ISSN: 2321-7782 (Online)
Volume 2, Issue 3, March 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

A Review on the Acceleration in Networking Fostered by Software Defined Networking

Rahul Dasharath Gavas[1]
Department of Computer Science & Engg,
KLE Dr.M.S.S CET, Udyambag
Belgaum – India

R. H. Vijaylaxmi[2]
Department of Electronics and Communication Engg
KLE Dr.M.S.S CET, Udyambag
Belgaum – India

Abstract: Software Defined Networking, abbreviated as SDN, is a renaissance in the field of networking that eases the work of network administrators by rendering a programmable, reliable and pluggable control over all the routers and network equipment using software approach. This has enabled a very dynamic approach to program the networks by the separation of control and data planes. Thus administrators can easily move the components across the network. It also aids in implementing virtual networks that overcomes many security and performance issues and other real time applications. The notable parameter in SDN is the data flow occurring in the network. The aim of this study is to focus on the growing acceptance of SDN and its applications in present day networking.

Keywords: SDN, control plane, data plane.

I. Introduction

A. Motivation

An evergreen topic of research, Internet, has been undergoing tremendous amendments right from its time of exploration. Many research projects and government programs have been introduced to make it better by focusing on network programmability. One such attempt by the U.S government was in the mid-1990s called the Defense Advanced Research Projects Agency (DARPA). It focused mainly on active networking so as to provide a programmable network interface in the data plane. This was followed by Future Active IP Networks (FAIN) program [1, 2]. After about ten years, the Forwarding and Control Element Separation (ForCES) [3] made use of a flexible modeling language for the flow logic and the flow tables. Though programmable network architecture seemed more appealing, it did not gain any real time acceptance.

Today's networks are poor in terms of security and are prone to errors. It includes many hardware components which are not reliable, since failure of one component has a negative impact on the other. This culminates in increased cost and adds to maintenance overhead. Also the complexity involved is very high. Software Defined Networking bestows us with the capability to innovate via software technology at a faster pace. This results in reduced network hardware that fosters easier and secure network deployments. This paves its way from the work at UC Berkley and Standford University around 2008 and is gaining momentum in terms of industrial and research communities.

B. Background

Based on the concept of Autonomous Systems (AS), networks which deployed Internet Protocols (IP) were used. This technique is robust, scalable and simple to use. The network components cannot move by changing their identity in the AS principle. Other drawbacks of AS include lack of access control, quality of service, etc. The next big thing by Internet Engineering Task Force (IETF) by the introduction of Complementary Standards like Virtual Private Networks (VPNs) and Virtual LANs. These standards have added to the complexity of networking. In contrast, the traditional networking practices provided no scope for programming. In the present era where mobile computer operating systems are used enormously, cloud architectures are evolving, virtualization has increased; the need for an additional layer of Software-Defined Networks (SDN) has arisen. It is possible to specify network services without combining the above mentioned services with the network interfaces. Due to this, the network elements can travel without having to change their identities.

C. Related Work

SDN enables severance of control and data planes [4] by 1. Discard off the control decisions from the forwarding hardware components. 2. Permitting the forwarding hardware to be programmable using and open interface. 3. Creating a separate unit called the controller to monitor the data transactions [5].

The functions of the controller unit influences the hardware used so far and can be virtualized. This proves to be a cost effective approach powered by reliability [6]. Earlier SDN applications and experiments built their own network maps using NOX [7]. But this urged the network operating system to build its own network maps [8]. SDN enables pluggable, programmable components in networking [9]. SDN also provides a logically centralized control plane.

D. Components and infrastructure of SDN

Like a traditional switch, the SDN controller implements the management plane and control plane. It is formally modeled as a dedicated appliance or software which is run using a server. There is logically centralized software for forwarding data packets and managing connectivity issues in the SDN. It uses overlay network or an existing physical network to provide connectivity. Another option available for providing end user connectivity is through programming individual network elements. The controller is responsible for providing virtual network abstraction. This prevents end stations from exposing themselves to the physical network infrastructure.

II. Categories and Interfaces of SDNs

A. Classification

A.1 Floodless and flood-based

In a floodless model, Distributed Caching and Distributed Hashing of SDN lookup tables are used to forward all the data packets.

Multicast and broadcast mechanisms are used for information sharing in a flood-based model. The major drawback of this approach is that the more the locations are added, the more is the increased load per location. This worsens the scalability.

A.2 Asymmetric and symmetric

Increasing the information distribution is of much concern in symmetric models. This model makes SDN elements to be a single component.

SDN information is centralized in an asymmetric model. The aspects of scale-out simplicity, coherency, are in critical state in this model. This was not the case while using the traditional AS based networking model.

A.3 Network centric and Host based

Host based deals with an assumption in which components with virtual machines move to enable elasticity.

B. OpenFlow

OpenFlow[10] is a protocol through which SDN is implemented. It lets network administrators to choose the route for the data to flow in the network. It is an interface between data and control planes. It enables direct control over the data plane. It is a communications protocol that gives access to the forwarding plane of a network switch or router over the network [11]. As per the contents of the “Flow Table”, this contains an entry for each flow. The flow table contains information of how to forward data for a given flow. This table is built using the rules given to the switch by the controller. It is managed by Open Networking Foundation (ONF) which is dedicated to promotion and adoption of SDN. This is led by a board of directors from seven companies like Facebook, Deutsche Telekom, Google, Yahoo, NTT, Microsoft, and Verizon. The “Open Flow Switching” trademark will be licensed by the ONF to the companies adopting this standard.

A software based approach to access the flow tables is attained via OpenFlow. It governs the routers and switches regarding network traffic. The traffic flow and network layout can be altered quickly using these flow tables by the administrators.

C. Programming SDNs

OpenFlow plays a major role in programming SDNs by rendering direct control over data plane and by providing network wide view at the controller.

Frenetic approach: 1) It is a language that raises the level of abstraction. 2) Capable of handling most of the errors as it is runtime.

III. Analysis and Applications

A. Using SDNs in heterogeneous networks

There are many requirements and challenges in using SDNs in heterogeneous environments which are as follows:

1) Gateway device incentives: Incentives are vital for collaboration in nodes for the gateway devices to transmit data within nodes.

2) Control plane: There are many network components which may not be SDN compatible. It is necessary for such devices to use some protocols like routing. This makes use of a hybrid control plane.

3) Security: This is a very crucial aspect though SDN can be used to enhance traffic policy enforcement and control over the network. Threats like worms and jamming at the application layer should be tackled.

4) Resource discovery: Based on the links and devices they use to interconnect, the infrastructure- less networks tend to become heterogeneous. The factors to be considered while choosing an end device are network connectivity, battery lifetime, trust, etc.

5) Distributed Control Plane: Multiple domains of network are spanned in heterogeneous networks. Hence in an infrastructure less network, it is mandatory for the network to be able to connect on their own to multiple controllers.

6) End-user device limitations: In infrastructure-less network environments like MANETs, there is no difference in end-user devices and network elements like switches, routers etc. which is reverse in case of infrastructure based networks. However the former poses limitations in terms of processing, power, storage, communication, etc.

B. Applications

Software Defined Networking (SDN) developed as a new archetype of networking with which vendors, network operators, owners, to deploy new know-how’s at a very quicker stride.

This can be used exclusively by cellular providers, data centers, homes, enterprises, service providers, etc.

Its usage has spread like a wild fire because of its higher degree of benefits and less complexity. Google has adopted SDN and OpenFlow [12] and even NTT communications [13]. Most of the network vendors have switched on to switches enabled by OpenFlow and SDN [14]. OpenFlow is used exhaustively in wireless mesh environments [15, 16].

SDN is been implemented by many vendors like IBM through OpenFlow. IBM was one of the first members of the ONF. It was the first to adopt OpenFlow 1.0 with an additional functionality in a 10 Gb Ethernet switch. An OpenFlow- based SDn controller, IBM Programmable Network Controller (IBM PNC) provides unlimited virtual machine (VM) mobility and centralized control of network flow. This controller software is capable of continuously and automatically discovering the OpenFlow Network topology and mapping the virtual and physical traffic flows.

Marc Mendonca, et al., [5] have proposed a feasible method of deploying SDNs effectively in heterogeneous environments comprising of mobile end user devices with sparse connectivity to the network; wired –cellular WiFi for instance, but at the rate of security.

Jafar Haadi Jafarian, et al., [17] have introduced the usage of SDNs in OpenFlow Random Host Mutation (OF-RHM) to alter the IP addresses quite often, in order to prevent attackers from encroaching the network. It spoils the unpredictable random mutation of IP addresses.

Georg Hampel, et al., [18] have made a scrutinizing utilization of SDN in telecom domains by showing the usage of SDN in IETF Mobility protocols and in Universal Mobile Terrestrial System (UMTS) and System Architecture Evolution (SAE)

IV. Comparison and Evaluation

Data flow is controlled by switches and routers in traditional network. Each router and switch consists of the following:

Management plane: using which a network administrator logs in for carrying out the desired operations.

Data plane: used to carry data packets from various ports. This is programmed into the device hardware.

Control plane: consists of necessary logic which is vital to program the data plane.

A. Traditional v/s SDN approach to networking

Traditional approach requires the network administrators to be trained with CCNA whereas SDN overcomes this by rendering an easy to use programming interface and many network provisioning apps are available. SDN requires the configuration of SDN controllers, switches and virtualization overlays but installation of cloud middleware is necessary in the prior approaches. There is distinction in the control and data planes in SDN but not in the traditional approach of networking.

B. Impacts

Networking will become software oriented: 1) All complicated data forwarding is done through software. 2) Control plane is a program on a server. 3) It aims at programming the network, not designing it. 4) Innovation is done at software speeds, not hardware. 5) Software lends itself to clean abstractions. 6) Clean abstractions lead to increased rigor.

Networking becomes normal: 1) Software: decoupled from hardware, frequent releases. 2) Hardware: cheap, interchangeable. 3) Functionality: simple, most driven by software.

V. Conclusion

Many network vendors are embracing SDN technology due to its ease of managing virtualized networks, reducing costs, enabling cloud computing, etc. It renders the ability to innovate through software thereby reducing the overhead of hardware maintenance. The architecture involves less complex networks that are easier to manage and are more secure. This aids in reducing the cost, and thus renders low probabilities of software failures, jamming of traffic, and network congestion. The proposed technology provides more bandwidth or fewer hops by controlling specific data flows more efficiently. It simplifies the task of creating private, secure paths for data flows in conjunction to routing the traffic effectively. Many applications which are low cost can be built on the grounds of the ability given by SDN, i.e. security.

Acknowledgement

The authors are immensely grateful to the valuable guidance and support provided by Prof. Geeta Navalyal, Department of Computer Science& Engg, KLE Dr.M.S.S CET, Udyambag, Belgaum and Prof. Shivleela Arlimatti.

References

1. D.L. Tennenhouse, D.Wetherall, “Towards an active network architecture”, ACM Computer Communication Review 37 (5) (2007) 81-94.
2. A.Galis, et al., “A flexible IP active networks architecture”, IWAN, Lecture Notes in Computer Science, vol.1942, Springer, 2000.
3. A.Doria, et al., “Forwarding and Control Element Separation (ForCES) Protocol Specification”, 2010.
4. ONF Market Education Committee, “Software-Defined Networking: The New Norm for Networks”, http://www.opennetworking.org, April, 2012.
5. Marc Mendonca, et al., “Software Defined Networking for Heterogeneous Networks”, IEEE MMTC E-Letters 8, 3 (2013) 36-39.
6. M. F. Mergen, V. Uhlig, O. Krieger and J. Xenidis “Virtualization for high-performance computing”, ACM SIGOPS Operating Systems Review, Vol. 40, April 2006.
7. NOX – An OpenFlow Controller. <http://www.noxrepo.org>.
8. The Future of Networking, and the Past of Protocols. <http://www.opennetsummit.org/talks/shenker-tue.pdf>.
9. M. Kobayashi et al., “Maturing of OpenFlow and Software-defined Networking through deployments”, Computer Networks (2013), http://dx.doi.org/10.1016/j.bjp.2013.10.011
10. N. McKeown, T. Anderson, et al., “Openflow: enabling innovation in campus networks”, ACM SIGCOMM Computer Communication Review, 38(2):69–74, 2008.
11. Nick Mckeown, et al., “OpenFlow :Enabling Innovation in Campus networks”, April 2008, ACM Communications Review.
12. Going With the Flow: Google’s Secret Switch to the Next Wave of Networking, April 2012 <http: // www.wired.com/ wired enterprise/2012/04/going-with-the-flow-google>.
13. NTT COM Announces First Carrier-based OpenFlow Service, June 2012.<http://www.nttcom.tv/2012/06/13/ntt-com-announcesfirst-carrier-based-openflow-service>.
14. Open Networking Summit. <http://www.opennetsummit.org>.
15. A. Coyle and H. Nguyen, “A frequency control algorithm for a mobile adhoc network”, Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, November 2010.
16. P. Dely, A. Kassler, and N. Bayer, “Openflow for wireless mesh networks”, Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), pages 1–6. IEEE, 2011.
17. Jafar Haadi Jafarian, et al., “OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking”, Copyright 2012 ACM 978-1-4503-1477-0/12/08
18. Georg Hampel, “Applying Software-Defined Networking to the Telecom Domain”.

AUTHOR(S) PROFILE

Rahul Dasharath Gavas is currently pursuing his Bachelor of Engineering in Computer Science and Engineering from KLE. Dr. M.S.S. College of Engineering and Technology, Belgaum, Karnataka. His areas of interest include Brain-Computer Interface, Cognitive Computing, Unpredictability theory and Astronomy.

Vijaylakshmi R. completed her Bachelor of Engineering in Electronics and Communication Engineering from HIT, Nidasoshi, India. She completed her M.Tech in VLSI and Embedded Systems from VTU, Belgaum, India. Her areas of interest include VLSI, Embedded, Digital Signal Processing and Computer Networking.