UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Mininet Network Emulator

by

Ignatius Thomas de Villiers
17502292

Department Electrical and Electronic Engineering,
Stellenbosch University,
Private Bag X1, Matieland 7602.

Dissertation Report

Study leader: Dr H.A. Engelbrecht

November 2016

Decleration

I, the undersigned, declare that this dissertation is my original work, gathered and utilized especially to fulfil the purposes and objectives of this study, and has not been previously submitted to any other university for a higher degree. .

Signature: .....
I.T. de Villiers

Date: .....

Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special griatitude I give to my study leader Dr. H.A. Engelbrecht, whose contribution in simulating suggestions, project coordination and guidance helped me to write this report

Furthermore I would Like to give my gratitude to Mr I.D. de Villiers (Software Development Engineer, Amazon), who's contribution in terms of software structure guidance and the reviewing of all functional code, has helped me to complete the Mininet Network Emulator Tool.

SKRIPSIE PROJECT: SUMMARY

Student: I.T. de Villiers

Project Title
Mininet Network Emulator
Aim of this Project
The aim of this project is to develop a tool with which one can build a virtual network topology, and simulate the Network's performance by using various user defined network parameters, without having to physically set up the nwtwork.
How the project was Implemented
A Literature study was performed before starting on the project, to sharpen up my knowledge on The following topics: 1. The Concept of a Network Emulator 2. Mininet (instant virtual network) 3. Object orientated programming 4. Invoking system utilities from Python 5. Parsing output files 6. Passing command line arguments to script and to see what tools are available The Concept of a Network Emulator, and how it can be used.
Conclusion of Results
...
Usefulness of Results?
...
What aspects of this project will continue after completion of the project
...

Student Date Lecturer

Abstract

The aim of this project is to develop a tool with which one can build a virtual network topology, and simulate the Network's performance by using various user defined network parameters, without having to physically set up the nwtwork. The project was broken up into two main components namely: (1) a completely vitual network and (2) a Semi-virtual network

Case 1: Fully-Virtual Network

In the completely virtual case, all components are simulated and a simulated network is used to emulate the performance of the network topology through various standard mininet tests.

Case 2: Semi-Virtual Network

In the semi-virtual case, all components are real hardware components (Hosts and Switches) and a simulated network is used to emulate the performance of the network topology through various standard mininet tests.

Furthermore Various test Situations were created to test both Cases 1 and 2, to ensure that both of cases function
Identically under certain network conditions

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Testing of network Topologies has become increasingly difficult, as the diversity and complexity of topologies and technology has increased dramatically. Networks did not simply get faster, but more diverse, carrying more diverse traffic. Links between network entities like hosts and switches vary in network conditions like bandwidth, delay, loss and jitter. Often Networks are set up, neglecting the varying network conditions of each link. Meaning that the network might not perform as initially expected, sometimes to the extent that the network is useless for relaying information, streaming of video or playing games. This can be an enormous waste of time, resources, productivity and energy. The solution to this problem is Network Emulation in the form of a Network Elulator.

The two most important methodologies for the design and testing of network protocols are simulation and testbeds. Simulation provides for controlled, repeatable tests, but uses simplified assumptions that are not realistic to real world conditions. Network testbeds successfully mimic network conditions, but are very labour intensive and costly. A network emulator may be considered a hybrid between these two main methodologies.
A Network Emulator can be described as a test platform that allows a user to mimic network conditions of a real network to test the network's performance, stability, or functionality. Network emulation is also used to test a product's performance under real world network conditions.These Network Emulators are handy tools that prevent costly post-deployment operational and performance issues.

To address the issue of increasing difficulty in testing complex network topologies and to create an environment where custom network topologies can be created and tested, a *Mininet Network Emulator* was developed. The Mininet Network Emulator allows the user to Emulate various custom topologies under user-defined network conditions like bandwidth limitations, delay, packet loss and jitter.

1.1 Literature Study

1.2 Aims and Objectives

1. Emulate various custom topologies under user-defined network conditions.
   (a) Using all virtual components (Hosts, switches, etc) over the emulated network.
   (b) Using some real components over the emulated network.
2. Create a easy-to-use Graphical User Interface, to allow customization of:
   (a) Topology
   (b) Network conditions

3. Allow user to save custom topologies
4. Test all custom topologies under same conditions
5. Stream a video and/or DHCP server over the network and to see accurate results
6. Create an accurate network simulator capable of:
   (a) Setting up network with it's conditions
   (b) Streaming over the network
   (c) Alternative testing of network conditions

1.3 Graphical User Interface

To ensure that the user gets the most out of the Mininet Network Emulator, it is important to allow the user to specify the network Topology and set various network conditions, like *bandwidth*, *delay*, *Packet loss* and *jitter*. When the program is Started the first window that appears, is the *Working Directory* Window, shown in Figure 1.4.1. This window prompts the user to enter a save location for the topology they are about to create. As seen in figure 1.4.1, the main window consists of three widget frames, namely the ***Hosts Widget***, ***Switches Widget*** and the ***Links Widget***. Each of these widgets allow the user to add hosts and switches and to customize the link parameters for each host to a switch. Figure1.4.2 shows the inner workings of the GUI.

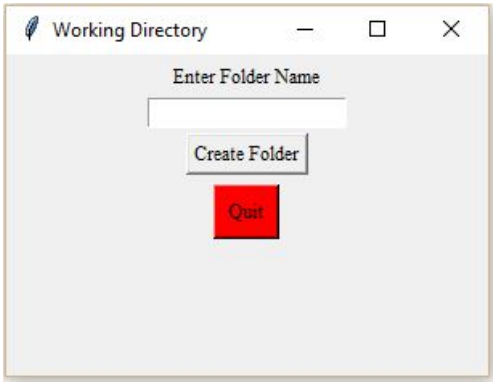

Figure 1.3.1: Initial Window on starting the Mininet Network Emulator

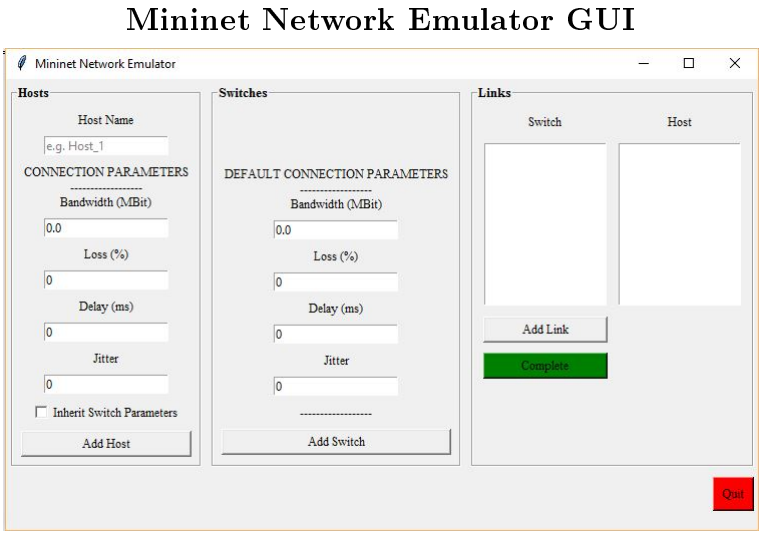

Figure 1.3.2: Mininet Network Emulator GUI Frame

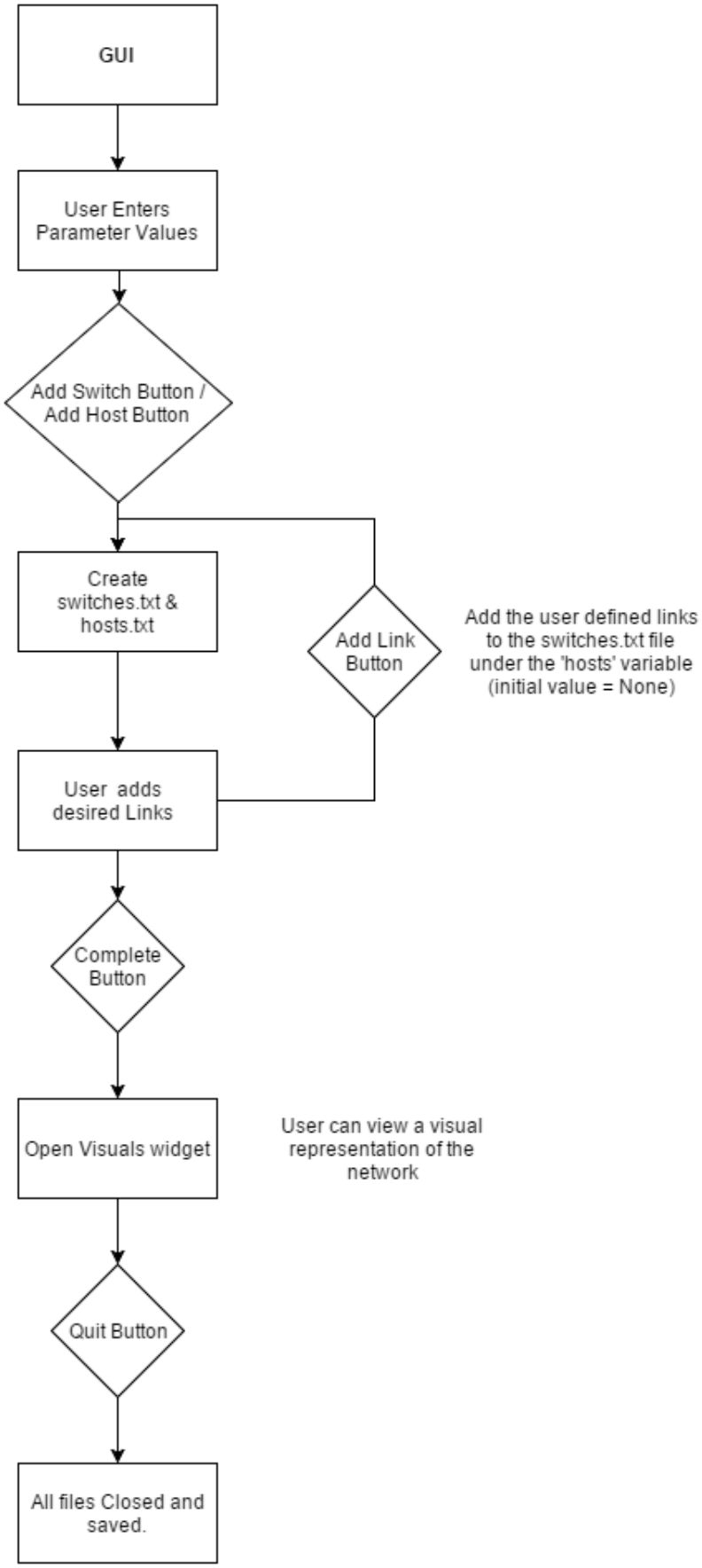

Figure 1.3.3: GUI flow chart - showing all steps the during the creation of a topology

1.3.1 Hosts Widget

The Host widget essentialy, as seen in Figure 1.4.3, allows the user to add host entities to the network. The user can customize the Host ID/Host Name by entering a name into the first textfield. In the case that the user leaves the textfield blank, a default name *'h1', 'h2', 'h3', etc* will be used. The user can also add connection parameters/non-idealities for each individual Host, *bandwidth*, *delay*, *Packet loss* and *jitter*, to simulate a real life connection. Value testing prevents the user from entering invalid parameters into textfields, like text and other ascii values, to ensure only valid entries are saved. Default values for all the connection parameters are 0. The radio button ***Inherit Switch parameters***, instantly changes the Host link parameters to inherit the switch's connection parameters when it is selected. When the ***Add Host Button*** is pressed, the host name and all it's connection parameters are written into a text file *'hosts.txt'*, as seen in Figure 1.4.3, which is then later used to create mininet objects- this will be discussed later in this document.

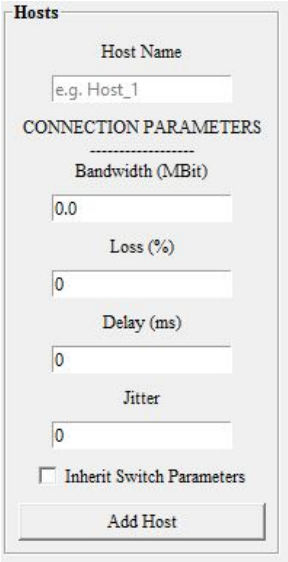

Figure 1.3.4: Hosts widget, which allows the user to add Hosts to the network, and customize it's connection parameters

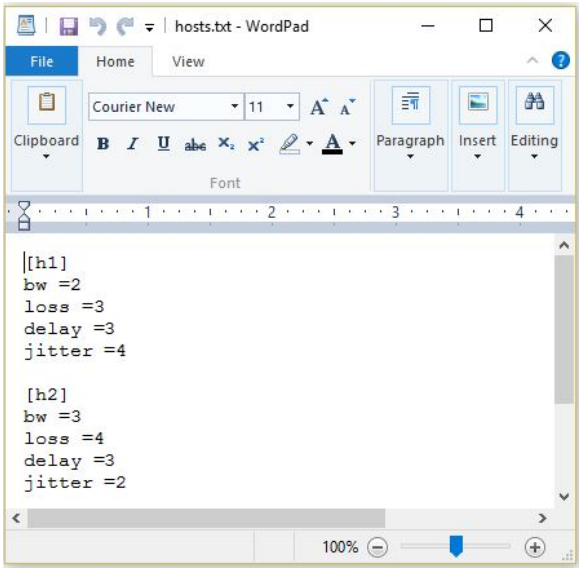

Figure 1.3.5: Example of created Hosts text file

1.3.2 Switches Widget

The Switches widget, as seen in Figure 1.4.5, essentialy allows the user to add switch entities to the network. A default name *'S1', 'S2', S3', etc* is automatically assigned to each switch entity. The user can also add connection parameters/non-idealities for each individual switch, *bandwidth*, *delay*, *Packet loss* and *jitter*, which is similar to the Host connection parameters but are only used when the user decides to add baseline parameters, which a link will inherit in the case that the Host connecting to the switch's parameters are 0. Value testing prevents the user from entering invalid parameters into textfields, like text and other ascii values, to ensure only valid entries are saved. Default values for all the connection parameters are 0. When the ***Add Switch Button*** is pressed, the switch name an all it's connection parameters are written into a text file *'switches.txt'*, as seen in Figure, which is then later used to create mininet objects- this will be discussed later in this document.

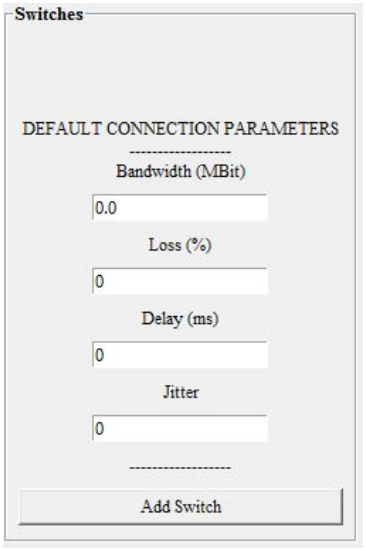

Figure 1.3.6: Switches widget, which allows the user to add Switches to the network, and customize the default connection parameters

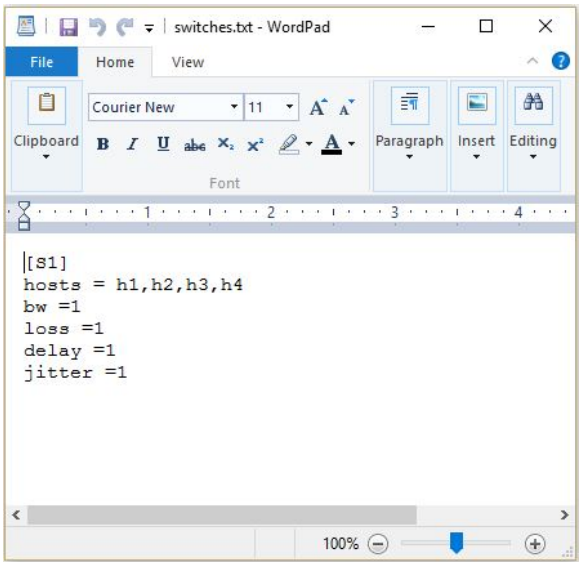

Figure 1.3.7: Example of created Switches text file

1.3.3 Links Widget

The Links widget essentialy allows the user to link network entities, namey x amount of hosts to 1 switch. As shown in figure 1.4.7, two Listboxes are used, the Host-listbox and the Switches-listbox, the user selects one or more hosts from the first listbox and then selects the switch they want to connect the hosts to. After they have decided which links they want make, the user can then click the ***Add Link Button*** which will then save the Hosts connected to a specific switch and save it to the textfile *'switches.txt'*, as seen in Figure 1.4.6, to indicate which entities are connected to one another.

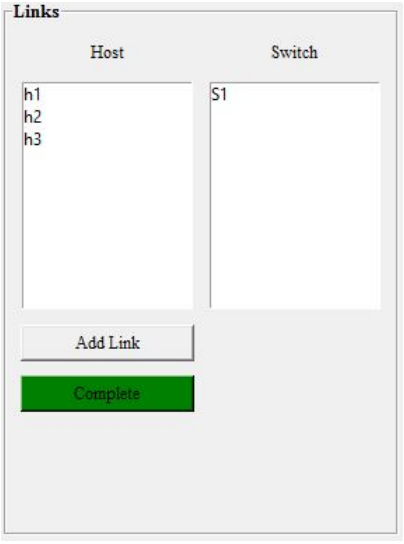

Figure 1.3.8: Links widget

1.3.4 Visuals Widget

Once all the Network Entities are added and the necassary links were created, the Visuals widget, in the form of a tkinter canvas widget, allows the user the graphically see the Network. This widget could come in handy when very large topologies are created - this will allow the user to review his work. The user is also able to move the objects around within the tkinter canvas, in order to created and test various setups. In Figure 1.4.8 we see the visuals created for a basic of a Star Topology example.

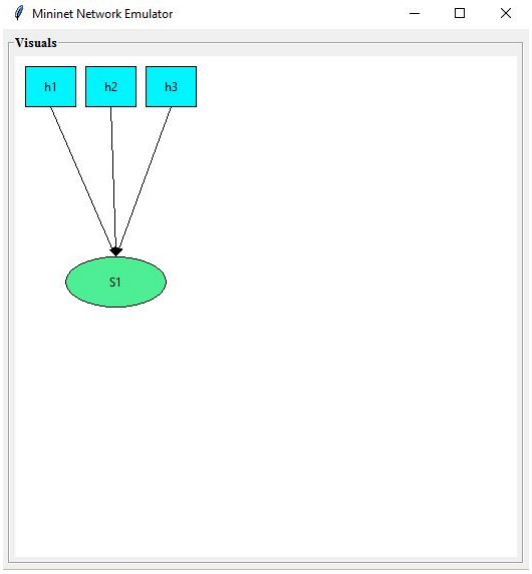

Figure 1.3.9: Example: Three Hosts, one Switch, star topology ***Complete Button*** is pressed.

Appendix A

Konsepte Gegenereer

A.1 Konsep I

A.2 Konsep II