Title: IMPLEMENTATION AND PERFORMANCE ANALYSIS OF SDN FIREWALL

Overview of firewall implementation –

I specified the rules for the Firewall by writing down Layer-2 addresses of hosts. I have then put the rules as rows in a list so that I can iterate over each rule separately later. Next created SDNFirewall class in which the actual controller is going to be accessing and checking flows and modifying flow tables accordingly. Whenever a host tries to reach other host over switch, it will first iterate over each rule in list. Here I created an OpenFlow flow_mod message using of.ofp_flow_mod() and set its match field to our blocking rule.

Concepts implemented –

- **POX Controller [0.3.0]-**

Pox is an open-source development platform for Python-based software-defined networking control appications

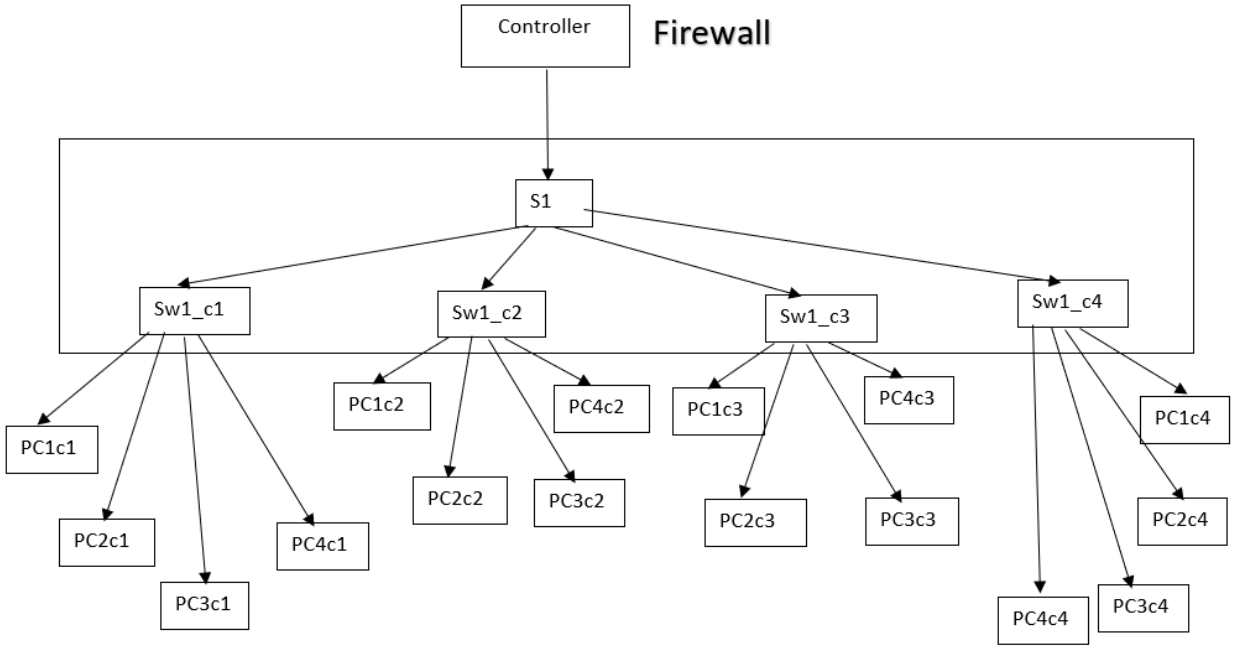
- **OpenFlow switch [Version 1.0]-**

Using OpenFlow protocol and related tools, you can program switches to do almost anything you want with the frames that enter them. OpenFlow makes emulator like Mininet much more useful, since network system designs, including custom packet forwarding with OpenFlow.

- **Simulator Used [Version 2.2.1]-**

Mininet is used to build the topology and run test cases.

Topology Used –



Procedure-

Part 1:

- Install mininet and Pox on system
- Run the python script to build topology

  Command –

  sudo mn --custom topo1.py --topo mytopo
- Created Flow between switches:
- Command -

  sh ovs-ofctl add-flow SW1c1 "prority=0, action=normal"

  sh ovs-ofctl dump-flows SW1c1
- Tested the topology using pingall to find the reachability

Part 2 –

- Created a myfirewall.py which will block the desired host among each other
- Run generate_rules.py to generate rule tables for blocked flow Layer-2 address
- Run following command enabling spanning tree to avoid loops-

Command-

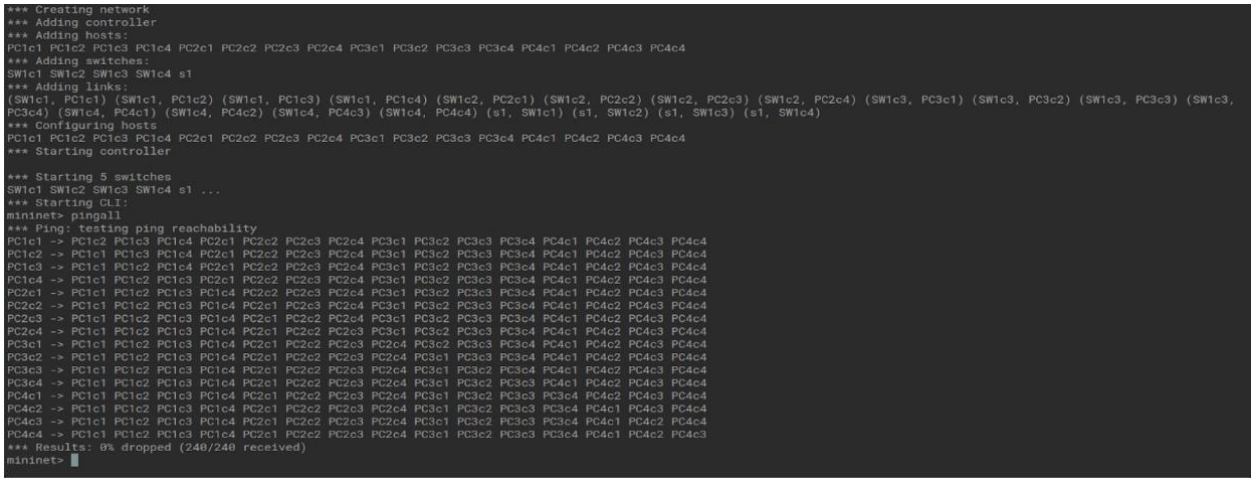
./pox.py forwarding.l2_learning openflow.discovery openflow.spanning_tree --no-flood --hold-down pox.misc.myfirewall

- In another terminal run the topology
- Test using pingall

Result –

Part -1

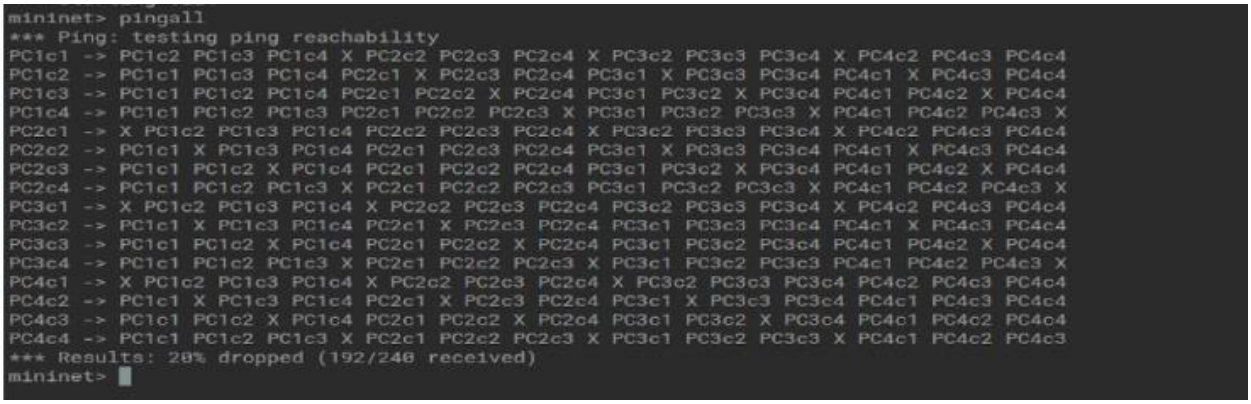
After adding flow amongst all switches. 0% packet dropped is observed, i.e. 240/240 packets were transferred successfully.



Result 0% dropped 240/240

Part -2

After adding the Firewall to topology 20% packed dropped is observed i.e. 192/240 packets transferred successfully.



Result 20% dropped 192/240