

Master Thesis

**SDN based Network Management in Emulated environment**

|  |  |
| --- | --- |
| Submitted by: | Harshal Rajan Vaze |
| Matriculation no.: | 1269879 |
| First examiner: | Prof. Dr. Ulrich Trick |
| Second examiner: | Prof. Dr. Armin Lehmann |
| External supervisor: | Dr. Peter Gröschke |
| Date of start: | 01.05.2022 |
| Date of submission: |  |

Statement

I confirm that I have written this thesis on my own. No other sources were used except those referenced. Content which is taken literally or analogously from published or unpublished sources is identified as such. The drawings or figures of this work have been created by myself or are provided with an appropriate reference. This work has not been submitted in the same or similar form or to any other examination board.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date, signature of the student

Content

[1 Introduction 4](#_Toc108739170)

[1.1 Aim and Motivation 4](#_Toc108739171)

[1.2 Problem Statement 4](#_Toc108739172)

[1.3 Thesis Structure 4](#_Toc108739173)

[2 Theoretical Background 5](#_Toc108739174)

[2.1 Software defined Networking 5](#_Toc108739175)

[2.2 SDN Controllers 5](#_Toc108739176)

[2.2.1 ONOS 5](#_Toc108739177)

[2.2.2 ODL 5](#_Toc108739178)

[2.2.3 Ryu 5](#_Toc108739179)

[2.3 Virtual Emulated Environment 5](#_Toc108739180)

[2.3.1 Mininet 5](#_Toc108739181)

[2.3.2 GNS3 5](#_Toc108739182)

[2.4 Previous Work 5](#_Toc108739183)

[3 Requirements Analysis 6](#_Toc108739184)

[3.1 General Objectives 6](#_Toc108739185)

[3.2 Work Plan 6](#_Toc108739186)

[3.3 Previous Work 7](#_Toc108739187)

[4 Realization 8](#_Toc108739188)

[4.1 With Mininet 8](#_Toc108739189)

[4.2 With GNS3 8](#_Toc108739190)

[4.3 Challenges faced 8](#_Toc108739191)

[4.4 Use Cases 8](#_Toc108739192)

[4.4.1 Basic network architecture 8](#_Toc108739193)

[4.4.2 Testing network with multiple same Controllers 8](#_Toc108739194)

[4.4.3 Testing network with multiple different Controllers 8](#_Toc108739195)

[4.4.4 Testing network with IPv6 addressing 8](#_Toc108739196)

[5 Summary and Perspectives 9](#_Toc108739197)

[6 Abbreviations 10](#_Toc108739198)

[7 References 11](#_Toc108739199)

[8 Appendix 12](#_Toc108739200)

# Introduction

Dr. Martin Casado developed an architecture to separate control and forwarding functions of networking de-vices, migrating control to a centralized policy server. This architecture evolved to what is now known as Software Defined Networking (SDN) today. One of the first challenges was the need for a common South Bound Interface (SBI) protocol between the SDN Controller and the forwarding networking device. OpenFlow developed by the Open Networking Foundation (ONF) and is used over a secure channel (Transport Layer Security (TLS) over Transmission Control Protocol (TCP) port 6633 and 6653) to modify the group and flow tables in a OpenFlow networking device. OpenFlow has evolved to version 1.5.1.

Graphical user interface, application

Description automatically generated

Figure of VMs in virtualization platform

## Aim and Motivation

## Problem Statement

## Thesis Structure

# Theoretical Background

## Software defined Networking

## SDN Controllers

### ONOS

### ODL

### Ryu

## Virtual Emulated Environment

### Mininet

### GNS3

## Previous Work

# Requirements Analysis

Due to advances in the Information and Communication Technology, the configuration and management of the network components becomes highly complex and time-consuming. A fundamental characteristic of SDN is the logically centralized, but physically distributed controller component. SDN offers to batch-configure automatically multiple components in one step, while the traditional way would mean to log into each device. Many operators struggle with the migration from IPv4 to IPv6, SDN with its centralized control and the possibility to reduce human error due to increased poses an opportunity to help make this migration easier. The controller maintains a global network view of the underlying forwarding infrastructure and programs the forwarding entries based on the policies defined by network services running on top of it. The traditional networking approach has very limited facilities to explore these aspects of networking and the goal would be to study these futuristic characteristics of networking.

## General Objectives

* Research possible open-source SDN controllers to implement.
* Research alternative configuration methods with the goal of finding the best possible method to configure and manage the network through Network Controller.
* How to provide different paths in the network with different QoS properties?
* Algorithms that are responsible for the optimization of the paths.
* When a service is accessible at multiple times, how to choose the best one

## Work Plan

* Various Open-source SDN controllers available for implementation. Currently many Open-source SDN controllers are available to be deployed and tested in the environment. To name few, OpenDayLight (ODL), Open Network Operating System (ONOS), Ryu and Faucet.
* ONOS and ODL are built to have centralized architectures. Hence, they tend to be easier to maintain and confer lower latency between the tightly coupled southbound APIs and Northbound APIs.
* Faucet is built to have distributed architectures which generally are more complex to maintain and deploy but can allow the platform to scale more effectively. By decoupling the processing of PCE, Telemetry and Southbound interface traffic, each function can be scaled independently to avoid performance bottlenecks.
* Whereas, Ryu is different to the other options, although having a core set of programs that are run as a platform, it is better thought of as a toolbox, with which SDN controller functionality can be built.
* ONOS and ODL are written in Java, for which development resources are abundant in the market, with good supporting documentation and libraries available.
* Ryu and Faucet are written in Python, a well-supported language and has an active community developing the framework. The documentation is concise and technical, aimed at developers to maximize the utility of the system.
* Both ODL and ONOS benefit from large developer and user communities under the Linux Foundation Networking banner. Many large international players are involved in the development and governance of these projects, which could add to the longevity and security over time.
* Ryu and Faucet are well supported, targeted controllers. Due to the emerging nature of the field, both options look to have a bright future, with a simpler, streamlined approach to change submission and testing.

## Previous Work

* SDN controllers are mostly associated with the Mininet for the Network elements to be tested. Mininet being network emulator to be used to deploy easy network topology with Open vSwitch supporting SDN protocols.
* Not many with GNS3

# Realization

After researching about various SDN controllers, the aim would be to choose a suitable SDN controller for building an environment and creating some use cases with it. The environment will be build keeping following points into consideration;

* Build a suitable network with different network devices in the emulation software.
* Manage different services and network configurations with SDN controller in an emulated environment.
* Create and distribute the network configurations for network devices.
* Develop a rationale and setup an IPv4 and IPv6 scheme for the network.

## With Mininet

## With GNS3

## Challenges faced

* Need to specify OpenFLow version while executing commands on Mininet or Open vSwitch.

## Use Cases

### Basic network architecture

### Testing network with multiple same Controllers

### Testing network with multiple different Controllers

### Testing network with IPv6 addressing

# Summary and Perspectives

# Abbreviations

**0**

…

**3**

3GPP Third Generation Partnership Project

…

**A**

…

**H**

**N**

NBI Northbound Interface

NetConf Network Configuration Protocol

**O**

ONF Open Networking Foundation

ODL OpenDayLight

ONOS Open Network Operating System

**S**

SBI Southbound Interface

SDN Software Defined Network

…

**Z**

…

# References

# Appendix