Project in Network Management

Sdi1700197- Thodoris Maximilianos Chytis
Sdi1700216- Stelios Xenofontos
Sdi1700086- Dimitris Mesisklis

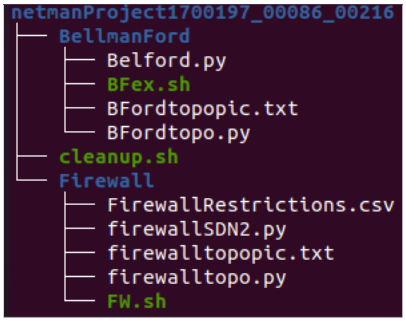
Requirements:

- Mininet-wifi (https://github.com/intrig-unicamp/mininet-wifi)
- OpenDayLight Oxygen (https://john.soban.ski/how-to-install-

opendaylight-as-a-service-on-ubuntu.html).

- Pox (https://noxrepo.github.io/pox-doc/html/)

The “netmanProject1700197_00086_00216” folder contains the project.

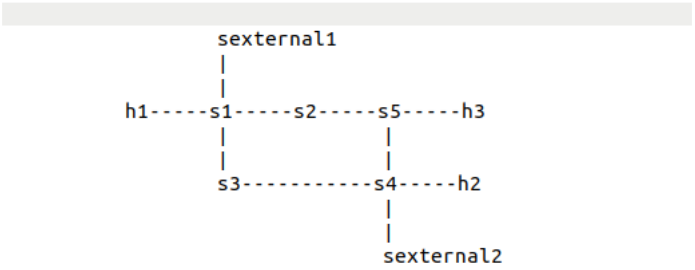


This project successfully illustrates two common software defined networking applications. The first application deals with the need for optimal routing between hosts via switches/access points using dynamic programming and the shortest path Bellman Ford algorithm.
The second application is a firewall configuration that allows or blocks package streams between certain host. Both applications are Python scripts that work with the Pox controller and library, using listeners, events and event handlers to deal with the networks’ states regarding each application.

The topologies for the virtual networks are built using mininet/mininet-wifi and connect to the remote SDN Pox Controller to be monitored.

The topologies support connectivity with another remote controller that is called OpenDayLight, for a Graphic depiction of the virtual Network.

1. **Bellman Ford**



The above picture is the BFordtopo.py topology. We are trying to find the shortest path between **h1** and **h2.**
After execution in the terminal running the Pox Controller we get the

```
Using Dynamic Programming and Bellman Ford Algorithm
The shortest path from 01 to 04 is:
[00-00-00-00-00-01, 00-00-00-00-00-03, 00-00-00-00-00-04]
--------------------------------
```

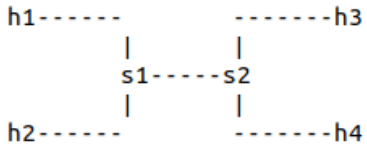
following answer:
Which is the shortest path between **h1** and **h2.**

Execution:

- **Copy** …/netmanProject1700197_00086_00216/BellmanFord/Belford.py **into** …/pox/ext
- **Copy** …/netmanProject1700197_00086_00216/BellmanFord/BFex.sh **into** …/pox
- **Run** cleanup.sh …/netmanProject1700197_00086_00216/cleanup.sh
- **Run** BFex.sh in …/pox
- **Run** cleanup.sh …/netmanProject1700197_00086_00216/cleanup.sh
- sudo python Bfordtopo.py in …/netmanProject1700197_00086_00216/BellmanFord
- pingall in mininet CLI
- Then exit in CLI
- **Run** cleanup.sh …/netmanProject1700197_00086_00216/cleanup.sh

Etc.
If you want to see the Graphic depiction of the network you can run “sudo -E karaf” in a different terminal after you run pox and before you exit mininetCLI. Then go to your browser and visit: http://localhost:8181/index.html#/topology.

2. Firewall



The above picture is the firewalltopo.py topology. We are trying to apply rules for connectivity between **h1 , h2 , h3** and **h4** via **s1** and **s2.** In detail packet flows from column **B** to column **C** are blocked.

	A	B	C
1	1	10.0.0.1	10.0.0.3
2	2	10.0.0.2	10.0.0.4
3	3	10.0.0.1	10.0.0.4
4			

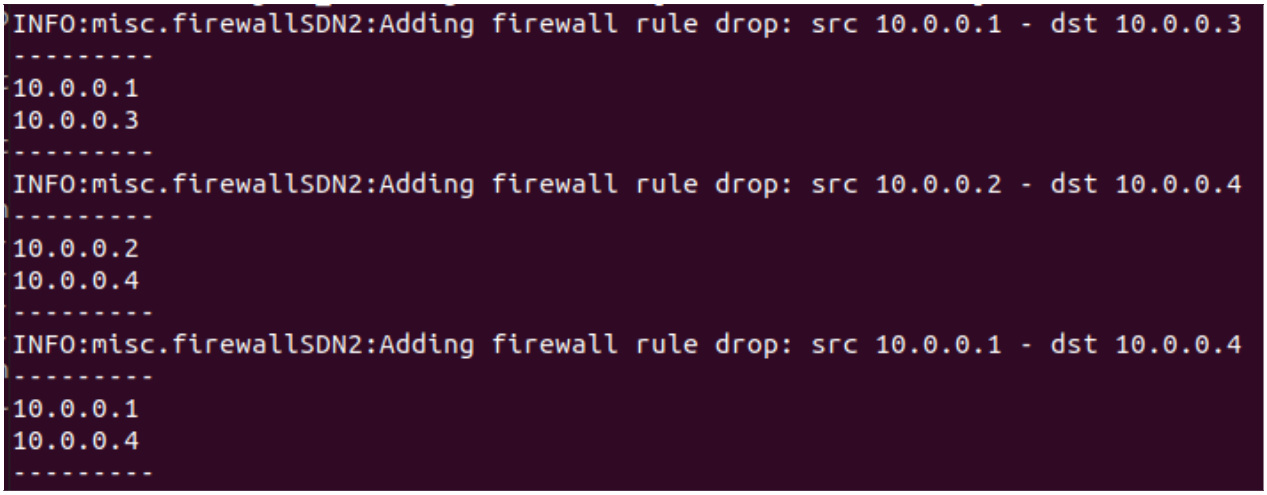
The above picture represents the csv file that we use to apply the rules in firewallSDN2.py script.

Execution:

- **Copy** .../netmanProject1700197_00086_00216/Firewall/firewallSDN2.py **into** .../pox/pox/misc
- **Copy** .../netmanProject1700197_00086_00216/ Firewall/FW.sh **into** .../pox
- **Run** cleanup.sh .../netmanProject1700197_00086_00216/cleanup.sh
- **Run** FW.sh in .../pox
- **Run** cleanup.sh .../netmanProject1700197_00086_00216/cleanup.sh
- sudo python firewalltopo.py in .../netmanProject1700197_00086_00216/Firewall
- Then exit in CLI
- **Run** cleanup.sh .../netmanProject1700197_00086_00216/BellmanFord/cleanup.sh

```
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 X X
h2 -> h1 h3 X
h3 -> X h2 h4
h4 -> X X h3
*** Results: 50% dropped (6/12 received)
```

After execution of the firewalltopo.py you get these results:



Etc.
If you want to see the Graphic depiction of the network you can run “sudo -E karaf” in a different terminal after you run pox and before you exit mininetCLI. Then go to your browser and visit: http://localhost:8181/index.html#/topology.

This project shows the advantages of software defined networking as it is evident that with the use of a Remote Controller we can monitor, control and modify network connectivity and routing as we did with applications written for the Pox Controller over mininet-wifi networks.