Software Defined Load Balancer with POX Controller

CS 6301.502 | Fall 2018

Priyank Shah

Manan Lakhani

Karan Motani

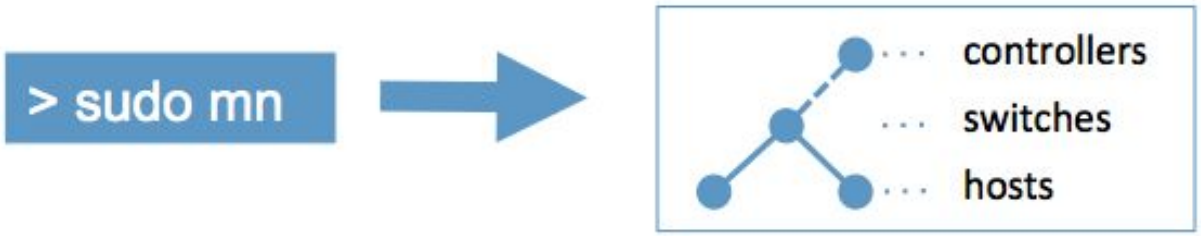
Ankita Patil

Kripanshu Bhargava

Objective

To develop a software defined load balancer with POX controller which makes flow balancing decisions based on the statistics such as bandwidth and latency, gathered from interfaces of switches and nodes in the network

Technology



1. Mininet

To simulate the network

Technology

2. POX Controller

Open source development platform for Python based software defined SDN control application to check the configuration of customized networking application

Technology

3. OpenDayLight

To visually represent and simulate the network topologies

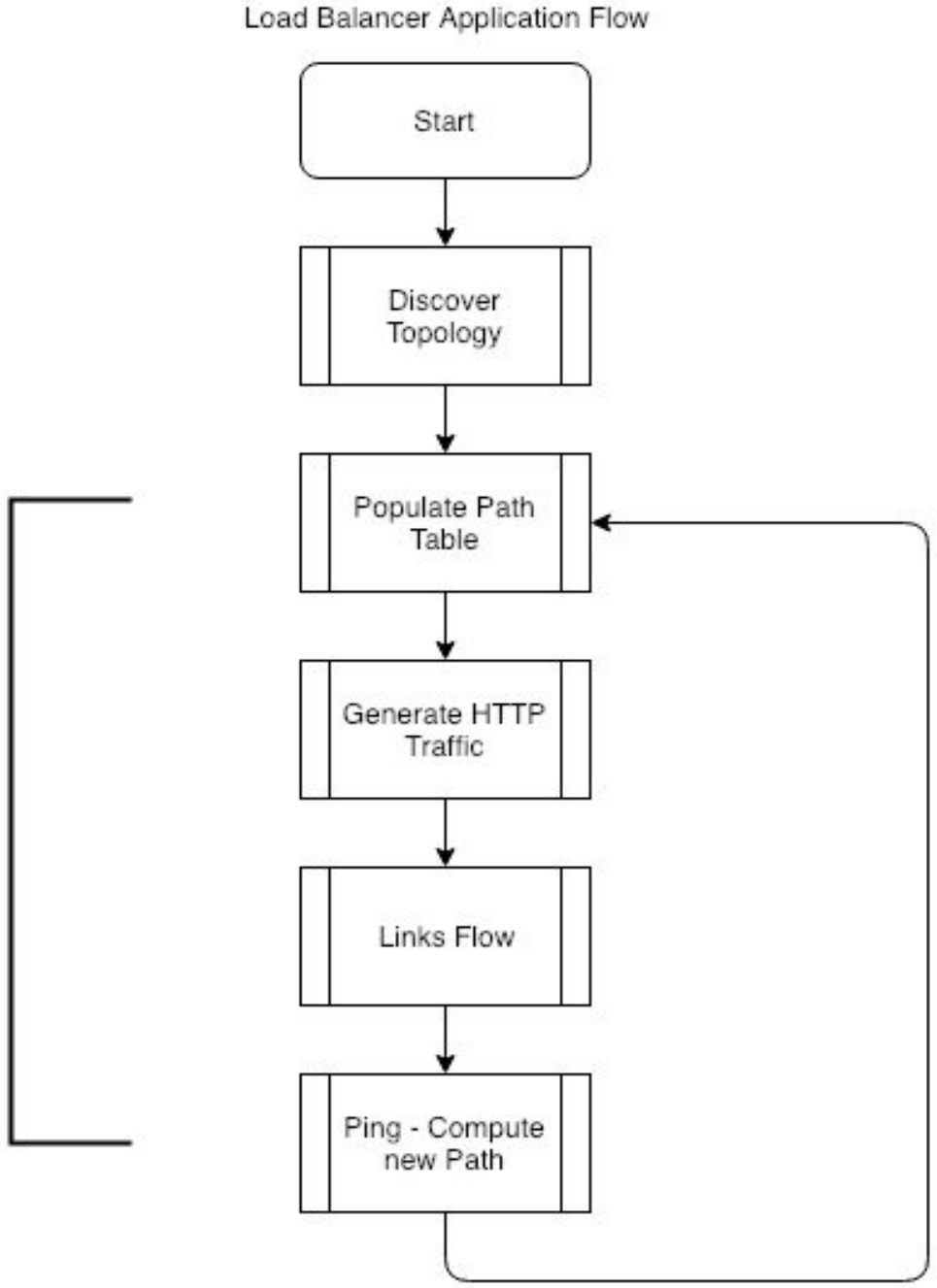
Network Topology

In mininet, a network topology is created which consists of three switches and three hosts. The hosts are labelled as h1, h2, h3 and switches are labelled as S1, S2, S3.

Application Flow

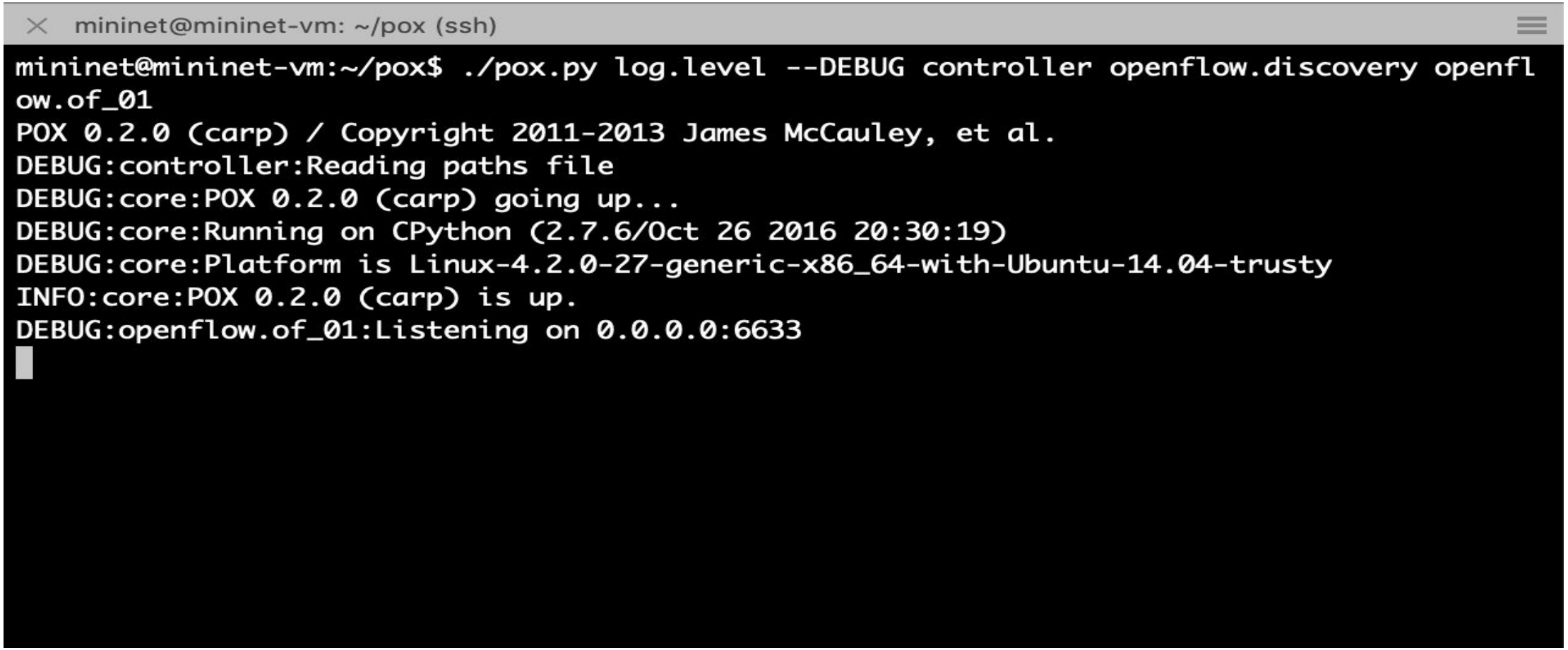
loop

This loop runs every 5 minutes in order to populate new tables for new links. loop is triggered using ping command from python file on mininet

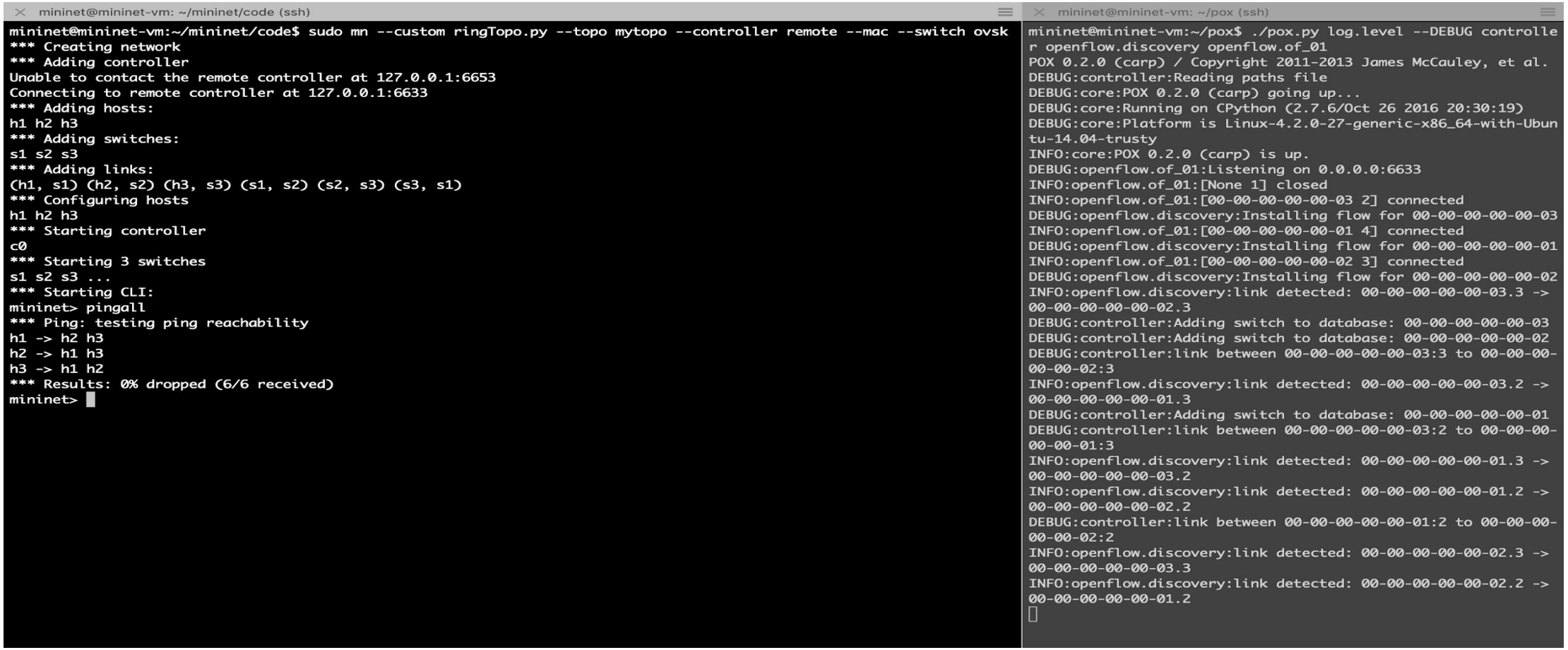


Screenshots

POX Controller Setup



Creating Custom Topology (RING)



Starting HTTP Server

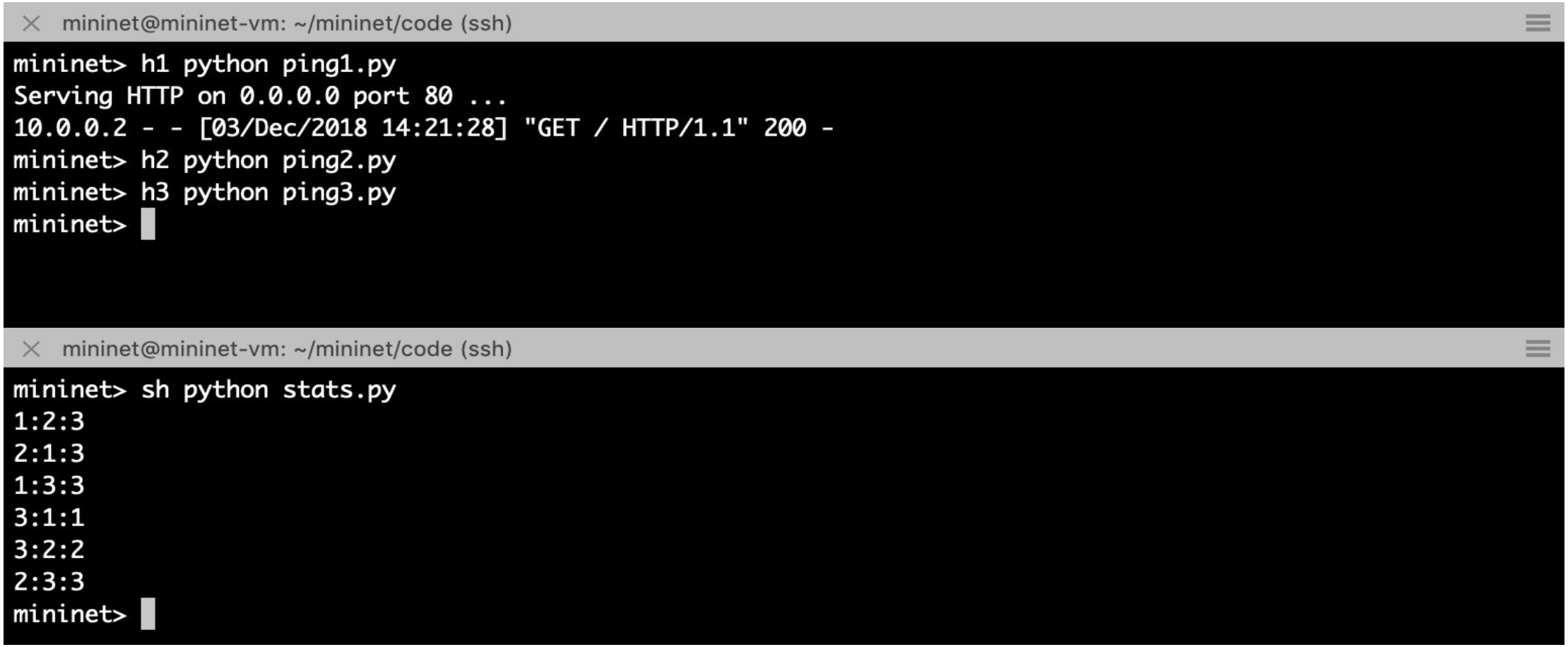
```
mininet@mininet-vm: ~/mininet/code (ssh)
mininet> h1 python -m SimpleHTTPServer 80 &
mininet> h2 wget -O - h1
--2018-12-03 14:21:28--  http://10.0.0.1/
Connecting to 10.0.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 406 [text/html]
Saving to: ‘STDOUT’

 0% [                                       ] 0           --.-K/s              <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href="ping1.py">ping1.py</a>
<li><a href="ping2.py">ping2.py</a>
<li><a href="ping3.py">ping3.py</a>
<li><a href="ringTopo.py">ringTopo.py</a>
<li><a href="starTopo.py">starTopo.py</a>
<li><a href="stats.py">stats.py</a>
</ul>
<hr>
</body>
</html>
100%[======================================>] 406         --.-K/s   in 0s

2018-12-03 14:21:28 (62.2 MB/s) - written to stdout [406/406]

mininet>
```

Generating Traffic and Collecting Statistics



Statistics

To get the statistics from the switches, a set of python scripts are written which will ping all the nodes from a particular node. After pinging the nodes, average latency between each node is collected. Depending upon the average latency, an alternative path with lowest overall latency is selected. This statistics are used by controller to modify flows.

import urllib, os

This script is used for pinging the client 2 and 3 from 1

- os.system("ping -c 5 10.0.0.2 | tail -1| awk '{print $4}' | cut -d '/' -f 2 >> ping_stat_1_2")
- os.system("ping -c 5 10.0.0.3 | tail -1| awk '{print $4}' | cut -d '/' -f 2 >> ping_stat_1_3")

Challenges

01

While setting up the whole project from a closed network

02

While setting up OpenDayLight

03

While setting up POX Controller

Conclusion

Implemented software defined POX controller which makes flow balancing decisions based on statistics such as bandwidth and latency gathered from the interfaces of nodes and switches in the network which is simulated in mininet.

References

- http://mininet.org/walkthrough/
- https://www.opendaylight.org/
- https://openflow.stanford.edu/display/ONL/POX+Wiki.html
- http://networkstatic.net/installing-mininet-opendaylight-open-vswitch/
- https://searchnetworking.techtarget.com/definition/Mininet

Thank you!