Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Software Defined Networking: A New Frontier in Networking

Jisha MS, Dr. N Radhika
Amrita Vishwa Vidyapeeth, Coimbatore,
Tamil Nadu, India

Abstract - Ever changing business conditions coupled with trends like server virtualization has triggered a considerable demand on the network which could not be handled by today's conservative network architectures. SDN (Software Defined Networking) facilitates a dynamic network architecture transforming conventional network backbones into rich service-delivery platforms. Open Flow-based SDN architecture abstracts the underlying infrastructure from the applications that use it by decoupling the network control and data planes permitting the network to become programmable and manageable. An SDN approach fosters network virtualization, enabling IT staff to manage their servers, applications, storage, and networks with a common approach and tool set. SDNs both simplifies existing applications serve as a platform for developing new ones. OpenFlow switching is already being incorporated into a number of infrastructure designs, both physical and virtual, as well as SDN controller software. OpenFlow design supports policy-based flow management within a network making it particularly well suited to use cases satisfied by pushing predefined policies to implement network segmentation. OpenFlow protocol can be viewed as a key enabler for software-defined networks which is currently the only standardized SDN protocol allowing direct manipulation of the forwarding plane of network devices. Employment of SDN distributed and global edge control also includes the ability to balance load on lots of links leading from the racks to the switching spine of the data-center. With several advantages and astounding business momentum, SDN is becoming the new standard for networks

Keywords— SDN, OpenFlow, ONF, Router, Switch, SDN controller

I. INTRODUCTION

Continuing the conventional approach to networking, all major network functionality is implemented in a dedicated appliance; i.e., router, application delivery controller. Adding to the same, most of the functionality within the dedicated appliance is implemented in dedicated hardware such as an ASIC (Application Specific Integrated Circuit).

Outburst of mobile devices and content, server virtualization, and initiation of cloud services are among the few trends which prompted the networking industry to revisit its conventional network architectures.

Most of the traditional networks are hierarchical in nature built with tiers of Ethernet switches arranged in a tree structure. The design was logical at the time when client-server computing was in prevalence; however, such a static architecture is ill-suited to the dynamic computing and storage requirements of present enterprise data centers and carrier environments.

It is to be noted that, traditional data network was largely hardware-centric. However, the last few years has seen the adoption phase of virtualized network appliances. This, along with the escalating interest in software defined data centers (SDDCs), has lead a movement towards an increased dependence on software-based network functionality. Owing to a wider industry effort lead by the Open Networking Foundation (ONF), Software Defined Networking (SDN) is transforming networking architecture.

II. SOFTWARE DEFINED NETWORKING

As an emerging network architecture, network control in Software Defined Networking (SDN) is decoupled from forwarding and is directly programmable. The group that is most associated with the development and standardization of SDN is The Open Networking Foundation (ONF).

SDN aims to provide open interfaces, which enables development of software that can control the connectivity provided by a set of network resources and the flow of network traffic though them, along with possible inspection and modification of traffic that may be performed in the network.

According to the ONF, the SDN architecture is:

- **Directly programmable:** In SDN, the control plane is decoupled from forwarding functions, which facilitates network control in using higher level programming languages.
- **Responsive:** Administrators are given the facility to adjust network-wide traffic flow due to the abstract control from forwarding.
- **Centrally managed:** Centralization of network intelligence in software based SDN controllers, maintains a global view of the network, which appears to applications and policy engines as a single, logical switch.

- **Programmatically configured:** Automated SDN programs lets network managers configure, manage, secure, and optimize network resources very quickly. As these programs do not depend on proprietary software they can be written by themselves.
- **Open standards-based and vendor-neutral:** As implementation happens through open standards, SDN facilitates network design simplification and operations as the instructions are given by SDN controllers instead of multiple, vendor-specific devices and protocols.

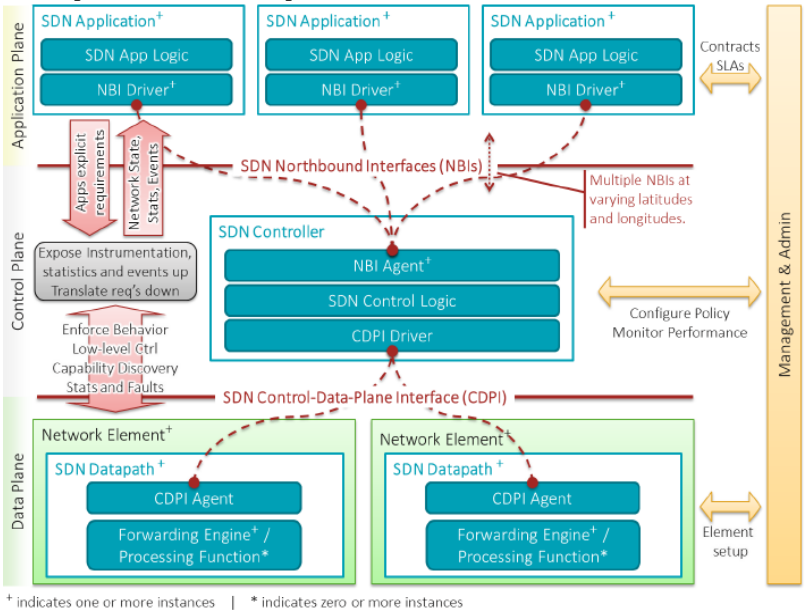

Fig. 1 Software Defined Networking architecture

III. SDN ARCHITECTURE

SDN aims to provide an open interface which enables development of software that controls the connectivity provided by a set of network resources and the flow of network traffic though them, along with possible inspection and modification of traffic that may be performed in the network. A graphical representation of the architectural components and their interactions is depicted in Figure 1.As seen at the bottom, the data plane is comprised of network elements, whose SDN data path exposes their capabilities through the Control-Data-Plane Interface (CDPI) Agent. At the top, SDN Applications exist in the application plane, and communicate their requirements via North Bound Interface (NBI) Drivers. In the middle, these requirements are translated by the SDN Controller and low-level control is exerted over the SDN Datapaths, while providing relevant information up to the SDN Applications. Setting up the network elements, assigning the SDN Datapaths their SDN Controller, and configuring policies defining the scope of control given to the SDN Controller or SDN Application is the responsibility of the Management & Admin plane. Coexisting of this SDN network architecture with a non-SDN network is to migrate to a fully enabled SDN network.

A. Architectural components

1) *SDN Application (SDN App):* These are programs that directly and programmatically communicate their network requirements and desired network behavior to the SDN Controller via NBIs. Adding to the same, they may consume an abstracted view of the network for their internal decision making purposes.
2) *SDN Controller:* This is a logically centralized entity in charge of:
   - Translating the requirements from the SDN Application layer down to the SDN Datapaths.
   - Providing the SDN Applications with an abstract view of the network (which may include statistics and events).

An SDN Controller consists of one or more NBI Agents, the SDN Control Logic, and the CDPI driver. Definition as a logically centralized entity neither prescribes nor precludes implementation details such as the federation of multiple controllers, the hierarchical connection of controllers, communication interfaces between controllers, nor virtualization or slicing of network resources.

3) *SDN Datapath:* This is a logical network device, which exposes visibility and un contented control over its advertised forwarding and data processing capabilities. The logical representation may encompass all or a subset of the physical substrate resources. This Datapath can also be defined across multiple physical network elements. This logical definition neither prescribes nor precludes implementation details such as the logical to physical mapping, management of shared physical resources, virtualization or slicing of the SDN Datapath, interoperability with non-SDN networking, nor the data processing functionality, which can include L4-7 functions.
4) *SDN Control to Data-Plane Interface (CDPI):* This is the interface defined between an SDN Controller and an SDN Datapath, which provides at least:
   - Programmatic control of all forwarding operations,
   - Capabilities advertisement,

- Statistics reporting, and
- Event notification.

5) *SDN Northbound Interfaces (NBI):* SDN NBIs are interfaces between SDN Applications and SDN Controllers and typically provide abstract network views and enable direct expression of network behavior and requirements. This may occur at any level of abstraction (latitude) and across different sets of functionality (longitude).

IV. OPENFLOW

Open flow can be defined as the first standard communications interface between the control and forwarding layers of an SDN architecture. This allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual. An open protocol is provided to program the flow table in different switches and routers. Network administrator can partition traffic into production and research flows. Researchers control their own flows - by choosing the routes their packets follow and the processing they receive, and even alternatives to IP.

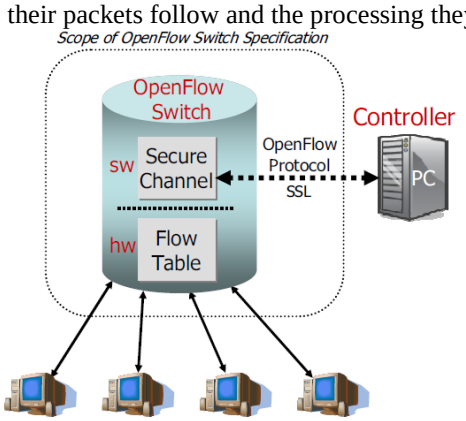

Fig 2 Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

An OpenFlow Switch datapath consists of a flow table, and an action associated with each flow entry. OpenFlow Switch supports set of actions which is extensible. This means forgoing the ability to specify arbitrary handling of each packet and seeking a more limited, but still useful, range of actions.

An OpenFlow Switch consists of minimum three parts:

1) A Flow Table, with an action associated with each flow entry, to guide the switch on how to process the flow,
2) A Secure Channel that connects the switch to a remote control process (called the controller), which allows commands and packets to be sent between a controller and the switch
3) The OpenFlow Protocol, facilitating an open and standard way for a controller to communicate with a switch. By specifying a standard interface (the OpenFlow Protocol) through which entries in the Flow Table can be defined externally, the OpenFlow Switch avoids the need for researchers to program the switch.

MAC src	MAC dst	IP Src	IP Dst	TCP dport	...	Action	Count
*	10:20:.	*	*	*	*	port 1	250
*	*	*	5.6.7.8	*	*	port 2	300
*	*	*	*	25	*	drop	892
*	*	*	192.*	*	*	local	120
*	*	*	*	*	*	controller	11

Fig 3 Flow Table comparable to an instruction set

Comparison can be made between an OpenFlow table and instruction set of a CPU. The former is implemented on both sides of the interface between network infrastructure devices and the SDN control software. The concept of flows is used by OpenFlow, to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed by the SDN control software. In addition it also allows IT to define how traffic should flow through network devices based on parameters such as usage patterns, applications, and cloud resources.

OpenFlow-based forwarding as well as traditional forwarding can be supported by network devices, making it very easy for enterprises and carriers to progressively introduce OpenFlow-based SDN technologies, even in multi-vendor network environments. An enterprise or carrier's existing infrastructure can seamlessly integrate OpenFlow-based SDN architecture to provide a simple migration path for those segments of the network that need SDN functionality the most.

V. CONCLUSION

A new, dynamic network architecture that transforms traditional network backbones into rich service-delivery platforms is provided by Software-Defined Networking. By decoupling the network control and data planes, OpenFlow-based SDN architecture abstracts the underlying infrastructure from the applications that use it, allowing the network to become as programmable and manageable at scale as the computer infrastructure that it increasingly resembles. Even though an

SDN is comprised of many enabling technologies, it cannot be viewed as a technology, but architecture. Whether it is fabric or overlay-based, network virtualization can be viewed as a SDN application. The primary benefit of a network virtualization solution is that it provides support for virtual machine mobility independent of the physical network. SDN, however, has other potential benefits including easing the administrative burden of provisioning functionality such as QoS and security. SDN adoption can improve network manageability, scalability, and agility in a carrier environment or enterprise data center and campus. It promises to transform current static networks into flexible, programmable platforms with the intelligence to allocate resources dynamically, the scale to support enormous data centers and the virtualization needed to support dynamic, highly automated, and secure cloud environments.

ACKNOWLEDGMENT

We make use of this opportunity to express our sincere gratitude to all those who have helped us to complete this work successfully. We wish to place on record our deep sense of gratitude to Dr.Latha Parameswaran, Ph.D., Chairperson, Dept. of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, for her generous guidance, help and useful suggestions. We express our sincere gratitude to Dr.Vidya Balasubramanian, Ph.D., Assoc.Professor, Dept. of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, for her stimulating guidance, continuous encouragement and supervision throughout the course of present work. We also would like to extend our thanks to M.Rithwik and A.K.Sumesh, Dept. of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, and other colleagues for their insightful comments and constructive suggestions to improve the quality of this work. Last but not the least; we would like to express our sincere thanks to our friends for their valuable suggestions and helpful discussions.

REFERENCES

[1] Bernardo ,Chua, et al, "A " *Introduction and Analysis of SDN and NFV Security Architecture (SA-SECA).*" *29th IEEE AINA 2015. pp. 796–801.*

[2] McKeown, Nick, et al, "A "OpenFlow: enabling innovation in campus net works."" *ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.*

[3] Christopher Monsanto, Joshua Reich , Nate Foster , Jennifer Rexford , David Walker, " Composing Software-Defined Networks",in 10th USENIX conference on Networked Systems Design and Implementation.

[4] Myung-Ki Shin , Ki-Hyuk Nam ; Hyoung-Jun Kim, "Software-defined networking (SDN): A reference architecture and open APIs," in ICT Convergence (ICTC), 2012 International Conference.

[5] (2014) Coursera course on SDN [Online]. Available: https://www.coursera.org/ course/sdn.

[6] (2013) ONF website[online].Available: https://www.opennetworking.org.

