Part 3

Eman Almadhoun
Student ID: 932909951

Mininet is an emulator for computer networks and distributed systems virtually, which we can use it to do our experiments and give us excellent results. Also, it gives us the idea of how to create the topology with hundreds of nodes and how it works on our computers. Mininet has written in Python language and use its API libraries. Mininet startup a virtual system in a few second building the controller, switches, with a help of Linux operating system inside Virtual Machine by using commands and specify how many nodes or switches need to create and what the type of the topology you want to create. It is a very powerful system. It used widely in development, research and teaching especially in Software-Defined Networking, and OpenFlow. Mininet is included in Ubuntu as the Mininet package and it exists under a liberal BSD-style.

I was testing topo.py file from Mininet system. With topo object, we can add nodes, add hosts, add switches, add port and add links. It is a well-structured programming code. Until now, I did not find any bug in the software under test. So, it is error free and there are no functions I tested them has any bug. The system works perfectly as it is an extremely important system. It is a very important system and wildly use to help researchers to do their experiments and get a perfect result. With the great propose of the system, the testers did a huge work to ensure that there is no any fault or error in the system. They were awarded of any bug might happen and work hard to fix it. So, the system tells the user what should he do first.

At starting using tstl, it was very hard to understand it as there is no manual or book to read which can help us learn fast, easily and efficiently. After that, I worked very hard to get familiar with it by reading my notes from the lectures, lots of practice in using tstl and help from the TA and some students especially Hafed Alghamdi, I get familiar with tstl grammar and understand it more. My system had very basic functions in the start just add nodes without any assert property. After that, my system asserts that the added node exists. I added add hosts and add switches with some assertion.

By the end of the term, my tstl file will be tested the most functions in topo.py file with more properties and assertion to ensure that every function is working correctly.

As I said the system is solid and there are no bugs in it and the system aware of any error the user might do it. For example, addPort() function does not allow you to add port before add switches. When I add port first the file complains that need switch first which is good to give me warning what should I do. Also, addLink() function we should add nodes first and then add a link between them. If I did not add node the system is complaining about it. It is a very strong system.

About test coverage, I am still working on that. I covered a large part of the python code. Using tstl tool is a very interesting as it is a new experience for me and it is a very helpful tool to find bugs in Software Under Test or prove that Software Under Test is perfect with bugs free.

References:

- **[1] https://github.com/mininet/mininet/wiki/Ideas#background**