Rede Definidas por Software
TP2 - Network Managment

Version: 1

Departamento de Informática
Universidade do Minho
April 2022

João Fernandes Pereira

Goal

The goal of this work is to refine your knowledge of the OpenFlow protocol and SDN architecture. To that end, this assignment focus in the development of a routing application and a firewall on top of the SDN controller.

Report & Submission

All the develop code must be submitted. The report should consist in a small pdf document.

- Introduction
- Exercise 1
  - Strategy
  - Implementation
  - Tests
- Exercise 2
  - Strategy
  - Implementation
  - Tests
- Conclusion

The submission is made via blackboard with a zip file containing all the files.
RDS_TP2_GroupX.zip

Required Software

- Mininet
- Ryu (python) or FloodLight (java)
- HTTP server
- Wireshark

Overview - Topology

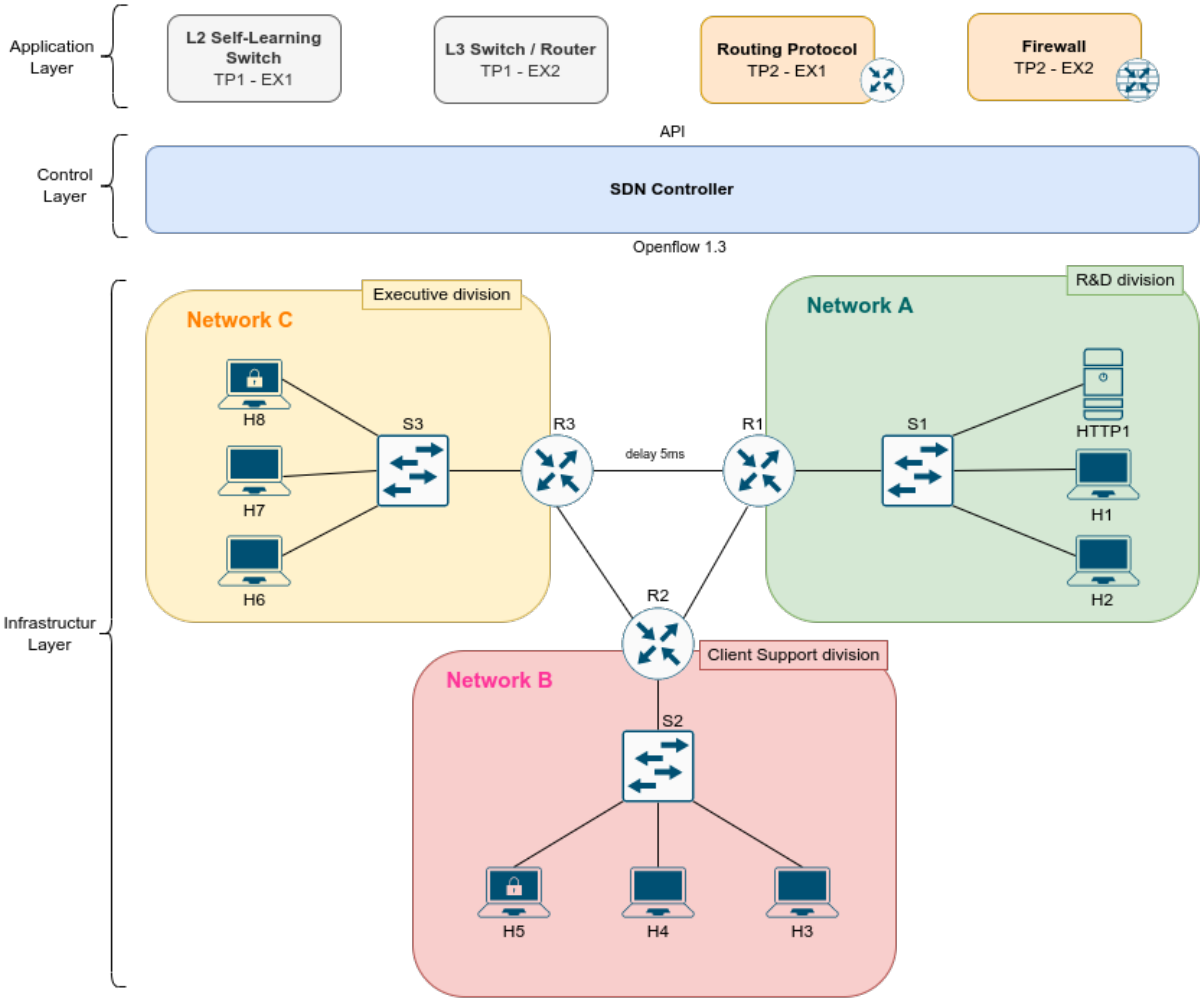

Figure 1: Example of topology for this practical assignment.

Figure 1 presents an example of a possible topology for this home assignment. You can create your topology, but it must meet the flowing requirements:

- at least three distinct networks
- at least three routers
- one HTTP server
- at least two hosts per network
- at least one link between routers must have a 5ms delay

Exercise 1 - Routing Protocol

The purpose of routing protocols is to tell other routers which paths are available so that the other routers can make a good forwarding decision. In SDN, the controller has a complete view of the network – it already knows all the paths. So the controller calculates the best path for a particular kind of data and then uses a protocol like OpenFlow to tell the network devices how to forward data.
Openflow is just a protocol for communicating between the 'forwarding units' and a controller. The controller itself determines what to do with packets, and it can do this in any way it likes. So you can implement most of the current routing protocols in an OpenFlow controller.
Here, the objective is to create a scenario where every Network (A, B and C) can communicate with each other even in the case of link failure. For example, imagine that H6 in Network C is exchanging information with H1 in Network A. If the link between R3 and R1 fails, the exchange must carry on via R2.

Exercise 2 - Firewall

In this exercise, you will develop a simple firewall. Imagine that you are in a corporate structure with three departments, Executive, Research and Development (R&D) and Client Support. The R&D develops a product and stores the release versions into the HTTP1 server. As the name implies, this server has an HTTP API so that the CEO (H8) can track the development of the product, and the Client Support Administrator (H5) can report bugs and user suggestions. Therefore, your goal is to block all traffic between Networks (A, B and C) and only allow HTTP traffic to a specific port from H8 and H5 to HTTP1.

Firewall rules:

- HTTP traffic from H8 to port 5555[1] of HTTP1: allow
- HTTP traffic from H5 to port 5555[1] of HTTP1: allow
- Traffic within the same Network: allow
- Traffic between different Networks: block

[1] use any port number, except 80 and 8080