 95.1 Mbits/sec
[ 3] 20.0-30.0 sec  113 MBytes 95.0 Mbits/sec
[ 3] 30.0-40.0 sec  114 MBytes 95.4 Mbits/sec
[ 4] local 10.1.1.1 port 5001 connected with 10.1.1.2 port 33966
[ 3] 40.0-50.0 sec  111 MBytes 93.0 Mbits/sec
[ 3] 50.0-60.0 sec  104 MBytes 87.7 Mbits/sec
[ 3] 60.0-70.0 sec  102 MBytes 86.0 Mbits/sec
[ 3] 70.0-80.0 sec  103 MBytes 86.3 Mbits/sec
[ 3] 80.0-90.0 sec  102 MBytes 85.1 Mbits/sec
[ 3] 90.0-100.0 sec 100 MBytes 84.0 Mbits/sec
[ 3] 100.0-110.0 sec 104 MBytes 87.3 Mbits/sec
[ 3] 110.0-120.0 sec 108 MBytes 90.3 Mbits/sec
[ 3] 120.0-130.0 sec 100 MBytes 84.0 Mbits/sec
[ 3] 130.0-140.0 sec 99.8 MBytes 83.7 Mbits/sec
[ 3] 140.0-150.0 sec 102 MBytes 85.6 Mbits/sec
[ 3] 150.0-160.0 sec 106 MBytes 88.7 Mbits/sec
[ 3] 160.0-170.0 sec 109 MBytes 91.2 Mbits/sec
[ 3] 170.0-180.0 sec 106 MBytes 88.9 Mbits/sec
[ 3]  0.0-180.0 sec 1.87 GBytes 89.1 Mbits/sec
[7]+ Done                                iperf -s

```

The client received a bandwidth ranging from 83.7 to 96.3 Mbits/sec while the server received a bandwidth ranging from 82.7 to 95.5 Mbits/sec

3)

```

ekargoug@server:~$ -----
Client connecting to client, TCP port 5001
TCP window size: 45.0 KByte (default)
-----
[ 3] local 10.1.1.2 port 33966 connected with 10.1.1.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec  99.1 MBytes 83.2 Mbits/sec
[ 3] 10.0-20.0 sec  104 MBytes 87.1 Mbits/sec
[ 3] 20.0-30.0 sec  98.8 MBytes 82.8 Mbits/sec
[ 3] 30.0-40.0 sec  105 MBytes 87.8 Mbits/sec
[ 3] 40.0-50.0 sec  98.6 MBytes 82.7 Mbits/sec
[ 3] 50.0-60.0 sec  101 MBytes 84.4 Mbits/sec
[ 3] 60.0-70.0 sec  107 MBytes 89.7 Mbits/sec
[ 3] 70.0-80.0 sec  102 MBytes 85.7 Mbits/sec
[ 3] 80.0-90.0 sec  103 MBytes 86.5 Mbits/sec
[ 3] 90.0-100.0 sec 99.4 MBytes 83.4 Mbits/sec
[ 3] 100.0-110.0 sec 105 MBytes 87.9 Mbits/sec
[ 3] 110.0-120.0 sec 106 MBytes 88.8 Mbits/sec
[ 3] 120.0-130.0 sec 106 MBytes 88.7 Mbits/sec
[ 4]  0.0-180.1 sec 1.87 GBytes 89.0 Mbits/sec
[ 3] 130.0-140.0 sec 110 MBytes 92.0 Mbits/sec
[ 3] 140.0-150.0 sec 114 MBytes 95.4 Mbits/sec
[ 3] 150.0-160.0 sec 114 MBytes 95.2 Mbits/sec
[ 3] 160.0-170.0 sec 113 MBytes 95.1 Mbits/sec
[ 3] 170.0-180.0 sec 114 MBytes 95.5 Mbits/sec
[ 3]  0.0-180.0 sec 1.85 GBytes 88.4 Mbits/sec

```

The meaning of this output is while both the client and the server were competing/sharing bandwidth, the possible amount transferable we in the high 90s to low 100s and the bandwidth was in the 80s range. However once the client's time had ran out, we see a rise in the bandwidth to mid 90s. The amount transferable also rose to the 110s range.

Exercise 1.2

1)

```
Before delay
--- client-link-0 ping statistics ---
73 packets transmitted, 73 received, 0% packet loss, time 73540ms
rtt min/avg/max/mdev = 0.266/0.533/0.853/0.105 ms
ekargoug@server:~$
```

```
After delay
--- client-link-0 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14024ms
rtt min/avg/max/mdev = 100.363/100.582/100.699/0.427 ms
ekargoug@server:~$
```

I noticed that after the delay there is a 100 ms of difference. By that I mean before the delay the RRT was in the 0 range and after the delay, it is in the 100 range

2)

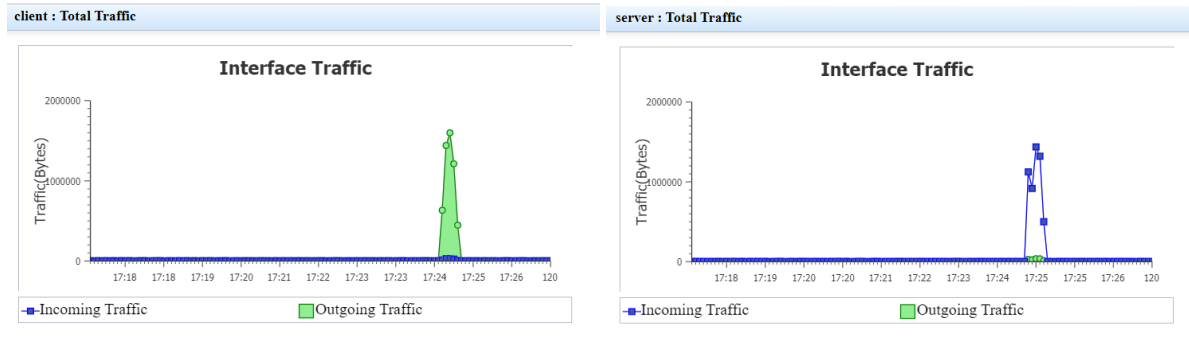
```
ekargoug@client:~$ iperf -c server -t 10
-----
Client connecting to server, TCP port 5001
TCP window size: 45.0 KByte (default)
-----
[  3] local 10.1.1.1 port 50482 connected with 10.1.1.2 port 5001
[ ID] Interval           Transfer     Bandwidth
[  3]  0.0-10.0 sec      115 MBytes  96.1 Mbits/sec
ekargoug@client:~$
```

```
-----
Client connecting to server, TCP port 5001
TCP window size: 4.50 KByte (WARNING: requested 1.95 KByte)
-----
[  3] local 10.1.1.1 port 50484 connected with 10.1.1.2 port 5001
[ ID] Interval           Transfer     Bandwidth
[  3]  0.0-10.0 sec      112 MBytes  94.1 Mbits/sec
ekargoug@client:~$
```

The average bandwidth of the 45.0 KByte window size was higher than that of the 4.50 KByte window size. There was a decrease of about 2 Mbytes/sec and I imagine that had we ran it for longer, it would have decreased even more. The server and client's matched pretty much exactly. I ran 2 test and the difference was about .1 to .2 difference in Bandwidth

3)

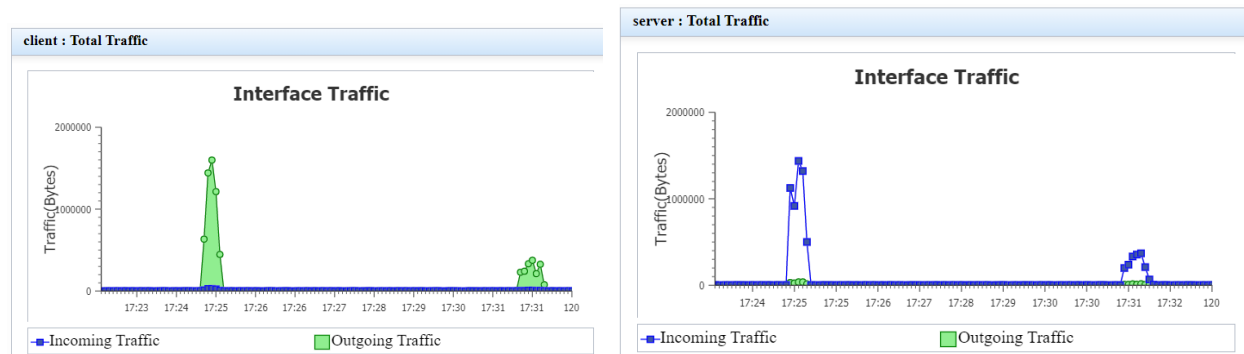
Window 4kb 50ms delay



```
ekargoug@client:~$ iperf -c server -t 20 -w 4 kb
WARNING: TCP window size set to 4 bytes. A small window size
will give poor performance. See the Iperf documentation.
iperf: ignoring extra argument -- kb
-----
Client connecting to server, TCP port 5001
TCP window size: 4.50 KByte (WARNING: requested 4.00 Byte)
-----
[ 3] local 10.1.1.1 port 50492 connected with 10.1.1.2 port
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-20.1 sec  24.8 MBytes 10.3 Mbits/sec
ekargoug@client:~$
```

```
ekargoug@server:~$ sudo tc qdisc add dev eth1 root netem delay 50ms
ekargoug@server:~$ iperf -s -w 4 kb
WARNING: TCP window size set to 4 bytes. A small window size
will give poor performance. See the Iperf documentation.
iperf: ignoring extra argument -- kb
bind failed: Address already in use
ekargoug@server:~$ [ 4] local 10.1.1.2 port 5001 connected with 10.1.1.1 port 50492
[ 4] 0.0-20.1 sec  24.8 MBytes 10.3 Mbits/sec
```

Window 4kb, 250ms delay

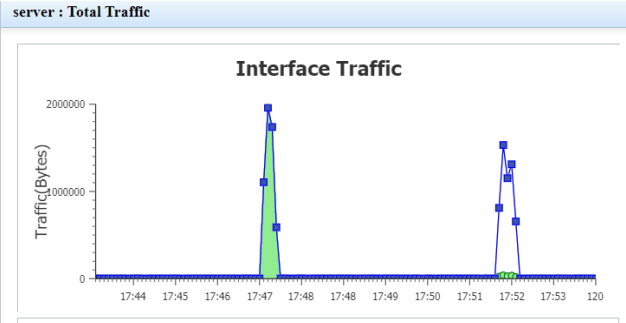
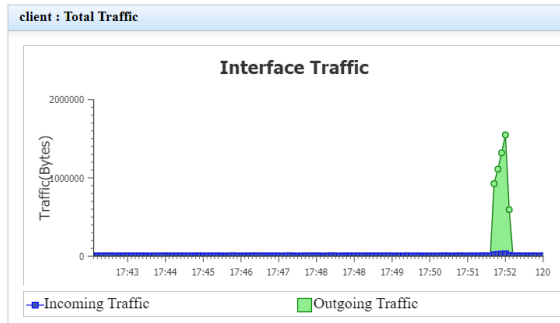


```
-----
Client connecting to server, TCP port 5001
TCP window size: 4.50 KByte (WARNING: requested 4.00 Byte)
-----
[ 3] local 10.1.1.1 port 50494 connected with 10.1.1.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-30.1 sec  8.25 MBytes 2.30 Mbits/sec
ekargoug@client:~$
```

```
sudo tc qdisc add dev eth1 root netem delay 250ms
RTNETLINK answers: File exists
ekargoug@server:~$ sudo tc qdisc del dev eth1 root
ekargoug@server:~$ sudo tc qdisc add dev eth1 root netem delay 250ms
ekargoug@server:~$ [ 4] 0.0-30.1 sec  8.25 MBytes 2.30 Mbits/sec
```

Window 32kb, 50ms delay

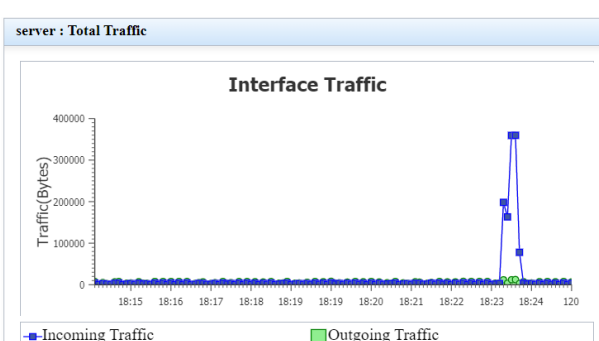
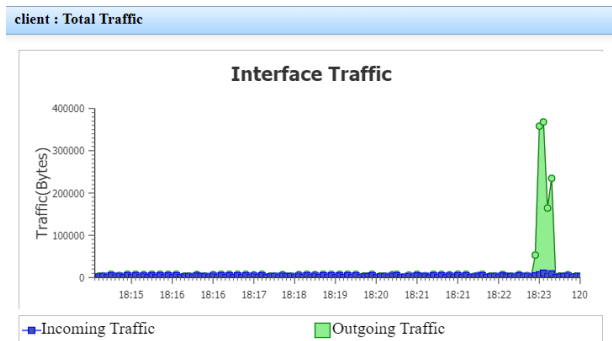
(Ignore the first spike of the server side)



```
ekargoug@client:~$ iperf -c server -t 20 -w 32kb
-----
Client connecting to server, TCP port 5001
TCP window size: 62.5 KByte (WARNING: requested 31.2 KByte)
-----
[ 3] local 10.1.1.1 port 50496 connected with 10.1.1.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-20.0 sec  25.5 MBytes 10.7 Mbits/sec
```

```
ekargoug@server:~$ sudo tc qdisc add dev eth1 root netem delay 50ms
ekargoug@server:~$ iperf -s -w 32kb
bind failed: Address already in use
ekargoug@server:~$ [ 4] local 10.1.1.2 port 5001 connected with 10.1.1.1 port 50496
[ 4] 0.0-20.0 sec  25.5 MBytes 10.7 Mbits/sec
```

Window 32kb, 250ms delay



```
ekargoug@client:~$ iperf -c server -t 20 -w 32kb
-----
Client connecting to server, TCP port 5001
TCP window size: 62.5 KByte (WARNING: requested 31.2 KByte)
-----
[ 3] local 10.1.1.1 port 50500 connected with 10.1.1.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-20.1 sec  5.38 MBytes 2.24 Mbits/sec
```

```
ekargoug@server:~$ sudo tc qdisc add dev eth1 root netem delay 250ms
ekargoug@server:~$ iperf -s -w 32kb
bind failed: Address already in use
ekargoug@server:~$ [ 4] local 10.1.1.2 port 5001 connected with 10.1.1.1 port 50500
[ 4] 0.0-20.1 sec  5.38 MBytes 2.24 Mbits/sec
```

The reason there is such a huge difference in the 4kb, 250ms and the 32kb, 50ms is because the bigger the TCP window is the more bytes it is able to handle, and with a 1/5 delay time, it can handle it at faster speeds

Exercise 2.1

```
ekargoug@switch:~$ sudo ovs-vs-vsctl add-port br0 eth1
sudo: ovs-vs-vsctl: command not found
ekargoug@switch:~$ sudo ovs-vsctl add-port br0 eth1
ekargoug@switch:~$ sudo ovs-vsctl add-port br0 eth2
ekargoug@switch:~$ sudo ovs-vsctl add-port br0 eth3
ekargoug@switch:~$ sudo ovs-vsctl list-port br0
ovs-vsctl: unknown command 'list-port'; use --help for help
ekargoug@switch:~$ sudo ovs-vsctl list-ports br0
eth1
eth2
eth3
```

After I had removed their IP addresses, I changed their port and the screenshot proves it as such

Exercise 2.2

```
ekargoug@switch:~$ sudo ovs-vsctl set-controller br0 tcp:143.215.216.200:6653
ekargoug@switch:~$ sudo ovs-vsctl set-fail-mode br0 secure
ekargoug@switch:~$ sudo ovs-vsctl show
7f96f552-dc1f-49d2-8d23-449b84165a2c
    Bridge "br0"
        Controller "tcp:143.215.216.200:6653"
        fail_mode: secure
        Port "eth3"
            Interface "eth3"
        Port "br0"
            Interface "br0"
            type: internal
        Port "eth2"
            Interface "eth2"
        Port "eth1"
            Interface "eth1"
    ovs_version: "2.9.5"
```

As the screenshot shows, my switch is pointed at a controller. I did so by first running the command: `sudo ovs-vsctl set-controller br0 tcp:143.215.216.200:6653` after that I changed the fail-safe-mode. Pretty much I ended up with this screenshot by following the instructions provided

Exercise 2.3