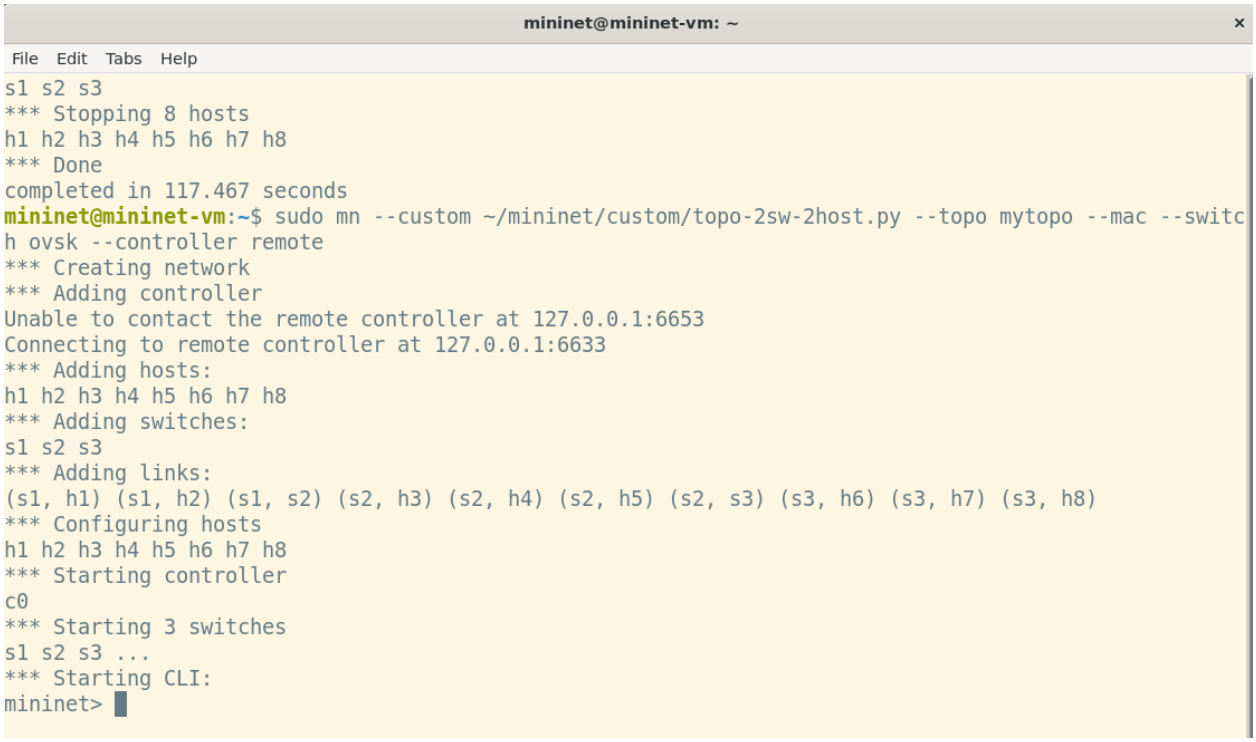
CN ASSIGNMENT 4
NIKITA RAJESH VERMA
2021546

Ques 1)
The custom network topology is created in python file and then the following command is run to start mininet -
sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo --mac --switch ovsk --controller remote



The topology is verified by the command mininet> net

Ques 2)

(a)

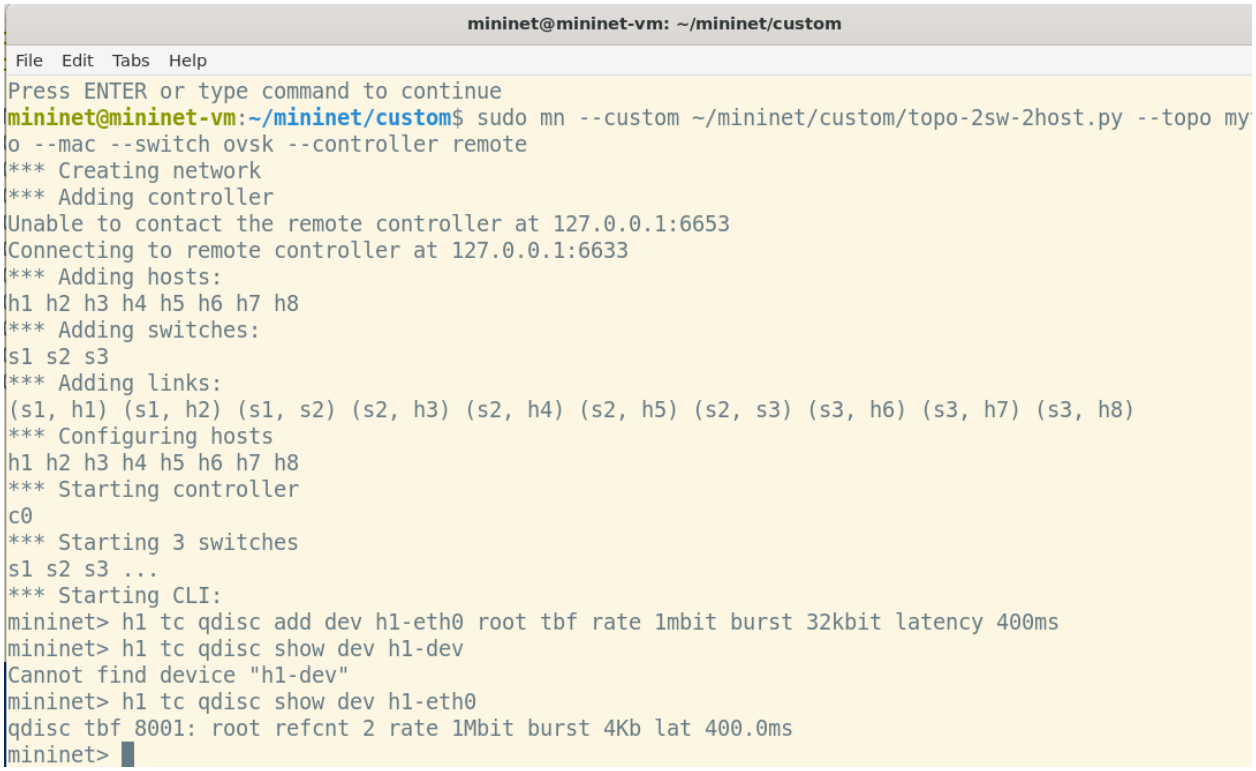
Commands ran -

cd pox

./pox.py forwarding.l2_learning

mininet> h1 tc qdisc add dev h1-eth0 root tbf rate 1mbit burst 32kbit latency 400ms

h1 tc qdisc show dev h1-eth0



(b) and (c)

xterm h1 h1 h6

In h1 window :

Packets captured using tcpdump -

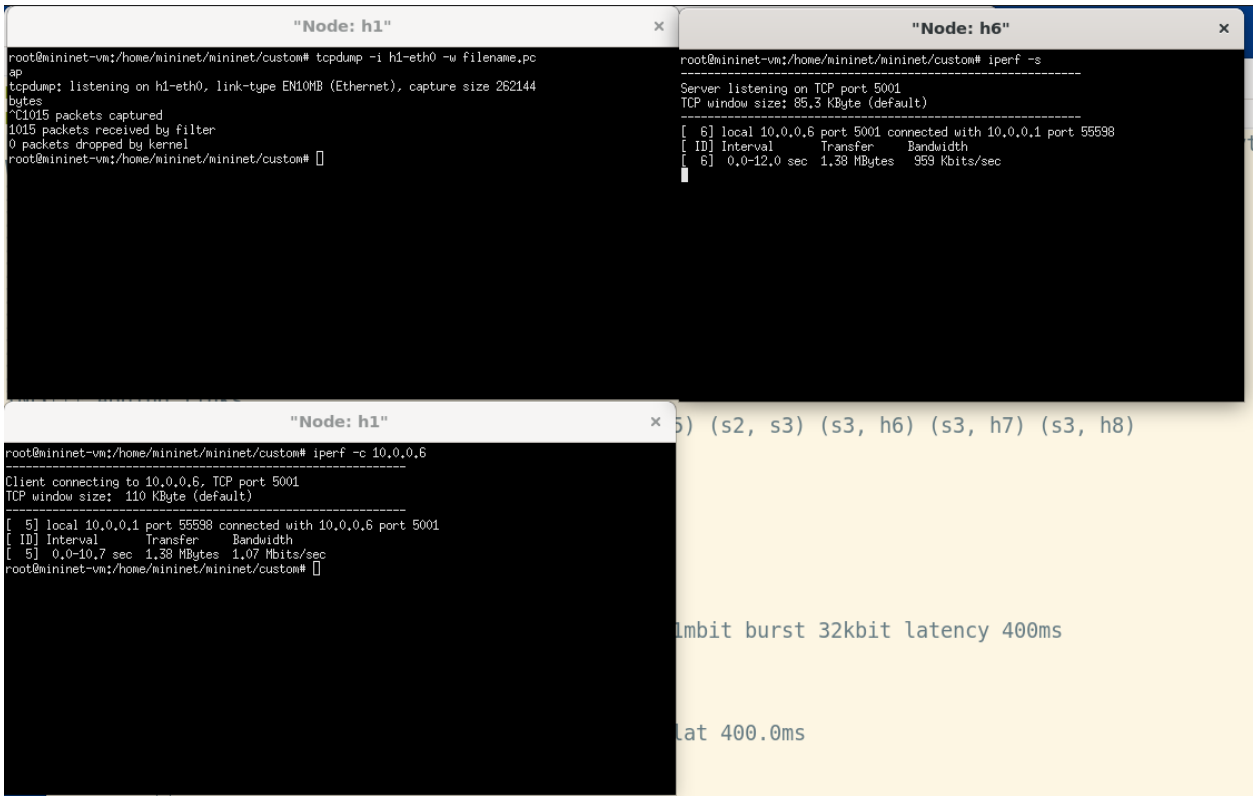
tcpdump -i h1-eth0 -w filename.pcap

In h6 window :

iperf -s

In other h1 window :

iperf -c 10.0.0.6


"Node: h1"
root@mininet-vm:/home/mininet/mininet/custom# tcpdump -i h1-eth0 -w filename.pcap
tcpdump: listening on h1-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C1015 packets captured
1015 packets received by filter
0 packets dropped by kernel
root@mininet-vm:/home/mininet/mininet/custom#
"Node: h6"
root@mininet-vm:/home/mininet/mininet/custom# iperf -s
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
[6] local 10.0.0.6 port 5001 connected with 10.0.0.1 port 55598
[ID] Interval Transfer Bandwidth
[6] 0.0-12.0 sec 1.38 MBytes 959 Kbits/sec
"Node: h1"
root@mininet-vm:/home/mininet/mininet/custom# iperf -c 10.0.0.6
Client connecting to 10.0.0.6, TCP port 5001
TCP window size: 110 KByte (default)
[5] local 10.0.0.1 port 55598 connected with 10.0.0.6 port 5001
[ID] Interval Transfer Bandwidth
[5] 0.0-10.7 sec 1.38 MBytes 1.07 Mbits/sec
root@mininet-vm:/home/mininet/mininet/custom#
5) (s2, s3) (s3, h6) (s3, h7) (s3, h8)
1mbit burst 32kbit latency 400ms
lat 400.0ms

(d)

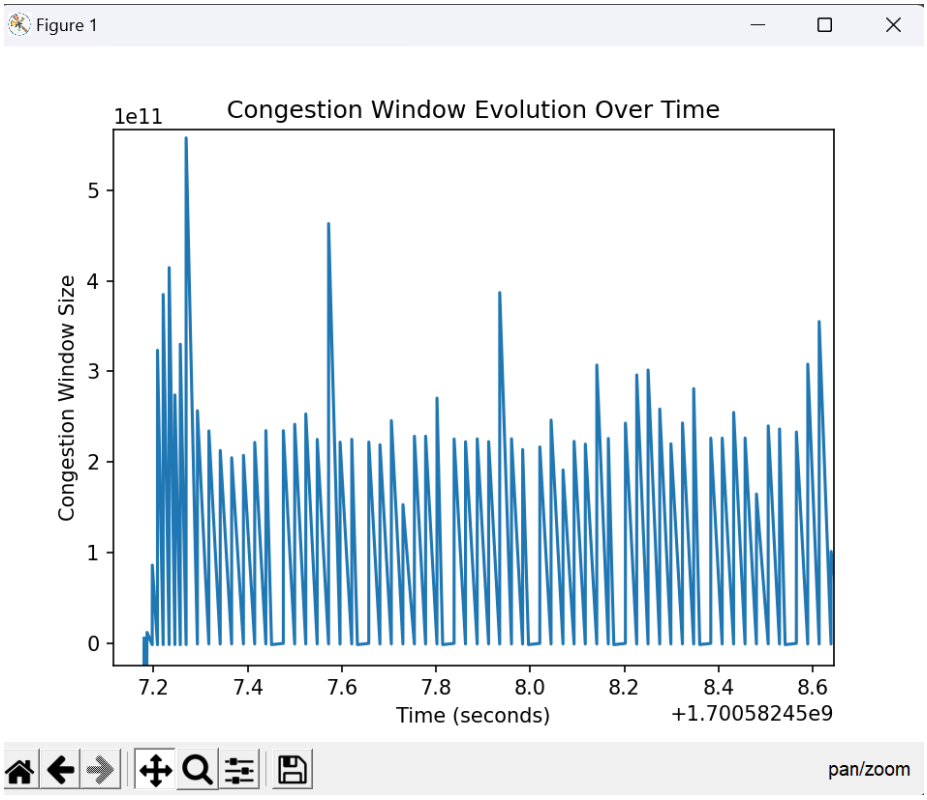


Figure 1
Congestion Window Evolution Over Time
1e11
Congestion Window Size
Time (seconds)
+1.70058245e9
pan/zoom

This pattern resembles a sawtooth shape on a graph. The congestion window represents the number of unacknowledged packets that a sender can have in transit at any given time.

The sawtooth behavior is a fundamental part of TCP's congestion control algorithm. It allows TCP to adapt dynamically to network conditions, finding an optimal balance between utilizing available bandwidth and avoiding congestion.

This dynamic adjustment helps TCP maintain network stability and efficiency, making it a reliable protocol for data transmission over diverse and changing network environments.