Controller Logic Explained

For this controller, the basic idea of how it functions is that any packet that is sent must be checked to ensure proper transmission protocol and that the untrusted host can't communicate with the rest of the system. To do this, we simply developed our flow table by implementing logic checks for ICMP, TCP, and IPv4, while leaving ARP.

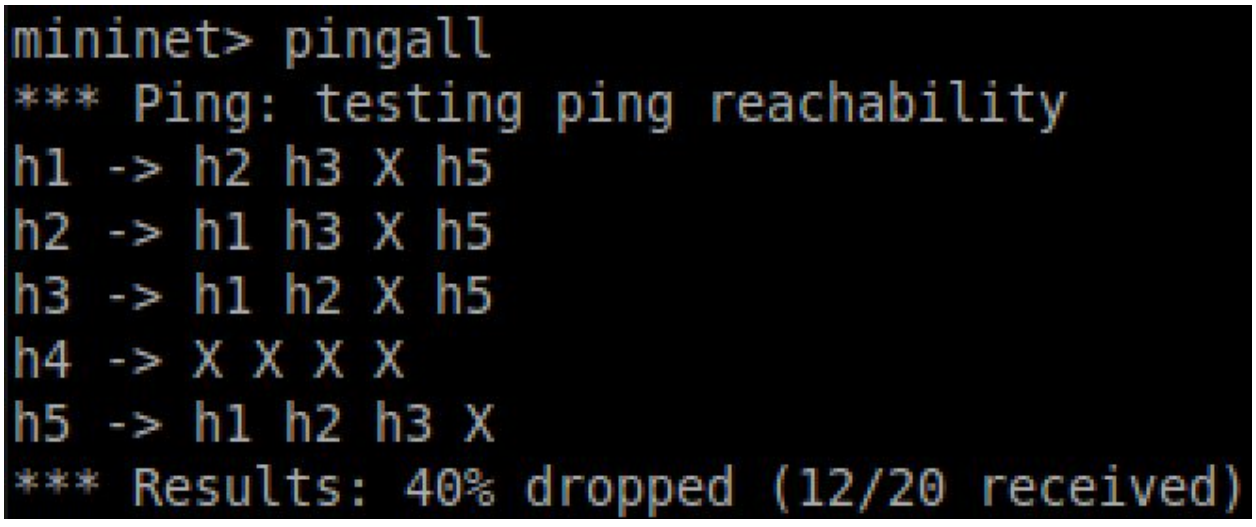
To do this, I split up how my code functions into segments based off of each switch. Of the switches, the topology makes it so that the Core Switch (Switch 4) is where all traffic will pass through. To take advantage of this, I did all my protocol checks within Switch 4 while allowing traffic to pass normally from the hosts through the other switches.

In my first if statement, I nested other if statements to confirm the destination IPs and ensured that they were being sent using ICMP and IPv4. In my second if statement, I nested other if statements to confirm the destination IPs and ensured that they were being sent using TCP and IPv4. And finally I checked for ARP and ensured that packets were being sent.

Once I passed switch 4, the other switches (1, 2, 3, 5) simply confirmed the port number on which the packet would be sent or received and ensured that was the behavior that was occuring. Upon confirmation, it would forward the packet to the correct outflow port on their respective switches.

Test Evidence

Pingall test



h4 is unable to communicate with other hosts

Net test

```
mininet> net
h1 h1-eth0:s1-eth8
h2 h2-eth0:s2-eth8
h3 h3-eth0:s3-eth8
h4 h4-eth0:s4-eth8
h5 h5-eth0:s5-eth8
s1 lo:  s1-eth3:s4-eth1 s1-eth8:h1-eth0
s2 lo:  s2-eth3:s4-eth2 s2-eth8:h2-eth0
s3 lo:  s3-eth3:s4-eth3 s3-eth8:h3-eth0
s4 lo:  s4-eth1:s1-eth3 s4-eth2:s2-eth3 s4-eth3:s3-eth3 s4-eth4:s5-eth3 s4-eth8:
h4-eth0
s5 lo:  s5-eth3:s4-eth4 s5-eth8:h5-eth0
c0
```

Iperf test between h1 and [h2, h3, h5, h4]

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['31.8 Gbits/sec', '31.8 Gbits/sec']
mininet> iperf h1 h3
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['22.8 Gbits/sec', '22.8 Gbits/sec']
mininet> iperf h1 h5
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['36.0 Gbits/sec', '36.0 Gbits/sec']
mininet> iperf h1 h4
*** Iperf: testing TCP bandwidth between h1 and h4
^C
Interrupt
```