Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 1, January 2015, pg.74 – 77

RESEARCH ARTICLE

Inter-Operability in Software Defined Networks

Sandeep Singh[1], R.A. Khan[2]

[1, 2] Department of Information Technology,
School for Information Science & Technology
Babasaheb Bhimrao Ambedkar (A Central) University, Lucknow, INDIA
[1] drsandeep.gbu@gmail.com

Abstract— Software Defined Network interoperate with other technologies like flow visor and open flow. This enables slicing of the network in many ways, including MAC, IP and VLAN. It also enables seamless interconnections with multiple controllers, providing the ability to run multiple applications each on a different controller. The purpose of this interoperability analysis is to test the applications of open flow in multi vendor environment. A Flow Visor has many applications; one of them being ideal for test environments to virtualized the network. The analysis shows different interoperable features in open flow based software defined networks.

Keywords— Flow visor, Open flow Switch, Software Defined Network, Open Flow Controller

I. INTRODUCTION

Open Flow Software Defined Networking came into existence at Stanford and University of California, Berkeley [1]. Initial development and a lot of work were done through Clean Slate program. Later Open Flow becomes popular around universities and research institutions during its deployment on Internet Backbone [2]. Open flow protocol was initially developed by Stanford University with its basic version 1.0 at the end of year 2009. Later the revised and improved version 1.1 was released at the starting of year 2011. The collective efforts were converted in to an organization as open networking foundation in March 2011 and all legal and intellectual property rights were transferred to it. This makes open flow to use as a commercial product [3].

A Software Defined Network (SDN) works on the open flow protocol which is made of by two basic components a controller and a switch [4]. Open Flow separates the control plane from the data plane and defines a standard open programming interface, previously not available [5]. The programming interface enables the controller to populate the flow tables of the switch, which determines what it matches on and what action the switch takes on each packet [6]. This open interface enables SDN where software can now customize how networks meet the needs of application and users. Now applications loaded on it can talk to controller via SDN application program interface. This programmable interface is well defined by each controller [7].

II. RELATED WORK

Software-defined networking (SDN) provides some important capabilities to help you get the best performance from applications running within a network environment. Essentially, both physical and virtual network devices, as well as the applications that leverage them (e.g., SQL Server, SharePoint), can work in concert and communicate with one another to deliver a truly dynamic, scalable network. This seamless communication requires interoperability between the various networking vendor products that are implemented on your network. Several standards have arisen around these north- and southbound interfaces [8].

Once a switch or router enables Open Flow, then the standard API is available and new and interesting ways to solve network problems can be developed in software and quickly deployed on the network [9]. Physical network devices and applications come from different vendors. Most customer networks are composed of a variety of these vendor products. Therefore, truly automated SDN that seamlessly lets workloads communicate with one another, regardless of where they are hosted, requires communication throughout all the physical and virtual pieces of the network. To accomplish this, SDN-solution vendors such as Microsoft must provide interoperability between their technologies and north- and southbound interfaces [10].

Interoperability standards between vendors, promise and demand are the tools to enable users to manage their virtual data centre and networks with a unified toolset, policies, and orchestration. Market demand has also influenced manufacturers' efforts to achieve interoperability. Without enough SDN penetration within their networks, customers simply weren't clamoring for assets that communicated widely. To date, SDN interoperability development has remained focused largely within vendors' own product lines. This step forward shouldn't be seen as trivial. In fact, it signifies an evolution within the marketplace. "It's happening because just about every switch vendor has decided they need to also supply a controller [11]. Many of the challenges surrounding interoperability have been a function of time. Vendors simply needed to gain experience with Open Flow 1.0 as a platform before tackling the task of getting everything to work together.

III. OPEN FLOW UTILITY

A. Benefits of Open flow

- Able to reduce CAPEX cost by decreasing the cost and complexity of networking hardware.
- Open flow is also having the potential to reduce OPEX cost by decreasing network management complexity.
- It has the potential to speed up application development and deployment through the use of SDN.
- A smooth transition from legacy equipment through the use of hybrid devices and the ability to work with legacy devices and protocols.

The greatest benefit of Open Flow is the flexibility that it has to address current and future network problems in many areas of the network, from the enterprise to the data centre to the service provider edge or core.

B. Examples of Open flow

- Traffic engineering in a typical open flow based environment.
- On-demand path provisioning for better network management.
- Dynamic path provisioning for avoiding temporary congestion.
- Load balancing for smooth packet delivery.
- Distributed firewall to ensure security parameters.
- Extending the network to the virtual switch for providing virtual connectivity.
- Testing the experimental protocols on network for research perspective.

Open Flow protocol provides the command set, or primitives, on top of the controllers for implementation purpose. By combining the strengths of each vendor’s approach, organizations will be positioned to maximize network and application performance with Software-Defined Application Services [12].

IV. INTEROPERABILITY

SDN solutions will interoperate with the rest of the infrastructure through standard protocols and APIs. Interoperability of Open Flow in a multi-vendor environment is critical to its success. This prevents users from getting locked into a single vendor solution and enables them to choose the best solution based on price, performance and features. We can divide interoperability into two layers: infrastructure and services. On the lower layer, infrastructure interoperability is how an SDN network communicates with other networks -- this is where the use of existing standard protocols makes the most sense [13]. It is critical to test real use-cases of Open Flow in a multi-vendor environment. While the most critical component is Open Flow running between a controller and a switch, it is also very important to understand how different switches behave and work together with various applications of Open Flow [14]. On the other side higher services layer is there, where most of the SDN magic will happen. Intelligently managing the resources of the network according to high-level, infrastructure-wide policies will require SDN vendors to interact with the systems and applications that the network supports. The services layer itself is generally divided into categories: operations and service differentiation. [15].

Another benefit of the testing and collaborative work is to ensure that the standard is clearly written and implemented in a standard way. In some cases there may be areas that need clarification or improvement. SDN

vendors should be integrating with infrastructure management tools to improve the operational footprint of the network. In addition to lower overall costs, SDN vendors can integrate with the surrounding systems to provide differentiated services on the network. Vendors can also move on to public demonstrations of their products and features after testing them in the private event [16].

A. Benchmarking

The primary goal of Interoperability in SD Networks is to define standards-based test methodologies to establish performance benchmarks for Open Flow devices. Methodologies will include testing Open Flow controllers and switches. Some examples include:

- Defining flow table capacity of an open flow switch.
- Setting the learning rate of an Open Flow system.
- Defining successful delivery of packets per second.
- Setting the handling capacity of maximum flow stats.
- Defining performance benchmarking parameters like throughput and latency.

B. Test cases of Open Flow

SDN advises the logically centralized network control on software platforms and applications, turns networks more and more into software-based systems. The hosts, emulated by test equipment, were connected using 0.5 GE copper. The uplink from the access to the core was 0.5 GE or 1GE, copper or fibre. The interconnection within the core was primarily 1 GE but some links were 0.5 GE. All of the Open Flow switches had an out of band (dedicated port) IP connection to an Open Flow controller. A single controller floodlight can be used for each test case until the Flow Visor phase of testing.

- Basic Testing
- Network Discovery LLDP
- Layer 2 circuit provisioning
- Layer 3 (IP) learning with dynamic provisioning and failover
- Load Balancing
- L2 MAC Learning
- Slicing the network with Flow Visor

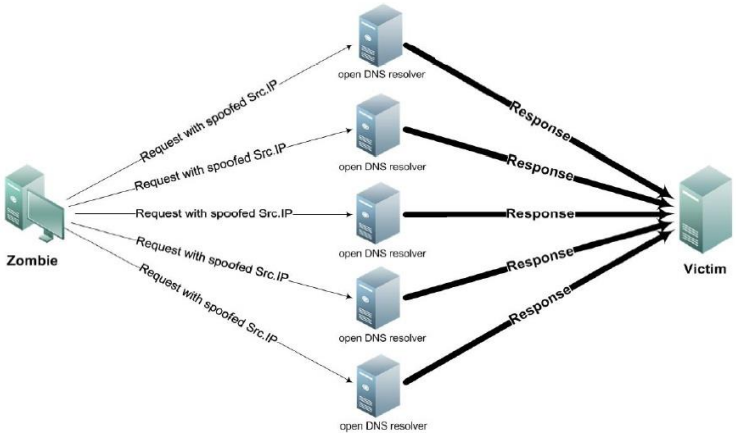

Figure 1. *Example Topology with Flow Visor as Proxy Controller*

V. CONCLUSION

To realize the vision, we need rich and open interoperability. We must follow standards but we all know that there are standards that fail to enable true interoperability. In purely software environments, open source can often simultaneously enable interoperability and API innovation better than standards, which typically move more slowly than open source software. On the other hand, insisting on end-to-end open source can potentially stifle innovation by discouraging investment. The core promise of SDN depends on efforts toward openness. Interoperability must include conformance testing. The ONF's Forwarding Abstractions working group's specially, which will be nearing release by the time of Interoperability, creates a framework that allows industry participants to precisely define and share specific SDN functionality "profiles" well ahead of SDN runtime.

REFERENCES

[1] Global Environment for Network Innovations. Web site http://geni.net.

[2] Wikipedia, "Internet backbone". Free encyclopedia of information [Online]. Available: http://en.wikipedia.org/wiki/Internetb_ackbone [Accessed: July 7th, 2014].

[3] Wikipedia, "Internet backbone". Free encyclopedia of information [Online]. Available: http://en.wikipedia.org/wiki/Distance-vector_routing_protocol [Accessed: July, 2014]

[4] Open Networking Foundation "Interoperability Event white paper" April 2012.

[5] "Why Interoperability Matters in Software Defined Networking, and How Microsoft Fits In" http://windowsitpro.com/rethink-your-datacenter/why-interoperability-matters-software-defined-networking. [Last accessed on Nov 30, 2014]

[6] "2014: The year of SDN Interoperability", Julie Knudson, http://www.enterprisenetworkingplanet.com/netsp/2014-the-year-of-sdn-interoperability.html [Last accessed on Dec 01, 2014].

[7] Priyanka Chopra and Pratha Datta, "An approach to Multi-vendor orchestration in a Software Defined Eco-system" white paper by Tata Consultancy Services, pp – 1-12, 2014.

[8] Curt Beckmann, "SDN: Open Source Advances Interoperability", March 2014, http://www.networkcomputing.com/networking/sdn-open-source-advances-interoperability/a/d-id/1114090 [Last visited: Dec 12, 2014]

[9] "Interface to the Routing System" IRTF Working Group. [Online]. Available: https://datatracker.ietf.org/wg/irs/charter/

[10] "Network Function Virtualisation" ETSI Industry Specification Group. [Online]. Available: http://portal.etsi.org/portal/server.pt/community/NFV/367

[11] "Netronome NFP6XXX Flow Processor" [Online]. Available: http://netronome.com/pages/ow-processors/

[12] "Use Cases for ALTO with Software Defined Networks," Internet Engineering Task Force ALTO Working Group. [Online]. Available: http://tools.ietf.org/pdf/draft-xie-alto-sdn-use-cases-01.pdf

[13] A. Tootoonchian and Y. Ganjali, "Hyper Flow: A Distributed Control Plane for Open Flow", in Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, 2010.

[14] S. Yeganeh et al. "On scalability of software defined networking", IEEE Communications Magazine, vol. 51, no. 2, pp. 136-141, February 2013.

[15] "Security Requirements in the Software Defined Networking Model", IETF Network Working Group Internet-Draft. [Online]. Available: https://datatracker.ietf.org/doc/draft-hartman-sdnsec-requirements/

[16] P. Porras et al. "A Security Enforcement Kernel for Open Flow Networks," in Proceedings of the 1st Workshop on Hot topics in Software Defined Networks. ACM, 2012, pp. 121-126.

AUTHOR

Sandeep Singh has received his M. Tech. in Wireless Communication & Networks from Gautam Buddha University, Greater Noida. He is a life time member of IACSIT, Singapore and member of CSI. His research interest includes wireless networks, Mobile IP; MPLS based networks and security of Software Defined Networks.

