Indian Institute of Information Technology, Allahabad

Course: Computer Network

Batch: B.Tech (IT) Section B

Lab Assignment #1

Deadline: 26-04-2020

QUESTION-1

1. Build a Data Center Topology as shown below using Mininet CLI (DatacenterBasicTopo). In this topology, we have four brackets, each with four forces and one top-of-rack (ToR) switch. These ToR switches are connected to a central root switch.

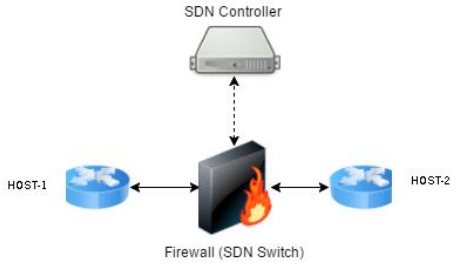
```
===========================[ s1 ]================================
        |               |                    |                |
  [ Sw1_c1]        [ Sw1_c2 ]           [ Sw1_c3 ]       [ Sw1_c4 ]
   [ PC1c1]-        [ PC1c2 ]-           [ PC1c3 ]-       [ PC1c4 ]-
   [ PC2c1]-        [ PC2c2 ]-           [ PC2c3 ]-       [ PC2c4 ]-
   [ PC3c1]-        [ PC3c2 ]-           [ PC3c3 ]-       [ PC3c4 ]-
   [ PC4c1]-        [ PC4c2 ]-           [ PC4c3 ]-       [ PC4c4 ]-
```

```
*** Starting 5 switches
s1 s1r1 s1r2 s1r3 s1r4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1r1 -> h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h1r2 -> h1r1 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h1r3 -> h1r1 h1r2 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h1r4 -> h1r1 h1r2 h1r3 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h2r1 -> h1r1 h1r2 h1r3 h1r4 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h2r2 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r3 h2r4 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h2r3 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r4 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h2r4 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h3r1 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h3r1 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r2 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h3r2 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r3 h3r4 h4r
1 h4r2 h4r3 h4r4
h3r3 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r4 h4r
1 h4r2 h4r3 h4r4
h3r4 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h4r
1 h4r2 h4r3 h4r4
h4r1 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r
4 h4r2 h4r3 h4r4
h4r2 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r
4 h4r1 h4r3 h4r4
h4r3 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r
4 h4r1 h4r2 h4r4
h4r4 -> h1r1 h1r2 h1r3 h1r4 h2r1 h2r2 h2r3 h2r4 h3r1 h3r2 h3r3 h3r
4 h4r1 h4r2 h4r3
*** Results: 0% dropped (240/240 received)
mininet>
```

Fig1: Output window should Prints these lines.

Note: Please change the name of switch and host as given above.

2. Implement the SDN firewall in the network to obstruct traffic coming its way and filters it according to some *rules*. A general firewall usually Protect the system from internet.
   Hint :
   [For an SDN based Firewall, an OpenFlow controller is required to filter traffic between hosts according to some rules and accordingly let it pass through or not]. [use POX controller to establish our required *policies* or *rules* and filter traffic between hosts using the switches.]



- PC1c1, PC2c1, PC3c1 and PC4c1 should mutually blocked
- PC1c2, PC2c2, PC3c2 and PC4c2 should mutually blocked
- PC1c3, PC2c3, PC3c3 and PC4c3 should mutually blocked
- PC1c4, PC2c4, PC3c4 and PC4c4 should mutually blocked

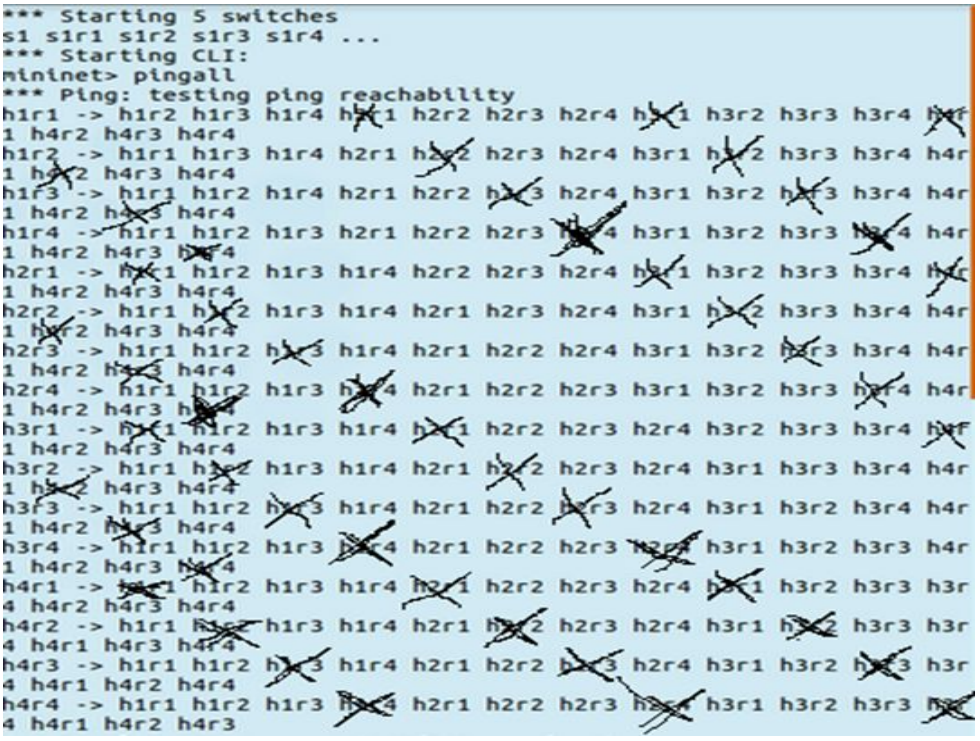


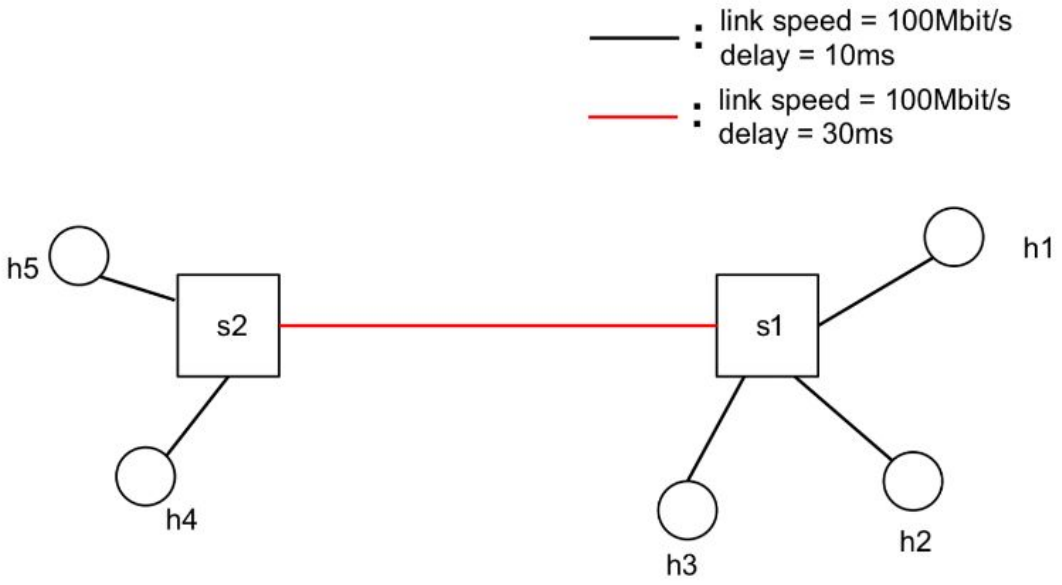
Fig2: Output should not display X pc in the above.

Results 19% dropped (192/240)

QUESTION 2:

TCP throughput & fairness

In this exercise we will study TCP's throughput and fairness characteristics with Mininet. Consider the topology shown in the figure below.



All links are 100 Mbit/s and have a one-way propagation delay of 10 ms, except for the link between S1 and S2, which has a 30 ms propagation delay. The switches have 1000 packets of buffer at each port, and use PIE active queue management with a target delay of 20 ms.

The goal of this exercise is to learn how the RTT and number of bottleneck links impacts throughput and fairness of TCP flows.

Tasks

Within Mininet, create the topology shown above, and enable PIE with a target delay of 20 ms on the switch interfaces. Then do the following:

Scenario 1:

- Start 10 long lived TCP flows sending data from h2 to h1, and similarly 10 long lived TCP flows from h5 to h1.
- Start back-to-back ping trains from h2 to h1, and h5 to h1. Record the RTTs with ping 10 times a second.
- Measure the following:
    - The average throughput for h2->h1 and h5->h1 flows. For each group, measure the aggregate throughput of the 10 flows.

- The average RTT for h2->h1 and h5->h1 flows.

Scenario 2:

- Now start 10 long lived TCP flows from h4 to h3.
- Repeat the above measurements (average throughput and RTT) for the three groups of flows: h2->h1, h5->h1, and h4->h3.

The starter code sets up the topology and configures PIE. You have to write code to generate the traffic (with Iperf) and do the RTT measurements (with ping).

Starter code

git clone https://github.com/hongzimao/6.829_lab1?files=1
or https://github.mit.edu/addanki/6.829-lab1.git

File	Purpose
tcpfairness.py	Creates the topology and sets up PIE.

Answer the following questions below. Remember to keep answers brief.

1. In Scenario 1, which of the following statements is more accurate:
   - The congestion windows of h2->h1 flows are larger than h5->h1 flows.
   - The congestion windows of h2->h1 flows are smaller than h5->h1 flows.
   - The congestion windows of h2->h1 and h5->h1 flows are roughly the same.
2. In Scenario 2, which of the two bottleneck links (S2->S1 or S1->h1) do you expect has the larger drop rate? Briefly explain why