Computer Networks

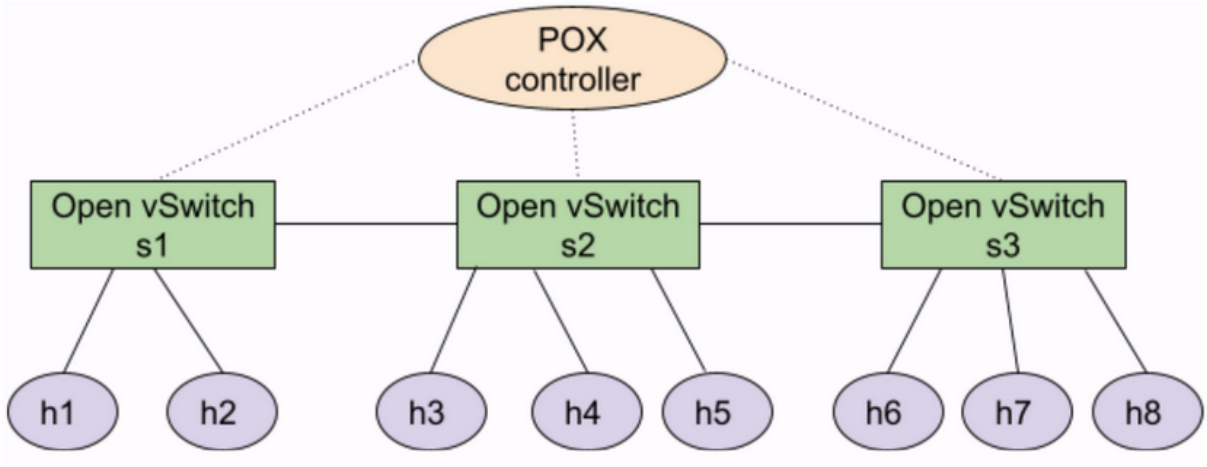
Assignment-4

Sachin Sharma

2021559, Section-B

Q1

Created a custom network topology using mininet python API



Started POX controller

```
mininet@mininet-vm:~$ sudo pox/pox.py forwarding.l2_learning
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
WARNING:version:Support for Python 3 is experimental.
INFO:core:POX 0.7.0 (gar) is up.
```

```
mininet@mininet-vm:~$ sudo mn --custom ~/mininet/custom/new_topo.py --topo newtopo
--mac --controller=remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (h5, s3) (h6, s3) (h7, s3) (h8, s3) (s1, s2) (s
2, s3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet>
```

Verified the network

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s2-eth1
h4 h4-eth0:s2-eth2
h5 h5-eth0:s3-eth1
h6 h6-eth0:s3-eth2
h7 h7-eth0:s3-eth3
h8 h8-eth0:s3-eth4
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:s2-eth3
s2 lo:  s2-eth1:h3-eth0 s2-eth2:h4-eth0 s2-eth3:s1-eth3 s2-eth4:s3-eth5
s3 lo:  s3-eth1:h5-eth0 s3-eth2:h6-eth0 s3-eth3:h7-eth0 s3-eth4:h8-eth0 s3-eth5:s2-eth4
c0
```

Q2

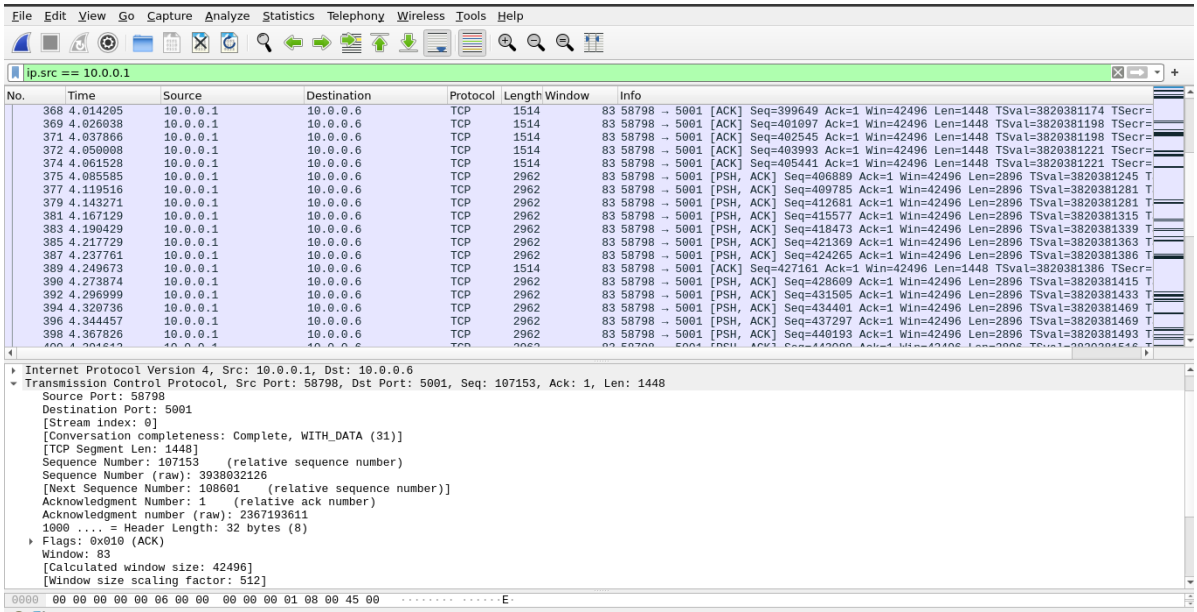
a. Created bottleneck bandwidth at host h1 of 1024kbps with 10% loss to simulate congestion using tc

```
mininet> h1 tc qdisc add dev h1-eth0 root netem rate 1024kbit loss 10% delay 10ms
mininet>
```

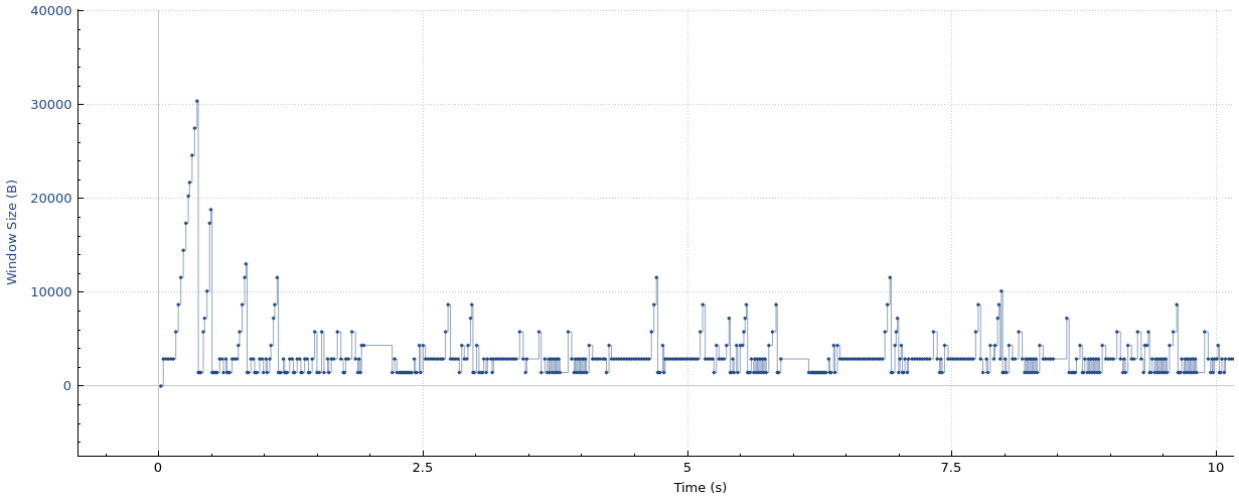
b. Generated traffic between h1 and h6 using iperf

```
mininet> h6 iperf -s &
mininet> h1 iperf -c h6 -t 10
------------------------------------------------------------
Client connecting to 10.0.0.6, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 38194 connected with 10.0.0.6 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.8 sec  1.25 MBytes   975 Kbits/sec
mininet>
```

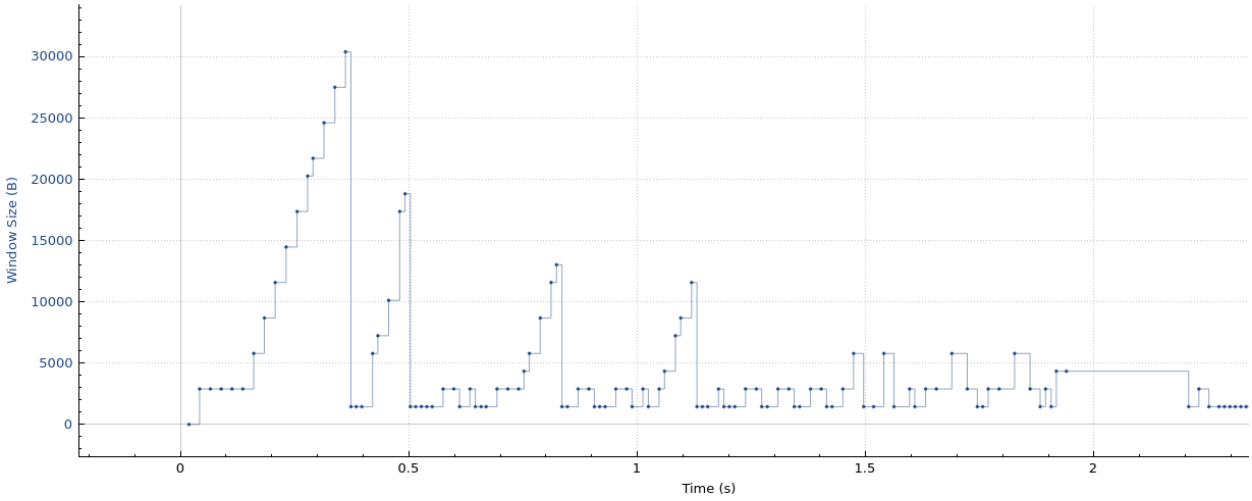
c. Captured the traffic using Wireshark at host h1



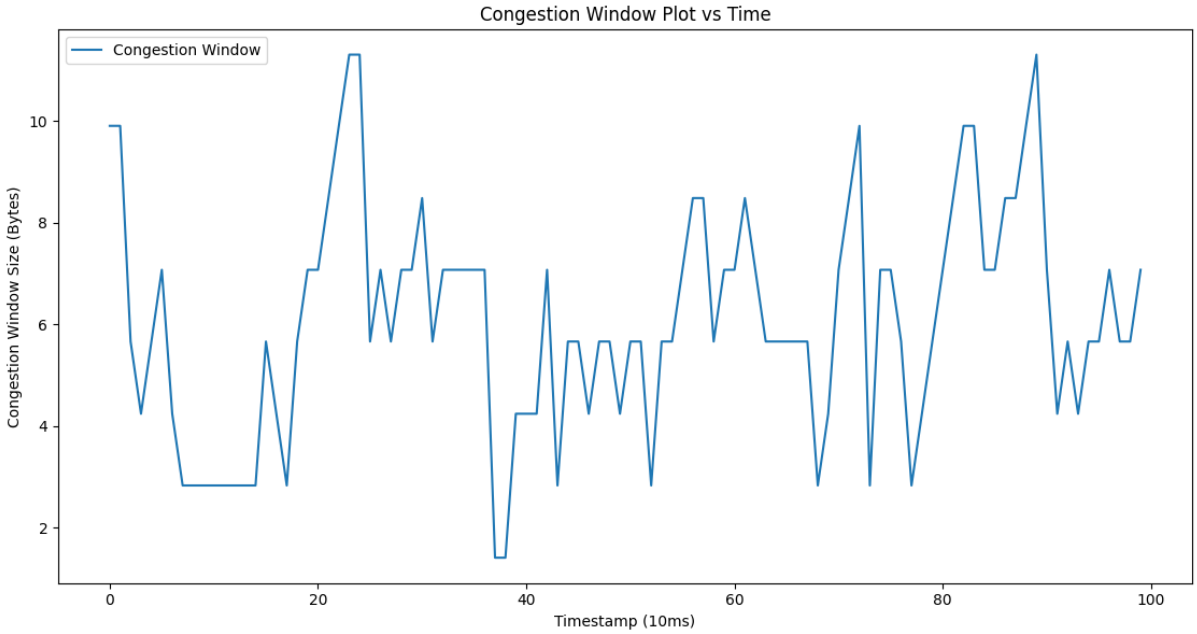
d. Plot of the congestion window size



Zooming in



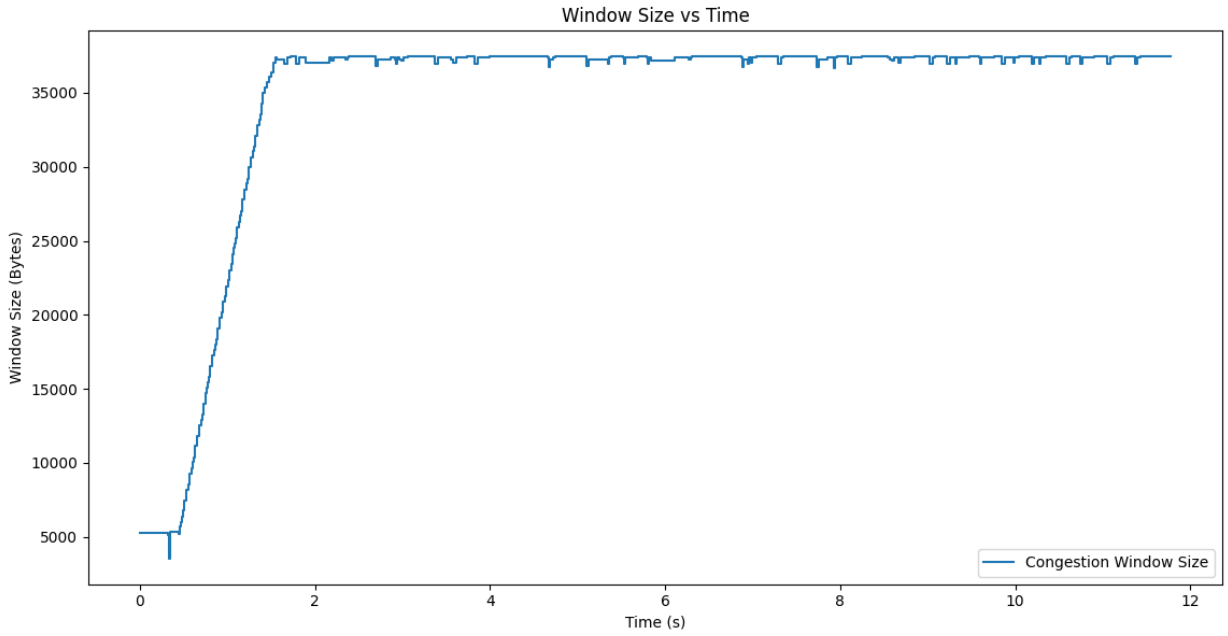
Snapshot of it



TCP's congestion control algorithm works to avoid and mitigate congestion in the network. Sender starts by sending CWND_INIT packets to receiver, and then starts exponentially increasing the number of packets until a predefined SS_THRESH. This is Slow Start period. After this, sender increments the number of packets by 1, leading to Linear Rising period or Congestion Avoidance period. Once it encounters a packet loss, it drops the CWND to init value and halves the SS_THRESH to again start the Slow Start period, as evident from the plot.

The y-axis of the plot denotes the size of the congestion window in bytes, and the x-axis is the time in seconds.

Similarly, we can also plot the RECV_WINDOW size as received from the server.



We can see that the RECV_WINDOW size slowly increases at the sender starts sending more packets, and then saturates as the subsequent adjustments will be handled by CWND size at the sender.

This can be proven by CWND given by iperf3.

```
root@mininet-vm:/home/mininet/mininet/netcaptures# iperf3 -c 10.0.0.6 -t 10 -i 0.1 --logfile cwnd.csv
root@mininet-vm:/home/mininet/mininet/netcaptures# cat cwnd.csv
Connecting to host 10.0.0.6, port 5201
[  7] local 10.0.0.1 port 57460 connected to 10.0.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-0.10   sec  77.8 KBytes  6.35 Mbits/sec    2   9.90 KBytes
[  7]   0.10-0.20   sec  0.00 Bytes  0.00 bits/sec    1   9.90 KBytes
[  7]   0.20-0.30   sec  46.7 KBytes  3.82 Mbits/sec    2   5.66 KBytes
[  7]   0.30-0.40   sec  0.00 Bytes  0.00 bits/sec    2   4.24 KBytes
[  7]   0.40-0.50   sec  0.00 Bytes  0.00 bits/sec    0   5.66 KBytes
[  7]   0.50-0.60   sec  0.00 Bytes  0.00 bits/sec    0   7.07 KBytes
[  7]   0.60-0.70   sec  0.00 Bytes  0.00 bits/sec    4   4.24 KBytes
[  7]   0.70-0.80   sec  53.7 KBytes  4.41 Mbits/sec    2   2.83 KBytes
[  7]   0.80-0.90   sec  0.00 Bytes  0.00 bits/sec    1   2.83 KBytes
[  7]   0.90-1.00   sec  0.00 Bytes  0.00 bits/sec    1   2.83 KBytes
[  7]   1.00-1.10   sec  0.00 Bytes  0.00 bits/sec    1   2.83 KBytes
[  7]   1.10-1.20   sec  0.00 Bytes  0.00 bits/sec    1   2.83 KBytes
[  7]   1.20-1.30   sec  0.00 Bytes  0.00 bits/sec    2   2.83 KBytes
[  7]   1.30-1.40   sec  0.00 Bytes  0.00 bits/sec    2   2.83 KBytes
[  7]   1.40-1.50   sec  41.0 KBytes  3.39 Mbits/sec    1   2.83 KBytes
[  7]   1.50-1.60   sec  0.00 Bytes  0.00 bits/sec    0   5.66 KBytes
[  7]   1.60-1.70   sec  0.00 Bytes  0.00 bits/sec    2   4.24 KBytes
[  7]   1.70-1.80   sec  0.00 Bytes  0.00 bits/sec    1   2.83 KBytes
[  7]   1.80-1.90   sec  39.6 KBytes  3.26 Mbits/sec    0   5.66 KBytes
[  7]   1.90-2.00   sec  0.00 Bytes  0.00 bits/sec    0   7.07 KBytes
[  7]   2.00-2.10   sec  0.00 Bytes  0.00 bits/sec    0   7.07 KBytes
[  7]   2.10-2.20   sec  39.6 KBytes  3.24 Mbits/sec    0   8.48 KBytes
[  7]   2.20-2.30   sec  0.00 Bytes  0.00 bits/sec    0   9.90 KBytes
[  7]   2.30-2.40   sec  0.00 Bytes  0.00 bits/sec    0   11.3 KBytes
[  7]   2.40-2.50   sec  0.00 Bytes  0.00 bits/sec    0   11.3 KBytes
[  7]   2.50-2.60   sec  42.4 KBytes  3.49 Mbits/sec    3   5.66 KBytes
[  7]   2.60-2.70   sec  0.00 Bytes  0.00 bits/sec    0   7.07 KBytes
[  7]   2.70-2.80   sec  0.00 Bytes  0.00 bits/sec    2   5.66 KBytes
[  7]   2.80-2.90   sec  56.6 KBytes  4.63 Mbits/sec    0   7.07 KBytes
```