# Software Defined Networking

## 3rd Lab project

2nd Semester 2010/2021

# SDN controller

## ONOS controller

Eliana Neuza da Silva

Fernando Mira da Silva

April 2121

# 1. Introduction

Before starting the lab work it is important to understand certain concepts that will be briefly addressed below.

## 1.1 ONOS controller

As explained in previous lab class, SDN is a network architecture introduces the decoupling of the data plane from the control plane. The control plane is comprised of a controller, that introduces network programmability allowing it to perform easy network configurations, management, monitoring or network reformulations (without reprogramming the physical network elements). An SDN controller configures network devices (routers, switches, ...) to achieve the best paths and modify traffic flows according to network, administration and applications needs.

The same as OpenFlow from the previous lab, ONOS[1] (Open Network Operating System) is an SDN controller allowing customization of routing, management and monitoring of services. It is a distributed system running on multiple servers providing fault tolerance, in case of server failure. ONOS has multiple core systems, Figure 1, it provides APIs, abstractions, resource allocation, permissions and user-facing software such as a CLI, a GUI, and system applications. ONOS also supports clusters of multiple controllers that share state amongst themselves.

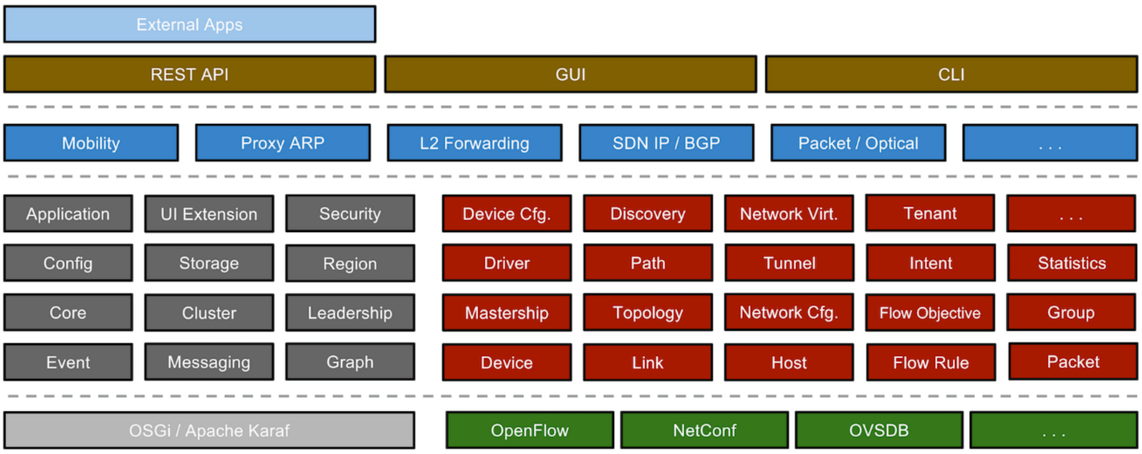

*Figure 1 – ONOS core subsystems.*

# 2. Goals

The third lab work will have the main goal of using ONOS controller with Mininet to emulate SDN networks.

# 3. Evaluation

To grade your work for the lab class #3 you will need to deliver a report, through the Fenix platform, Sunday, May 2nd. All reports should have the group number as well as the names and numbers of each member.

The report should respond to the questions throughout the first exercise and explain all the steps and queries that you used to set up and test the topology for the second exercise.

[1] opennetworking.org/onos

# 4. Setup and Testing

In this laboratory exercise, you will need the following:

- A Linux distribution with a graphical environment (Ubuntu 18.04 Desktop or any of its flavours are recommended, with `root` access or a user with `sudo` privileges)
- Mininet;
- Wireshark;
- Java, default JDK[2];
- ONOS controller[3].

## 4.1 Connecting to the controller

1. Start the ONOS controller:

```
$ ./opt/onos/bin/onos-service start
```

2. Make sure the ONOS GUI is accessible:

https://wiki.onosproject.org/display/ONOS/Accessing+the+CLI+and+GUI

3. Run Mininet with the following start up configuration:

```
$ sudo mn --mac --controller=remote --topo=linear,2
```

4. Check the connected devices in the controller's GUI. Explain in your report what happened and why. Use Wireshark to see what is happening by capturing packets from the localhost interface of the virtual machine.

5. Enable OpenFlow Protocol. Using the ONOS GUI: In the top left corner menu choose "Applications" and then activate the "OpenFlow Provider Suite".

6. Run Mininet again with the same start up configuration and notice that the switches now appear in the ONOS GUI.

7. Check for connectivity:

```
mininet> h1 ping -c5 h2
```

Explain again in your report what happened and why. Use Wireshark to see what is happening by capturing packets from the localhost interface of the virtual machine.

8. Activate forwarding in the ONOS GUI. In the same "Applications" menu as before, activate the "Reactive Forwarding".

8. Check again for connectivity:

```
mininet> h1 ping -c5 h2
```

Provide screen shots of Mininet and your Wireshark captures to show connectivity.

[2] `sudo apt install default-jdk`

[3] wiki.onosproject.org/display/ONOS/Installing+on+a+single+machine

## 4.2 Flow manipulation using a REST API

Using the ONOS REST API documentation[4] implement the flow rules from the lab work 2, with the topology in Figure 2. The red link between S1 and S3 has a bandwidth of 100 Mb/s and 10 ms of delay and the other two links have a bandwidth of 1000 Mb/s. Traffic coming from h3 must always go through the red link and traffic coming from h4 must always go through the other.

Use Mininet to implement the topology, which will be identical to the one created for the previous lab, but now you will use ONOS as a remote controller.

The flow rules will be implemented using ONOS CORE REST API[4] that you can use through the following link: `http://localhost:8181/onos/v1/docs/`.

Your report should include the following:

- Detail the steps to assemble the topology and implement the flow rules with ONOS
    - Present al the requests/queries made to ONOS API to implement the flow rules;
- Demonstrate that the traffic paths are correct using `iperf` tests;
- Explain what happens when you delete a link that had a static flow rule (i.e. implemented by you from the API). Is the traffic dropped? Do you lose connectivity? Does the controller automatically redirect traffic? Does the controller implement a new flow rule with higher priority? Does the static flow rule become inactive?

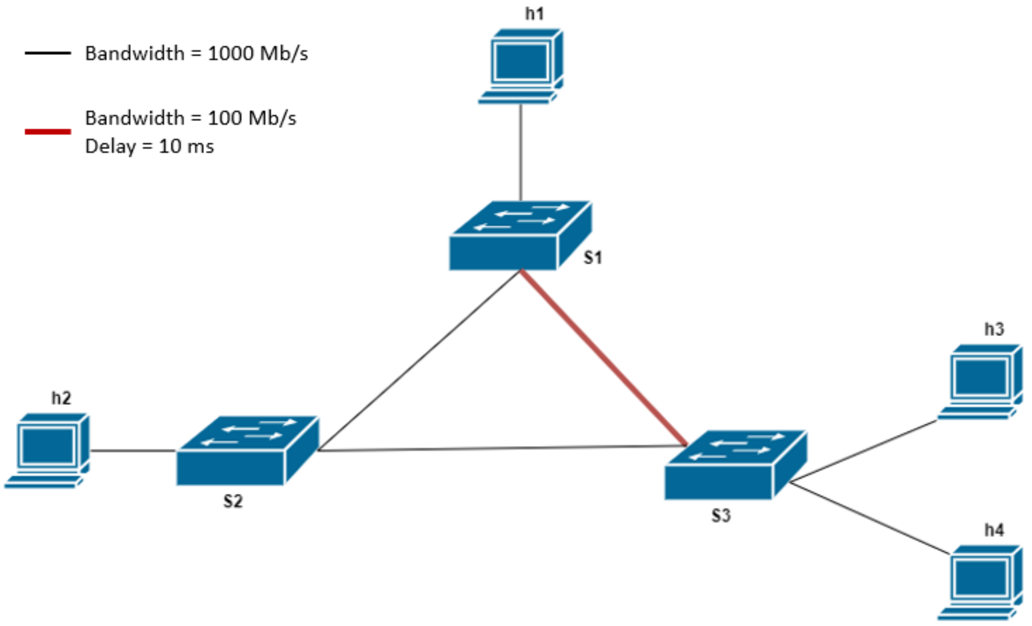

*Figure 2 – Custom topology from Lab work 2.*

[4] wiki.onosproject.org/display/ONOS/Appendix+B%3A+REST+API