Study on Optimization for Software-Defined Networks Controller

Omran Maki Abdelsalam Alssaheli[1,*], Z. Zainal Abidin[1], N. A. Zakaria[2],

[1]Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
[2] Centre for Advanced Computing Technology, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

[*]Corresponding e-mail: imran20032006@yahoo.com

Keywords: Optimization; SDN; Network

ABSTRACT – Software-defined networks (SDNs) are a new type of network for solving the limitation of legacy networks. In fact, SDNs is capable to control the network traffic performance from the centralized center remotely. However, packet loss and link congestion lead to poor network performance. Therefore, an enhancement approach is proposed, which the spanning tree algorithm is implemented on the SDN controller to handle loops, delays and packet losses. The expected outcome should produce an increment value of throughput and achieve optimization in the network performance. The experiments were conducted using Mininet emulator tool (SDNs) Floodlight (SDNs Controller) on Open Flow protocol.

1. INTRODUCTION

The Internet is a computer networking system that connects thousands of computing devices worldwide. Earlier, gadgets have included desktop workstations, computers and servers that store and transfer information, such as e-mail messages and web pages. At present, more devices are being connected to the network, such as game consoles, TVs, laptops, mobile phones, webcams, electronic security systems and even cars. The concept of computer networking is becoming outdated as a large number of non-traditional devices are connected to the Internet. At present, networking technology has become an integral part of most human activities. The number of devices and the amount of traffic moving on the Internet are increasing rapidly [1].

Computer networks are constantly evolving. New techniques and methods are required to manage a network. Software-defined networks (SDNs) are a type of network that can solve the drawbacks of legacy networks. The main difference between SDNs and legacy networks is rooted in the manner in which these networks manage the entire network. SDNs use a centralised method to aid network administrators in configuring the setting of a network since it is convenience and flexible method. On the contrary, legacy networks manage a network by using a distributed method, which the architecture and protocol is using OpenFlow protocol.

The OpenFlow protocol is supported by network providers, such as Amazon, Google and Microsoft, and equipment vendors, such as NEC and Cisco. This protocol is one of the mechanisms that fit into the concept of SDNs. An OpenFlow controller delivers control messages to OpenFlow switches to direct them in processing incoming packets. At present, SDN OpenFlow networks are more applicable because of their improved networking efficiency [2]. SDNs architecture comprises data plane, control plane and SDNs applications. The data plane is made of OpenFlow-based physical or virtual switches that enables decoupling of control and data plane to independently produce a short evolutionary cycle. On the other hand, the forwarding plane evolution cycle aims to attain fast packet delivery. Thus, SDNs separate the forwarding plane and control planes. The separated control plane is called the SDNs controller [3]. Figure 1 illustrates the architecture of SDNs [4]. Theoritically, decoupling and separation of planes of virtual networks from the underlying network hardware is named as the network virtualization.

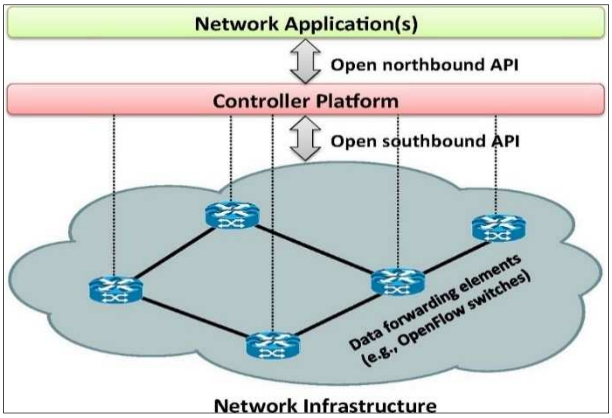

Figure 1 Simplified view of an SDN architecture

Today, network virtualisation is the main focus of research because of its dynamic programming, low cost and energy-saving capability. Virtualisation refers to creating a virtual version of an object. It takes the form of platform, hardware, network or server virtualisation. In contrast, simulation refers to designing a model of a real system, conducting experiments on a node, understanding the behaviour of a node in the network and evaluating its various operations. Many network simulators are available, such as NS2, NS3, OPNET, OMNeT++ and Mininet [2].

The Mininet emulator is an economical and quick means to configure a network test bed. In fact, the tool that supports the SDNs is OpenFlow network, as reported in the ONS conference. Mininet uses switches, virtual hosts and links to design a network on a single OS kernel that uses a real network stack to process packets and link them to real networks. Moreover, Unix/Linux-based network applications also run on virtual hosts. In an OpenFlow network emulated by Mininet, a real OpenFlow controller application can be executed on the same machine or on an external machine that emulates the virtual host. Although various network simulator tools exist [1,4], the current study focuses on optimising the performance of a network using Mininet.

2. Methodology

This research aims to implement a spanning tree algorithm on an SDNs controller to handle loops, delays and packet losses. This study is divided into three phases to achieve its objectives. The three phases are as follows.

a. **Phase I**: This phase is the initial phase. It involves the process of creating two scenarios of SDN networking
b. **Phase II** This phase focuses on implementation. Test cases are generated and testing tools are applied on the two scenarios.
c. **Phase III** This phase analyses the test results. Then, a comparative study is conducted between the results obtained by applying the testing tools on the two SDNs scenarios. Figure 2 illustrates the methodology.

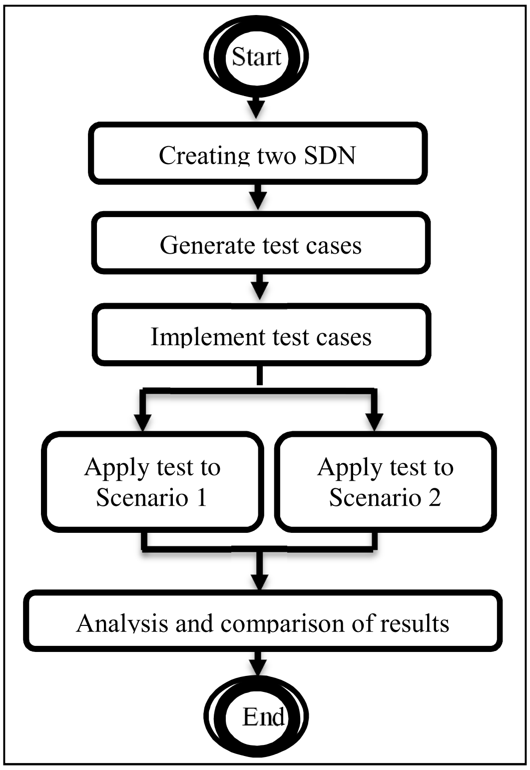

Figure 2 Methodology Process

3. Expected Results and Discussion

The expected results obtained by Mininet testing in the two SDNs scenarios are analysed based on three criteria: loops, delays and packet losses. The results of both scenarios are then analysed and compared to improve network performance. Table 1 illustrates the simplified view of the results of both scenarios.

First Scenario: Software defined network created by using Mininet-Network Emulation Tool and Floodlight. This scenario does not include any change in SDNs controller. The goal of this Scenario is to be used as basic SDNs-network and compare it with the second scenario. Moreover, to show where the loops, delay and packet loss are possible to happen.

Second Scenario: Software defined network created by using Mininet-Network Emulation Tool and Floodlight. This scenario includes implementing spanning tree algorithm in SDNs controller, to handle loops in the custom topology and to forward the packets between different networks. Similarly, the configuration file created in the new framework needs to be enhanced to specify the link characteristics such as bandwidth, delay, packet loss in SDNs controller.

Table 1 Simplified view of result

Scenarios	Time	Loops	Packet Loss
Scenario I	√	√	√
Scenario II	¯√	×	¯√

4. Conclusions

SDNs, with its inherent decoupling of the control plane from the data plane, offers considerable control of a network via programming. These combined features will result in potential benefits for enhanced configuration, improved performance and encouraged innovation in network architecture and operations. The potential benefit of SDNs is further evidenced by the fact that SDNs provides a convenient platform for experiments on new techniques and encourages new network designs. Such a possibility is attributed to SDNs' network programmability and its capability to define isolated virtual networks via the control plane. Regardless of these SDNs benefits, the existence of loops, packet loss and congestion which are cyclical paths through the network's switches that may prevent some packets from ever leaving the network, will be a potential problem in computer networks. Thus, implementing a spanning tree algorithm on the SDNs controller will improve network performance as it shown in Scenario II. For future works, we will deploy the proposed SDNs–Floodlight controller with STP and compare the result using the Mininet emulator tool.

5. References

[1] R. L. S. De Oliveira, C. M. Schweitzer, A. A. Shinoda, and L. R. Prete, In Communications and Computing (COLCOM), IEEE Colombian Conference on 2014, pp. 1-6.

[2] Veena, S., Rustagi, R. P., & Murthy, K. N. B..In *Advanced Computing and Communications (ADCOM),* IEEE, *Annual International Conference on 2014,* pp. 29-31.

[3] Erel, M., Teoman, E., Özçevik, Y., Seçinti, G., & Canberk, B. In *Network Function Virtualization and Software Defined Network (NFV-SDN), IEEE Conference on 2015,* pp. 18-19.

[4] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. Proceeding of IEEE, vol. 103(1), pp. 14-76, 2015.