SDN Fundamentals & Techniques

Demo: Mininet – as a SDN emulator

1 Introduction

The main objective of this demonstration is to get familiar with Mininet as a SDN emulation environment. By doing this assignment, students will master both Mininet's CLI and Python API. Besides, students will also learn to create fast prototyping topologies.

2 Setting things up

In this course, we will be using ONOS as a SDN controller along with some dependencies: Docker, OVS, Mininet, and Linux Machine. It is noted that networking knowledge, familiarity with Linux shell bash, and python languages are considered as prerequisites.

The following gives guidelines on how to set us these dependencies.

- ONOS is officially supported by the Open Networking Foundation (ONF) and runs on Linux machines and containers, i.e., LXC, LXD, and Docker. If you do not have any of them on your machine, you can either set up a virtual machine (using VirtualBox or VMWare) or use one of the Aalto computers. We recommend using Ubuntu versions 18.04 and 20.04 as we have been able to test them and they work fine.
- All files referred to in this assignment are available on MyCourses demo1
- We recommend using ONOS as a Docker image and we provide a simple shell script to instantly get ONOS ready. Please refer to the file "Install ONOS".
- Docker is required for the installation of ONOS, please refer to the "Install Docker & OVS" file for installation guidelines. Please change the mode of the shell script to add execution, i.e., chmod +x your_file.sh.
- After successfully installing ONOS a link should appear in your terminal. To be able to access ONOS, the default ONOS credentials are – username: onos, password: rocks.
- The next step is to start ONOS applications. For this purpose, you may refer to the file "Using ONOS" and follow the instructions therein. Those are activated as this assignment assumes that the application and the controller layers are out of the scope of this demonstration.To install Mininet, students are requested to follow the provided document. This document explains the steps for getting the Mininet emulator ready for work.

3 Tasks

Task 1

- Create using the CLI interface and the OpenFlow Reference Controller:
  - A single-switch topology with 10 hosts. Note that a “single-switch topology” means one switch connected to all hosts.
  - A linear topology of 5 switches and 5 hosts.
  - A tree topology depth 3 fanout 2.
- Repeat the same operations using the remote SDN controller installed before, i.e., ONOS.

Note that for each question in Part 1 of Task 1, students are asked to provide the command used in addition to a brief explanation.

Task 2

Create using the Python API interface and the ONOS SDN Controller:

- A single-switch topology with 13 hosts. Note that a “single-switch topology” means one switch connected to all hosts.
- A linear topology of 10 switches and 10 hosts.
- A tree topology depth 3 fanout 2.

Note that you are free to mix API levels, i.e., low, middle, or high, or to use only one of them. Make sure that your code is well documented and follow python programming principles.

Task 3

Try to automate the precedent task, i.e., Task 2, to be able to create a topology given the type, i.e., tree or linear, and for each type its specifications.

4 How to submit?

The deadline to submit the solutions through MyCourses is 7 March. 2022 at 23:55. Your submission should contain answers to the above-mentioned questions as well as the code with solutions used for the assignment.

Good luck!