

Lab 3

[70 points] iperf : This should succeed. -40 points: iperf fails The remaining 30 points will be awarded depending on the quality of the explanation given.

```

mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['32.2 Gbits/sec', '32.2 Gbits/sec']
mininet>
g file /home/mininet/pox/pox/misc/lab3controller.py...
/home/mininet/pox/pox/misc/lab3controller.py
g file /home/mininet/pox/pox/misc/lab3controller.py...
/home/mininet/pox/pox/misc/lab3controller.py

```

Iperf is utilized so as to test network performance so we can then measure and tune it. In this case we run i perf between h1 and h4 so as to test the network speed against said nodes. We can see this as 32.2 gbts per second. This is the amount of data per second when testing bandwidth between h1 and h4.

[30 points] pingall : This should only work between h1 and h4, since ICMP traffic should be blocked between the rest of the hosts. -20 points: ping succeeds between any hosts other than h1 and h4 The remaining 10 points will be awarded depending on the quality of the explanation given.

```

mininet> pingall
*** Ping: testing ping reachability
h1 -> X X h4
h2 -> X X X
h3 -> X X X
h4 -> h1 X X
*** Results: 83% dropped (2/12 received)
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h4
exit
^C
Interrupt
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X h4
h2 -> X X X
h3 -> X X X
h4 -> h1 X X
*** Results: 83% dropped (2/12 received)
mininet>

```

Basically what is done here is the firewall in it of itself. We see the pings being dropped which ascertains to the rules that were provided in the document at hand. We follow the guidelines and only allows packets via h1 -> h4 and h4 -> to be received. This is done by utilizing and if

statement in the icmp segment to determine whether the incoming and outgoing ip address match the one that is specified. The rest of the packets that dont follow the given rules are weeded out such as IPV4 packets. We then allow for arp and tcp packets if they are to come down the stream of data.

[70 points] dpctl dump-flows : This should show a few entries. These are the entries that you installed into the switch with of_flow_mod. You'll need to do this within the timeout you specified in your of_flow_mod for the entries to show up! -40 points: no flows shown.

```
*** Results: 83% dropped (2/12 received)
mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
mininet> █
```

Here I only have one entry because I did not quite have enough time to finish. However we can see that I was able to get at least one flow working. My flow mod however was not in the correct synchronization that I would have liked in reference to the timer

I utilized the pox documentation through this assignment as well as a lot of help from the ta's in the lab. They basically showed me how to do this assignment. .