International Journal of Engineering & Technology, 7 (2.29) (2018) 821-825

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

The Implementation of DHCP Relay Using Pox Controller on Openflow Protocol

Eki Ahmad Zaki Hamidi *, Mufid Ridlo Effendi, Nanang Ismail

Electrical Engineering Department, Universitas Islam Negeri (UIN) Sunan Gunung Djati Bandung Indonesia
**Corresponding Author E-Mail: Ekiahmadzaki@Uinsgd.Ac.Id,*

Abstract

The fastest of development in a network which is longer more complex, it needed developing and managing efficiently. On a network which has more computer, it needed effectivity of regulation of IP with DHCP server, many more of a subnet, has become DHCP Relay to be a solution. DHCP relay or agent relay is a protocol Bootstrap which is DHCP asks to have a message between client and server for DHCP to the different network. Software Defined Network (SDN) offers a new paradigm in network design, manage, and implementation, especially to support a needed and innovation in this case, which is longer more complex..

Keywords: *DHCP Server; DHCP Relay; Software Defined Network.*

1. Introduction

Communication between a computer in a networking system needed IP address which can do by the manual system to make the configuration in each computer. It could be not effective if it used in big scale. DHCP (Dynamic Host Configuration Protocol) server became an automatic machine to rule IP address when it connected to the network.

DHCP (Dynamic Host Configuration Protocol) relay is used to the big scale network which has more computer, so it can make more subnet. A subnet is a small network but it principal parts of the network between subnet which is connected with gateway or router. By using DHCP, each subnet needs DHCP server. So it can make a load of the computer, especially if there is more of the subnet which is used. To solve the problems above, we could use DHCP relay agent. So by using DHCP relay agent, it only uses one DHCP server, and request of IP address which comes from each subnet will be sent by DHCP relay agent to each subnet to DHCP server.

Software Defined Network (SDN) offers a new paradigm in which how to design, to rule and to implement the network, especially to support a need and innovation in this case, which is more complex. DHCP relay could be implemented to the OpenFlow protocol. OpenFlow is part of the component in architectur SDN, OpenFlow its open standard for protocol communication between control and forwarding plane.

2. Literature Review

2.1. Dhcp (Dynamic Host Configuration Protocol)

DHCP (Dynamic Host Configuration Protocol is one of protocol in networking system which can give or lent IP address to the host in a network automatically. In general, DHCP server has a group of address which is permitted to distribute to the client, which is called DHCP pool. Each client will give IP address from DHCP pool will the time determined by DHCP. When the time of IP address is expired, a client will ask for the server to give a new IP address or lengthen. DHCP client will try to get while IP address from a DHCP server by four steps, here is the steps on one subnet and different subnet as on Fig 1 (1).

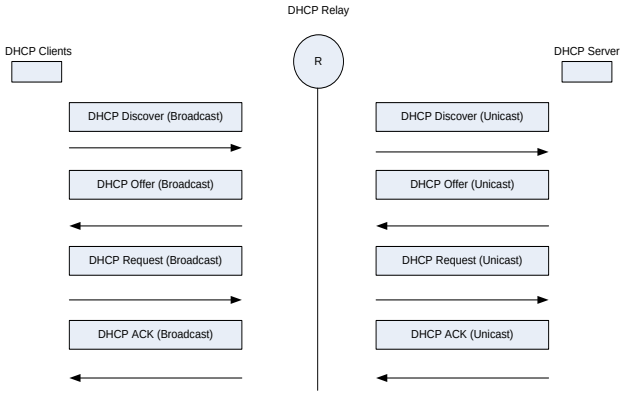

Fig. 1: Steps of DHCP

The way how IP address to the client which has not IP address:

1. DHCP Discover: DHCP Client will distribute a request by broadcast to find an active DHCP server.
2. DHCP Offer: after DHCP receives a broadcast from DHCP client, DHCP will offer an address to DHCP client.
3. DHCP Request: Client asks DHCP server to rent IP address by one of an address of DHCP pool to DHCP server it self.
4. DHCP ACK: DHCP server will be the response a request from a client by sending an acknowledgment packet. Then, DHCP server will determine an address (and configure another TCP/IP) to the client, and renewal database itself. The next client will do the binding process by protocol pile of TCP/IP and because it has an IP address, a client will start a network.

The four of steps above are valid only to the client which has not an address. A client which has asked an address to the same

DHCP server before, only the third and fourth step which can be used, that is address renewal it is more fastest.
It is different with DNS system that is distribution, a character of DHCP is stand alone, so, it does not need many DHCP in a network, database IP address in DHCP server would not be replicated to another DHCP server. It could be a problem.
If configuration between two DHCP server are the collision because both two host are not permitted to have the same address by protocol IP. Beside could prepare a dynamic address to the client, DHCP server also could determine a static address, so the address of client could be fixed all the time.
To administrate a small network, it would be easy to a network administrator if IP static is given. But, if a network is widely probability, using the same IP is more biggest, so it could be a conflict, because of this, so using DHCP server is more suggested by (1).

2.2. DHCP Relay

In a network, a computer which places in the same of a subnet with a computer which asks IP address, it is functioned as DHCP relay agent. DHCP relay agent is to continue a request message to DHCP server which places in another subnet.
By using DHCP relay agent, so DHCP server will be used to serve a request from the different subnet, so, it will be minimized a cost of tool and regulation of configuration.
Another advantage of DHCP relay agent is, it could be delay response to the computer which asks serve in many seconds, so it could erase traffic network if there was DHCP server on the local subnet and DHCP server to the another subnet. If DHCP relay agent continues a request of the computer to DHCP server to another subnet, while on local subnet was a DHCP server. So duplicate answering of data from each DHCP server would be happened at the same time. It would make a crowd of the traffic network. But, by doing delay which has been done by DHCP relay agent, so request data will be continued after a while. So that if in a local subnet was a DHCP server, so computer which asks to serve, would get an IP address from DHCP server local. Whereas answering from DHCP server to another subnet would be neglected and would not be sent again to the computer which asks to the server.

2.3. Software Defined Network

Software Defined Network (SDN) is a new concept to design, to manage, and to implement the network, especially to support a needed and innovation in this network, that has more complex. The base concept of SDN is by dividing explicit between control plane and forwarding plane, and by doing abstraction system and to cover the complexity in which component or subsystem by definite standard interface.
There are many aspects that have more important by using SDN, such as:

1. Separatelized between *forwarding/data-plane* and *control-plane*
2. Interface standard (*vendor-agnosic*) to programme network.
3. Centralized *Control-plane* or network operation system. That cloud make logical map by all the network such as API *(Application Programming Interface*)
4. Virtualized, which is many network operation system could control (*slices* or *substrates*) by the same network.

The facts of SDN with innovation needed to the network are more complex (2).

1. Virtualized and Cloud: are component and entity hybrid network between bare metal physically and virtual.
2. Orchestration and Scalability: are an ability to role and manage thousands network by a point of management.
3. Programmability and Automation: are ability to change network behaviour and to do the changes automatically, for example: troubleshooting, policy changes, etc.
4. Visibility: is an ability to monitor a network, such as resources, connectivity, etc.
5. Performance: is an ability to maximize the use of the network, for example, optimizing bandwidth, load balancing, traffic engineering etc. (connectivity with Programmability and Scalability).

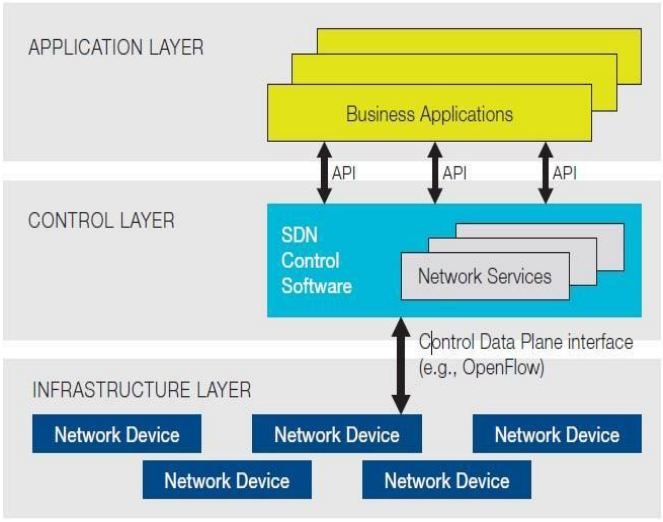

Fig. 2: SDN Architectur3,

SDN Architecture are3:

- In infrastructure (data-plane / infrastructure layer): parts of sub-network, that could role SDN Datapath based on instructions which are given by Control-Data-Plane Interface (CDPI).
- Control (control plane/layer): entity control (SDN Controller) translates application needed to infrastructure by giving right instruction to SDN Datapath and good information that needed by SDN Application.
- Application (application plane/layer): stay on the first layer, communicate via NorthBound Interface (NBI) system.
- Management and Administration have a responsibility to initiation subnetwork, to pair with SDN Datapath by SDN Controller, or configuration (coverage) by SDN Controller and SDN App.

2.4. OpenFlow

OpenFlow is an open standard that offers to control the networking equipment pro-grammatically. It was originally designed for network researchers to test their experimental protocols on real networking hardware devices and campus networks that represent everyday networking environments like LAN and WAN. The networking research community has been facing extremely high barriers to experimenting new ideas or protocols with the production or traditional networking environment. In order to use the available networking hardware and deployed the network to test on experimental protocols and new ideas, OpenFlow emerged out in late 2008 and released its first specifications in December 2008 (3).
OpenFlow offers a control program for the network administrator to run properly by taking the flow from the source to the destination and take advantage of the flow-based processing for packet forwarding. OpenFlow offers a way to eliminate the router to define packet processing path, saving power and cost in network management primarily in expanding the network (3).

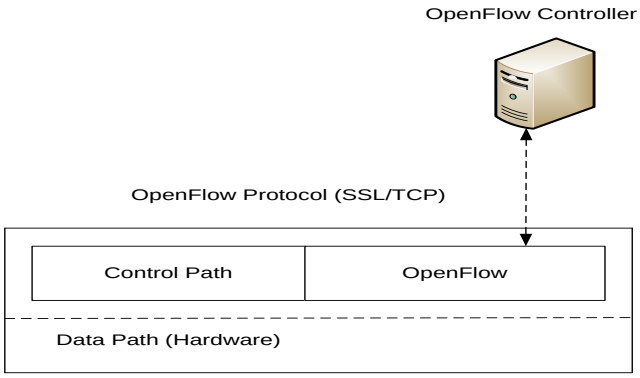

Fig. 3: OpenFlow Architectur.7

In OpenFlow architecture, datapath flow forwarding still resides on the switch, but flow forwarding decisions are made in a separate OpenFlow controller or hierarchy of controllers, which is implemented in a serv-er(s) that communicates with an OpenFlow-enabled switch(es) in the network through OpenFlow protocol.

Therefore, the main components of an OpenFlow network are:

- Switch(es) with OpenFlow support
- Server(s) running the controller(s).

2.5. Open vSwitch

Open vSwitch is an open-source, multi-layer software switch that has been aimed at managing large-scale virtualized environments. It is motivated by growing virtualized environment needs, and a superset of OpenFlow protocol is utilized for configuring switch forwarding path. It utilizes centralized controller approach for connecting to OpenFlow enabled switches; however additional management interfaces such as SNMP can be used for configurations.

It functions as a virtual switch, provides connectivity between virtual machines (VMs) and physical interfaces. It also emulates OpenFlow protocol in a Linux-based virtualized environment including XenServer, kernel-based virtual machine (KVM), and VirtualBox. It also provides useful tools and utilities for emulating OpenFlow protocol, which is:

- ovs-vsctl - a utility that queries and updates configuration of soft switch
- ovs-controller - a reference OpenFlow controller
- ovsdbmonitor - a GUI tool for viewing Open vSwitch databases and flow tables.
- ovs-ofctl - a utility that queries and controls OpenFlow switches and controllers.

The architecture of Open vSwitch can be seen in Figure 4 below, where ovsdb-server is the database holding the switch level configuration, and ovs-vSwitch is the core component of the Open vSwitch. Ovs-vswitchd supports multiple datapaths and checks datapath flow counters for flow expiration and stats queries. Ovs-vswitchd is the communication hub that communicates with outside world, ovsdb-server, kernel module and the whole Open vSwitch system. VMs connect to the Open vSwitch kernel through virtual interfaces, and kernel provides connectivity to OpenFlow protocol and the under-lying physical interfaces (3).

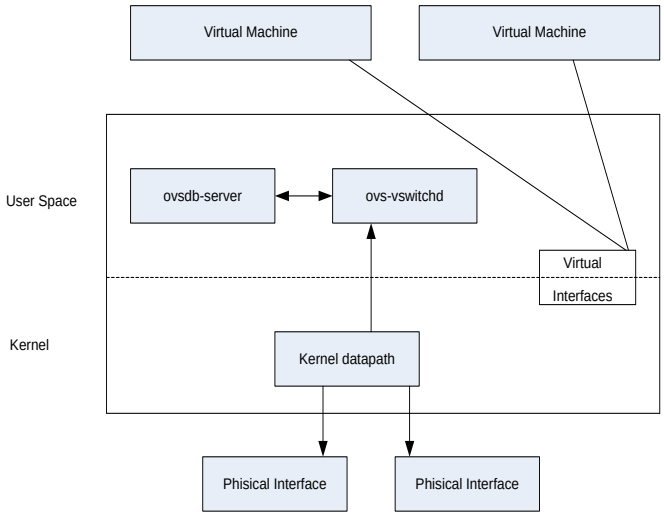

Fig. 4: Architecture of the Open vSwitch

2.6. Controller

The controller is a centralized entity that gathers control plane functionality – creates updates and removes flow entries in flow tables on a switch, where a flow refers to the unidirectional sequence of packets sharing a set of common packet header values. Along with its primary function, it can further be extended to perform additional critical tasks such as routing and network access.

Currently, there are several controller implementations available, which are open-source and are based on different programming languages such as Python, C++, and Java 8. In this thesis, an open-source Java-based controller named Floodlight controller has been chosen for conducting experiments. Typically, a controller runs on a network attached server and can serve one or multiple switches depending on the network design. It can be designed with centralized hierarchy where one controller handles and controls all the switches in a network or distributed hierarchy where two or more controllers handle and control two or more groups of switches in a network. In a centralized hierarchy, if a controller fails then all network operations are interrupted, and it poses a single point of failure in an OpenFlow network(3).

In the distributed hierarchy, all the controllers should have the same copy of the network topology view in real time to avoid the packet losses. The network topology view includes the switch level topology; the locations of users, hosts, middleboxes, and other network elements and services. Moreover, it includes all bindings between names and addresses. The most popular OpenFlow controllers: NOX, POX, and Floodlight.

2.7. POX Controller

NOX and POX are two complementary Open Source control platforms for Software Defined Networks. NOX is the original OpenFlow controller which initially developed at Nicira Networks side-by-side with OpenFlow. Since the time it being released under the GPL in 2008, it has been used by dozens of research groups as well as industry stakeholders. The most recent version of NOX represents a change of focus: providing a lighter, higher-performance framework for controller development using C++ on Linux. NOX modules were originally written in C++ or Python [7]. The development of NOX has brought up some issues when using two programming languages in implementation. Thus, POX is developed to run purely in Python and the Python modules in NOX is removed so that NOX can be a better platform for those who want to develop a controller module in C++. Besides, NOX is more difficult to deploy than POX. Most of the users find it difficult to build and run NOX in their network environment because of NOX's high number of dependencies.

POX is actually NOX's younger sibling project. In contrast with NOX that targets high-performance, POX primarily targets re-

search and education. It aims to facilitate jumping right into SDN and rapidly prototyping new ideas. POX applications are written purely in Python, and run easily on Windows, Linux, and Mac OS. The research group in Stanford University uses POX for research on key problems of SDN: abstractions, programming models, and debugging tool. Their intention is to continually move good ideas from research to the public release of POX, attempting to converge on a prototypical modern SDN controller design, with a meaningful collection of applications, and a suite of tools for debugging and simulation [7]. Therefore, we chose to use POX controller because it is more user-friendly and easier to implement.

3. Methodology/Materials

In this chapter, Implementation of DHCP Relay was conducted in the laboratory of electrical engineering UIN Sunan Gunung Djati Bandung.

3.1. Hardware

Hardware used is as follows:

- Intel Core2Duo E7400 @2.8GHz
- RAM DDR2 PC5300 2GHz

3.2. Software

Title must Software which is used in this experiment is open source software.

- OpenVswitch v. 2.3.1
- ISC DHCP Server 4.3.1
- Controller POX brach eel

3.3. Network Topology Setup

Topology models used is implementing the DHCP Relay are is follows:

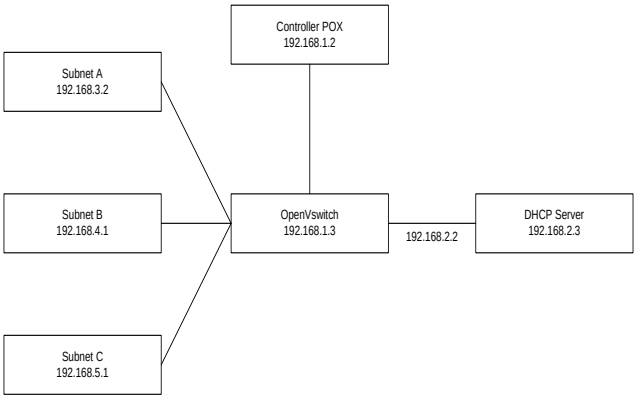

Fig. 5: Topology Models

3.4. Research Flow Chart

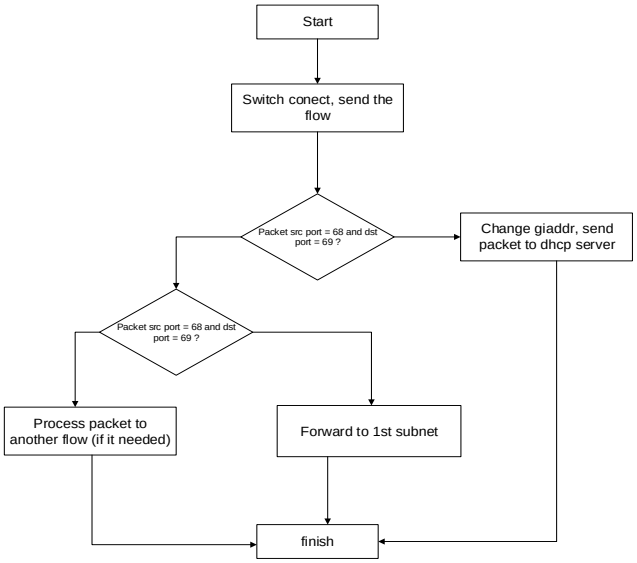

Fig. 6: Reseacrh Flow Chat

Fig..6. Shows the Flow DHCP Relay is processing which has been used OpenFlow and shows that using OpenFlow will do DHCP Relay.

3.5. The use of DHCP Relay

If the host asks IP, so it can do broadcast packet by using port source 68 and port destination 67. If there is not DHCP server in a subnet, but it was a DHCP relay, so that packet would be accepted by DHCP relay and then it would be forwarded to DHCP server in another subnet. But before forwarded to DHCP server, DHCP relay will change the contents grader based on IP gateway subnet host which is requested by IP.

DHCP server receives a packet from DHCP Relay and then go to check content giaddr to make sure IP address which is given to the host. After IP address is fixed by DHCP server, the packet will be sent to DHCP relay to be forwarded to the host.

The way of DHCP Relay, a packet which is received by DHCP relay from host to be forwarded to DHCP server before, if would be changed the content of giaddr without change content of another packet. So that, what it needs is to change the content of giaddr in OpenFlow.

Giaddr on the application layer of switch OpenFlow could be read packet until the fourth layer. So, it needed the controller to read layer application plus to change the content of giaddr.

3.6. Notes:

- The packet which is received by switch OpenFlow, if it does not suite the with table flow, so it will be sent to the controller, but packet which is seat to the controller only a piece of information, that is a header packet which contents of information which is needed by the controller.
- A solution of a packet which is seat by switch OpenFlow to the controller, in order to make the flow to the switch to forward packet DHCP to the controller.
- Flow is sent to the switch when a switch is doing connection the first time to the controller.

3.7. Operational System

Linux Operating System used on all computers, real or virtual. GNU / Linux Debian operating system selected is free and open code (open source). Debian has one policy in any publishing software must be stable and all hardware is supported Debian should be run so that if one only hardware not supported by the software,

Debian will not be published. Therefore, the expected operating system GNU / Linux Debian completely stable when used in this implementation.

4. Results and findings

In this research, a contribution that is resulted are:
1. Implementation DHCP Relay to the OpenFlow Protocol.
2. Make a testbed DHCP Relay by OpenFlow in Laboratory Electrical Engineering State Islamic University of Sunan Gunung Djati Bandung.
3. OpenFlow could change DHCP Relay function.

5. Conclusion

By doing the experiment above, it gets:
1. On subnet A, subnet B, and subnet C success to get an IP subnet which is different and can be connected to the network has been done.
2. The implementation DHCP relay by using box controller on OpenFlow protocol could work well.

Acknowledgement

I would like to thanks the Dean of Science and Technology Faculty who has facilitated the participation in the conference. Also, thanks to Electrical Engineering Department to allow me using the facilities in the network laboratory.

References

[1] Yeganeh SH, Tootoonchian A, Ganjali Y. On scalability of software-defined networking. IEEE Communications Magazine. 2013;51(2):136-41.
[2] Mateo MP. OpenFlow switching performance. Politecnico Di Torino. 2009.
[3] Khatri V. Analysis of openflow protocol in local area networks. 2013.