Network Softwarization: Technologies and Enablers
Assignment - 2
Due date: Jan 30 (Wednesday) 11:59PM

Overview

In this assignment you will write a controller application using ONOS's REST API. You can use C/C++, Java, Python or Bash (right now we have these four in the testing environment; if there is a compelling need to use any other language then please consult with the TA) for completing the assignment. The application will proactively setup a bi-directional path between a pair of hosts, monitor the network topology, and re-setup the path if any link on the existing path fails. In other words, the application behaves in a way similar to a host-based intent.

Description

The application will take exactly two command line arguments. The arguments represent a pair of host identifiers (the identifiers will be the same as the one from running hosts command in ONOS). Once the application starts, it will use ONOS's REST API to obtain the network topology, find shortest path between the specified pair of hosts and install flow rules in the switches to setup a bi-directional path between the hosts. Then, the application will periodically (every 5 second) monitor the port status of the switches along the path. If any link goes down on the path then the application will recompute a new path between the specified pair of hosts, remove old flow entries from the switches, and install new flow rules on the appropriate switches to re-establish the bii-directional path between the specified host pair. You can assume that no no forwarding application will be loaded as part of ONOS controller during testing, *i.e.*, there is no other way of setting up a path other than using your application. You can assume the following applications will be loaded during testing: **`hostprovider`**, **`lldpprovider`**, and **`openflow-base`**.

What to submit

The submission package should contain the source files and a Makefile. After running **make**, an executable (it can be a binary or a bash script containing necessary commands and with appropriate permissions to run the program. *e.g.*, for Java programs it will invoke **`java`** with the main class) named **`hostIntentApp`** should be generated. **`hostIntentApp`** will be executed in the following way:

```
$ ./hostIntentApp <h1Id> <h2Id>
```

Compress the submission package and name it <lastname_firstname_university.zip>, where university is one of **waterloo, toronto, laval, ets**. *Please note that, an automated testing script will be used. So please pay attention to the name of the executable and also do not add any named command line argument.*

Marking Scheme (Total: 100)

1. Correct path setup (without any failure)	(30)
2. Detect link failure	(30)
3. Correct path setup after failure	(20)
4. Remove old flow entries after setting up a new path	(20)