International Journal of Engineering Research and Reviews ISSN 2348-697X (Online)
Vol. 3, Issue 3, pp: (35-39), Month: July - September 2015, Available at: www.researchpublish.com

Automating The Open Flow Rule Engine Datapath Automation

[1]Prerana B.H, [2]A Thyagaraj Murthy

[1] Dept. network & Internet SJCE Mysore, India
[2]Dept. Electronics & communication SJCE Mysore, India

***Abstract*: Open Flow is a programmable network protocol designed to manage and direct the network traffic among the network devices like switches and routers etc. Open Flow separates the control plane of a switch from the switch itself. The high-level routing control which includes packet routing (forward, Drop and redirect) and so on, is moved to a centralized controller. This paper proposes a Data path automation, where Open Flow rules are simulated and set up on switches and the controller imposes the flow policies on the network packets generated by Ixia, via Open Flow protocol. The Open Flow rule simulation engine is much faster, requires less man power and it is platform independent. Open Flow-capable switches, enables a wide range of new network functionality. Open Flow rule simulation engine is helpful in debugging software faults (or bugs) as well. We believe that tools for testing Open Flow programs make it more agile and flexible.**

***Keywords:* Openflow rules, columbus app, Ixia-generate the packets .**

I. INTRODUCTION

Open Flow is an open standard that was initially designed to be able to run experimental protocols in production networks. By now it has evolved into a key enabler of Software Defined Networking. It provides a separation of control and data plane on a switch (or any networking device) and an open interface between control and data plane. It also provides an open interface to the control plane. The combination of all these allows one to have network control and management features in software on an external controller, which becomes the new centralized control plane. Meanwhile switch is unburdened to do what it does best – move packets around.

Open Flow – a pragmatic compromise:

- Utilizes the speed, scale and fidelity of vendor hardware
- Provides the flexibility and control of software and simulation
- Leverages a common set of functions supported by vendor hardware

In Software defined networking world, to validate the OpenFlow software quality defined procedures are executed to configure the flows manually on the switch, configure the match parameter fields in ixia, and once the packet hits the rule, manually observe the packet capture or number of the frames received from the output port of the ixia. This consumes lot of man power. This problem is addressed using the OpenFlow Rule Engine.

This application based on the OpenFlow would provide a platform to validate the Open Flow data path on the switch. This would involve retrieving the flows from the Columbus database, parsing them, creating c URL commands to push them to IXIA and switch and finally validate the datapath and hence the software functionality. The whole framework will assist in ensuring the quality of Switch software. The application will let the user to validate various packet types and match parameters there by covering the complete scope of data path.

Our goal is to help these programmers produce reliable new Open Flow applications. Designing domain-specific programming languages is one approach to prevent common coding mistakes. Not surprisingly, existing Open Flow applications are written in TCL scripting. Rather than design a new language, we are creating tools and techniques for testing Open Flow applications, to detect and eliminate bugs and in turn checking the quality of switch.

II. OPEN FLOW BACKGROUND

The Open Flow protocol allows programs running on a logically-centralized controller to coordinate a distributed collection of switches

A. OpenFlow switches:

An **Open Flow switch** has a flow table that stores an ordered list of rules for processing packets. Each rule consists of a pattern matching on packet header fields, actions such as forwarding, dropping, flooding, or modifying the packets, or sending them to the controller and priority to distinguish between rules with overlapping patterns, and a timeout indicating whether/when the rule expires. A pattern can require an “exact match” on all relevant header fields i.e., a micro flow rule, or have “don’t care” bits in some fields i.e., a wildcard rule For each rule, the switch maintains traffic counters that measure the number of bytes and packets processed so far gives the statistics and In our rule engine based on the match parameters hits the packet count and the successful action(s) is performed. When a packet arrives, a switch selects the highest-priority matching rule, updates the traffic counters, and performs the specified action(s).

B. Centralized controller:

An OpenFlow network has a centralized programming, where Flare software controllers manages the underlying switches. The controller installs rules in the switches and columbus application installed on the controller produced flows/packets based on the table capabilities advertised by the switch to the controller. A controller application defines a handler for each event like rule timeout and packet arrival, which may install new rules or issue new request for traffic statistics.

Most OpenFlow applications are written on the Flare controller platform [12], which offers OpenFlow API for applications written in Python or Java. These controller applications are general-purpose programs that can perform arbitrary computation and maintain arbitrary state.

III. OPEN FLOW RULE SIMULATION PSEUDO CODE

The automation of Open Flow rule simulation needs the following steps

1. Start
2. Design topology mapping
3. Configure the workstation with flare controller
4. Connection between the DUT (switch) ,IXIA and Open Flow controller
5. Check For the Activation between the switch and controller
6. Install the Columbus Application at controller end.
7. Based On the table Capabilities advertised by the switch Columbus application generates the flows
8. The Flows are pushed one by one and check for the required match Parameters to which each flow performs required actions
9. Packet count is Hit when the match Parameters are equal
10. Process repeats for n flows

IV. ARCHITECTURE

The below schematic gives the brief idea how the open flow rule simulation works.

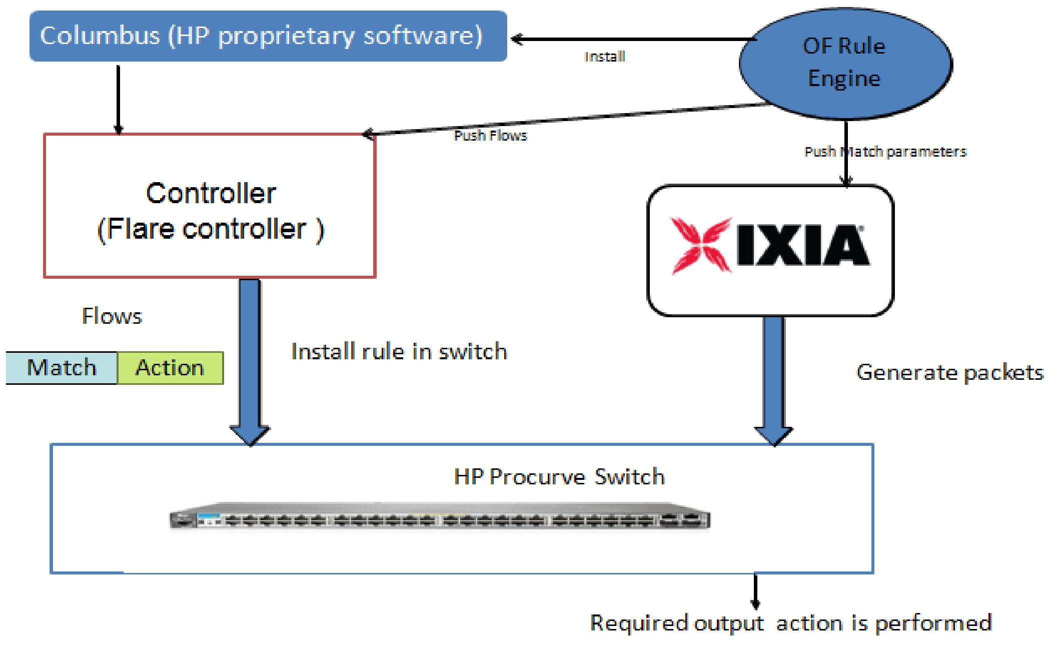

Figure 1. OpenFlow Rule engine system Architecture

The OpenFlow Rule simulation consists of basic components like controller ,Ixia(traffic generator) and the switch .These components are connected as shown in the architecture .Basic configurations are done from the TCL scripting

Configurations for the controller end are as follows:

- Installing the flare controller
- Installing the Columbus app
- Getting the Authentication Key and the Datapath ID

***-Installing the flare controller*:**

Flare is an SDN controller that simplifies the creation of software for controlling and monitoring the network .programs written within flare (using java) have flow-level control of the network .this means they can determine which flows are allowed on the network and decides to which path the flow should take . This installation of flare is done on the workstation by using commands. After the installation we need to check whether flare controller is up .

***-Columbus Application*:**

Installing the Columbus application on the controller. The main function of the application is when the OpenFlow switch enabled and the application is in running state ,switch advertise the table capabilities to the controller .This application Builds the database based on the table capabilities. This database consists of flows that can be pushed to the IXIA and switch.

-*Authentication key* is used for the secure communication between the controller and the switch .*Datapath ID* is used to determine the instance that is connected to the controller

Configuration of Open Flow switch:

To configure the OpenFlow switches we need to enter to the config mode and enable Open Flow. The following parameters need to be configure to enable the Open Flow switch

- controller instance-name
- controller- id
- listen-port
- member-vlan and untagged port

- create controller-vlan, assign IP address and untagged the ports

Ixia

Ixia is mainly used for traffic generating purpose .This provides test systems and service verification platforms for wired and wireless infrastructure and services .The flows generated by the Columbus application is pushed to ixia such that traffic is generated based on the match parameters

The Verification is done by using the packet count. Switch software compares the match parameters sent by Ixia and the controller is same .If its equal then packet count is hit and required actions are performed based on the rule simulation. Removing the previous rule and adding the next flow, the verification process repeats for n number of flows

V. SEQUENCE DIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operates in Openflow Rule engine and gives a view of Message Sequence Chart. The sequence diagram shows object interactions arranged between the hardware and software .Sequence diagrams is typically associated with a Logical View .The Packets or flows sent to the switch end from a controller where the Columbus application generates the database of flows as shown in below figure

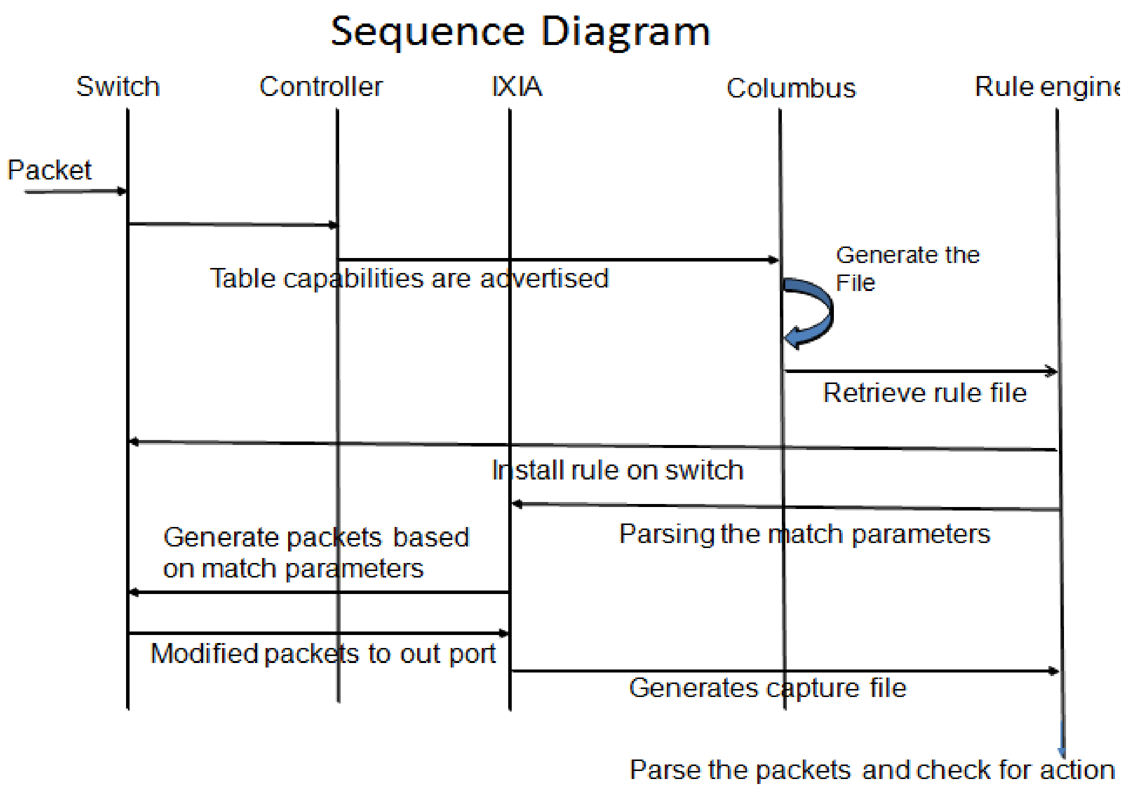
Sequence Diagram
Switch
Controller
IXIA
Columbus
Rule engine
Packet
Table capabilities are advertised
Generate the File
Retrieve rule file
Install rule on switch
Parsing the match parameters
Generate packets based on match parameters
Modified packets to out port
Generates capture file
Parse the packets and check for action

Figure 2. Open Flow rule Simulation Engine Sequence Diagram

VI. CONCLUSION

The Open Flow rule simulation engine is intelligent enough to pick the flows and generate the flow based on the match parameters of the flow and perform the required actions. The TCL scripting helps to reduce the man power in configuring each device. This can be implemented on any switches independent of platform . The Open Flow rule simulation engine helps to pick up the flows fast and perform the required actions

VII. FUTURE WORKS

The Open Flow rule simulation at datapath end testing can be used for the complete pipeline process at present our engine is used for single table .Currently the efficiency of flow rate is 1flow per min, it can be improved .The Open Flow rule simulation engine can be made even more intelligent so that it can pick some flows based on the rules.

REFERENCES

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," SIGCOMM Comput. Commun. Rev., vol. 38, pp. 69–74, March 2008.

[2] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. Gude, N. McKeown, and S. Shenker, "Rethinking enterprise network control," IEEE/ACM Transactions on Networking, vol. 17, Aug. 2009.

[3] A. Nayak, A. Reimers, N. Feamster, and R. Clark, "Resonance: Dynamic Access Control for Enterprise Networks," in ACM Workshop on Research on Enterprise Networks (WREN), Aug. 2009.

[4] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, "Plug-n-Serve: Load-balancing web traffic using OpenFlow," Aug. 2009. Demo at ACM SIGCOMM.

[5] R. Wang, D. Butnariu, and J. Rexford, "OpenFlow-Based Server Load Balancing Gone Wild," in Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), Mar. 2011.

[6] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving energy in data center networks," in Proc. Networked Systems Design and Implementation, Apr. 2010.

[7] D. Erickson et al., "A demonstration of virtual machine mobility in an OpenFlow network," Aug. 2008. Demo at ACM SIGCOMM.

[8] "Research experiment disrupts Internet, for some," http: //www.computerworld.com/s/article/9182558/Research experiment disrupts Internet for some.

[9] "AfNOG Takes Byte Out of Internet," http://www.renesys.com/blog/ 2009/05/byte-me.shtml.

[10] "Reckless Driving on the Internet," http://www.renesys.com/blog/2009/ 02/the-flap-heard-around-the-worl.shtml.

[11] open flow switch components "https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.3.4.pdf"

[12] Flarecontroller"https://www.google.co.in/?gfe_rd=cr&ei=dLerVYJroe_zB8j8tJAD&gws_rd=ssl#q=hp+flare+control-ler".

[13] http://wripe11.cis.upenn.edu/program/papers/wripe11-paper5.pdf