Routing Implementation based-on Software Defined Network Using Ryu Controller and OpenvSwitch

Yuli Sun Hariyani*, Indrarini Dyah Irawati, Danu Dwi S., Mohammad Nuruzzamanirridha
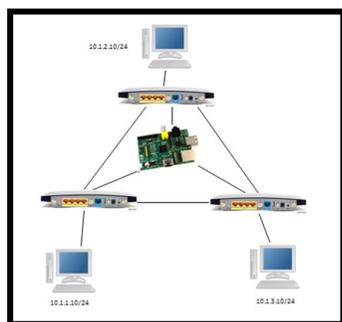
Diploma of Telecommunication Engineering, Telkom Applied Science School, Telkom University, 40257, Bandung, Indonesia

Graphical abstract

Abstract

Open Flow is a standard protocol for differentiating forward function and control functions to facilitate the management of big network of SDN. The research have been carried out before using the emulator SDN Mininet. However Mininet has many shortcomings, such as the performance of which is less than the maximum due to simulation. Then some researchers also use the Net-FPGA as device. This device is less suitable for small scale because the prices are quite expensive and programming is quite complicated. In this study, SDN implementation carried out using OpenvSwitch as forwarding function mounted on TP-Link that has modified using openwrt as firmware and Raspberry Pi with Ryu SDN Controller as control functions. The result shows that routing static can be implemented on SDN Network which use Raspberry Pi with Ryu Controller as control function with average bandwith 536.0909 Mbits/sec and average uptime network is 10.45 second.

Keywords : Software Defined Network, OpenFlow, OpenvSwitch, Raspberry Pi, TP-Link

1.0 INTRODUCTION

Nowadays, the development of computer network technology being rapidly. This evidenced by the addition of increasingly congested traffic because the service needs data. Traffic on the network is also becoming more complex due to the need of the user. However, the network infrastructure is still considered not capable of handling the requirements for network architecture at this moment still combines the functions of the controller in same device. Now, forwarding in emerging methods of new network architecture called Software Defined Network (abbreviated SDN).

SDN OpenFlow is a standard protocol for differentiating forward functionality and control functions to facilitate the management of big network of SDN. The research have been carried out before using the emulator SDN Mininet [1]. However Mininet has many shortcomings, such as the

performance of which is less than the maximum due to simulation. Then some researchers also use the Net-FPGA as device. This device is less suitable for small scale because the prices are quite expensive, programming is quite complicated. In this study, carried out using the SDN implementations OpenvSwitch as forwarding function mounted on TP-Link that has modified using openwrt as firmware and Raspberry Pi Ryu SDN Controller as control functions [2].

2.0 LITERATURE REVIEW

In this section, we will describe the architecture used to build a network-based SDN. Based on research that had been done earlier, SDN architecture can be divided into three layers; infrastructure layer, control layer and application layer. Infrastructure layer is a layer which only handles forwarding function, and

does not handle control functions such as routing and switching. Existing devices on the infrastructure layer such as routers, switches and hosts. Control layer is a layer that functions to regulate all the functions of the network centralized. Infrastructure layer performs a request to the control layer to determine the function of each device. The application layer serves to set policies that will be made to the SDN architecture can be run in accordance with the rules that have been made.

2.1 OpenFlow

OpenFlow is a protocol that allows for setting routing and delivery of packets on a switch. In conventional network architecture, each switch only serves to forward the packet through the appropriate port without being able to distinguish the type of data that is sent [3, 7]. OpenFlow protocol can perform the functions of flow-based forwarding network layer and packet flow settings centrally from layer 2 to layer 7 (granuality flow), so that the flow of packets in the network can be set up and configured independently. This can be done by making algorithms and its forwarding rules on the controller which are then distributed to the switches that are in the network. Methodology known as software-defined networking (SDN) has been implemented in various companies such as Google, HP and IBM.

2.2 OpenWRT

OpenWRT is an operating system / Linux-based embedded operating system kernel, which is basically used for embedded devices that serves as the routing of network traffic. The main component is the OpenWRT Linux kernel, util-linux, uClibc, and busybox.Semua components are optimized to match the wireless router that has a RAM capacity and limited storage media [9].

OpenWRT configuration can be done using command-line interface or GUI using Web.Selain, there are about 3500 packages, one of which is OpenvSwitch available as additional applications and can be installed using packages opkg management system.

2.3 OpenvSwitch

OpenvSwitch is an application that represents a multilayer virtual switch under the Apache 2.0 license as per shown in Figure 1. OpenvSwitch designed to support changing network architecture with network automation technology towards programmatic extension [7]. OpenvSwitch also supports standard management interfaces and protocols such as NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag. In addition, OpenvSwitch can also be used as a distribution between different server platforms. Some features supported by OpenvSwitch are:

1. Automated Control
2. Monitoring
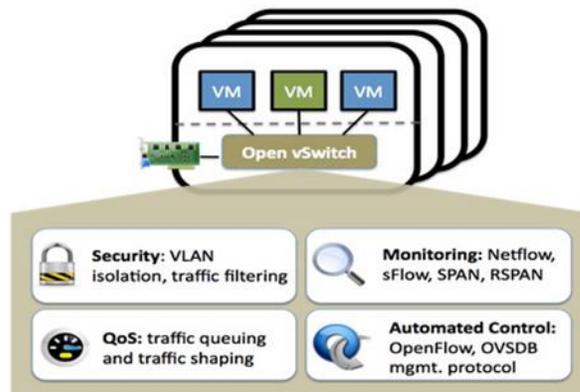3. Quality of Service (QoS)
4. Security

Figure 1 OpenvSwitch Architecture

3.0 IMPLEMENTATION SCHEME

In this research, we use single Raspberry Pi B model as a SDN Controller, three TP-Link WR1043ND v.2 are installed OpenWRT and embedded OpenvSwitch as switches and forwarding of data flows, and three PC or Laptop as hosts.

3.1 Network Design

Software Design are using Ryu Controller on Raspberry Pi operating system Debian Wheezy 3.12 kernel version that has the ability to control functions of routing and switching.

Then to switch is used OpenWRT Linux-based operating system that has been modified with the addition of OpenvSwitch as a function of the data flow forwarding.

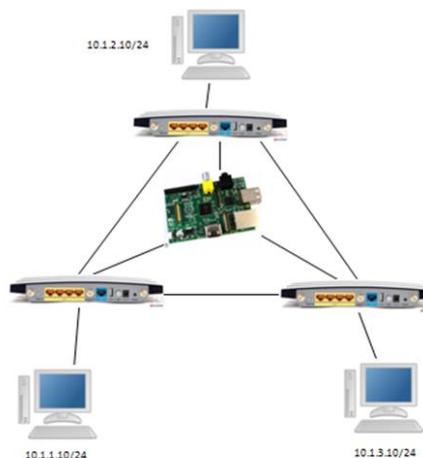The network architecture used on the software can be seen at Figure 2.

Figure 2 SDN Network Topology

3.2 Software Design

Figure 3 shows the software architecture of the implementation. We use Raspberry Pi model B as an embedded controller and Ryu as a framework. TP-Link using OpenWRT operating system and embedded 14:07 Barrier Breaker 2.3.1 OpenvSwitch as forwarding.
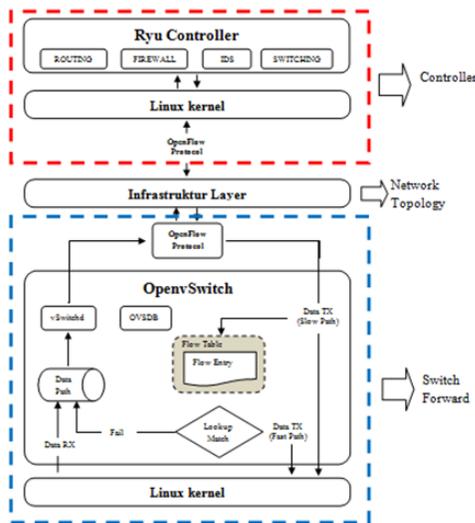

Figure 3 SDN Software Architecture

Figure 4 SDN Software Architecture

Figure 5 Link Uptime Chart

4.0 RESULTS AND DISCUSSION

We implement the SDN using topology that shows on Figure 2 using static routing. First, we operate the TP-Link as OpenFlow switch using OpenvSwitch and connect all switch to Raspberry Pi. Next we run the static routing of controller and configure static routing.

The result shows that our test successfully operate SDN function with static routing. Each host has a different network based on subnetting. We do testing for each host and measure network performance using jperf.

In this research, there are two scenario that implemented. First scenario is used to measure average bandwidth on network. We measure bandwidth within 30 seconds and show the bandwidth average is 536.0909 Mbits/sec. The graphic result in Figure 4.

Second, we measure uptime link within 10 trials for every link. The different Link are being unconnected and reconnected again. Uptime parameter used to measure how long the network can operate again after reconnected. The average of Link uptime are 10.45 second. The graphic result can be seen on Figure 5.
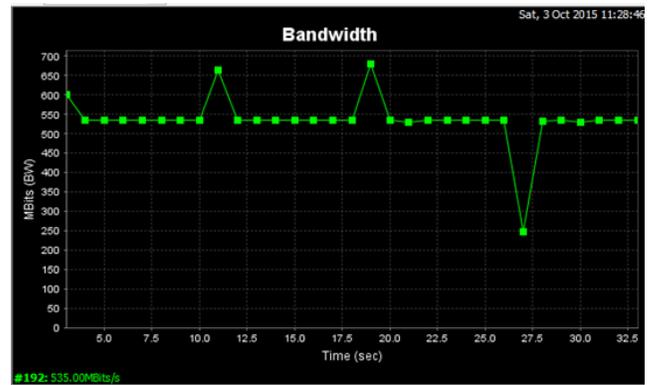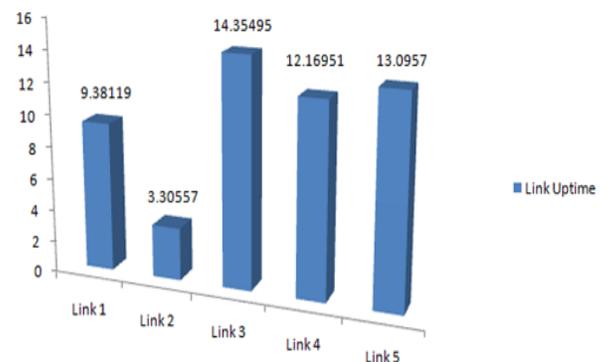
5.0 CONCLUSION

In this paper can be concluded that the TP-Link and Raspberry Pi can provide a satisfactory measurement results for the implementation of SDN network for small scale. SDN applications can run well on low-end hardware specification. In addition, the price of the hardware is cheap and easy to do the programming and measurement. Also, the TP-Link also has more than one port to test more than one host on each network.

Acknowledgement

References

[1]   Irawati, Indrarini D. and M. Nuruzzamanirridha. 2015. Spanning Tree Protocol Simulation Based on Software Defined Network Using Mininet Emulator. *Communications in Computer and Information Science.* 516: 395-403.

[2]   Braun, Wolfgang, Michael Menth. 2014. Software-Defined Networking Using OpenFlow: Protocols, Applications and

Architectural Design Choice. Germany: Future Internet 2014.

[3]   Kim, Hyunmin, Jaebeom Kim, Young-Bae Ko. 2014 .Developing a Cost-Effective OpenFlow Testbed for Small-Scale Software Defined Networking, Korea: ICACT2014.

[4]   On data center scale, OpenFlow, and SDN. [Online] Diakses pada 13 November 2014 Available: http://bradhedlund.com/2011/04/21/data-center-scale-openflow-sdn/.

[5]   Ryu SDN Framework. [Online]. Available: http://osrg.github.io/ryu-book/en/Ryubook.pdf.

[6]   What is OpenFlow? [Online] Available: http://archive.openflow.org/wp/learnmore/.

[7]   What is OpenvSwitch? [Online] Available: http://openvswitch.org/.

[8]   What is OpenWrt? [Online] Available: https://openwrt.org/.

[9]   What is Ryu? [Online] Available: http://osrg.github.io/ryu/.

[10] Yamahata, Isaku. 2013 Ryu: SDN framework and Python experience, Japan: Pycon APAC 2013, September 14.

[11] TP-LINK WR1024ND Build OpenvSwitch with OpenWrt. [Online] Available: http://hoootony.logdown.com/posts/259102-tp-link-wr1024nd-build-openvswitch-with-openwrt.

[12] OpenvSwitch Lab 7: Setting OpenWrt [Online] Available: http://roan.logdown.com/posts/239799-openvswitch-lab-7-setting-openwrt.