iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: I Month of publication: January 2018

DOI: http://doi.org/10.22214/ijraset.2018.1399

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Software Defined Networking:An Understanding

Farheen Fatma Ansari[1,] Prof. D.P.Mishra[2], Prof. Sumit Kumar Sar[3]
[1]M.Tech Scholar) Department of Computer Science and Engineering Bhilai Institute of Technology Durg, India
[2, 3](Associate Professor) Department of Computer Science and Engineering Bhilai Institute of Technology Durg, India

Abstract: Now-a-days, Software-defined networking (SDN) is an ever-growing technology that allows the design and implementation of more flexible and programmable networks. In the past several years, both researchers and vendors have shown tremendous interest on SDN technology. Typical network devices such as switches contain two main components: control plane and data plane. Control plane computes forwarding paths of incoming packets with pre-built network protocols. Data plane sends those packets to the next hop defined in the forwarding paths.
This paper gives a detailed explanation of SDN, its planes and architecture. Also what are the aspects in this field too.
Keywords:Planes,Controllers,Forwarding.

I. INTRODUCTION

Software-defined networking (SDN) technology is an approach to computer networking that allows network administrators to programmatically initialize, control, change, and manage network behavior dynamically via open interfaces and provide abstraction of lower-level functionality. The fact that the static architecture of traditional networks doesn't support all the dynamic, scalable computing and storage needs of more modern computing environments such as data centers as addressed by SDN Technology. This is done by decoupling or disassociating the system that makes decisions about where traffic is sent (the SDN controller, or control plane) from the underlying systems that forward traffic to the selected destination (the data plane). SDN concerns with the OpenFlow protocol (for remote communication with network plane elements for the purpose of determining the path of network packets across network switches) since the latter's emergence in 2011. Since 2012, however, many companies have moved away from OpenFlow, and have embraced different techniques. These include Cisco Systems' Open Network Environment and Nicira's network virtualization platform. SD-WAN applies similar technology to a wide area network (WAN).

II. SOFTWARE DEFINED NETWORKING

Software-defined networking (SDN) is defined as a platform which provides manageable, cost-effective, adaptable, dynamic support and, seeking to be suitable for the high-bandwidth, dynamic nature of today's applications. SDN architectures basically perform decoupling network control and forwarding functions, enabling network control to become directly programmable and the underlying infrastructure to be abstracted from applications and network services.
SDN technology has a flavor of the OpenFlow Protocol.
The SDN architecture is:

A. Directly Programmable

Since it is decoupled from forwarding functions network control is directly programmable.

B. Agile

Abstracting control from forwarding administrators dynamically adjust network wide traffic flow to meet changing needs.

C. Centrally Managed

Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.

D. Programmatically Configured

SDN Technology helps network managers to secure, manage, configure and optimize network resources very calmly, easily, quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software with the help of SDN.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

E. Open standards-based and vendor-neutral

Operation and Network design are simplified by SDN technology because instructions are provided by SDN controllers instead of vendor-specific devices, multiple and protocols as when implemented through open standards.

III. PLANES OF NETWORKING

A. The planes of Networking are

1) *Data Plane-:* It deals with all activities involving as well as resulting from data packets sent by the end-users. It performs:
a) Forwarding
b) \Fragmentation and re-assembly of packets
c) Replication for multicasting

2) *Control Plane:* It deals with all activities that are necessary to perform data plane activities, but do not involve end-user data packets. It performs:
a) Creation of routing tables
b) Setting packet-handling policies

3) *Management Plane:* It deals with all activities related to provisioning and monitoring of the net- work, but it is not necessary for network functioning. Its functionalities include:
a) Fault configuration, accounting, performance and security
b) Instantiation of new devices and protocols. It may turn devices ON/OFF.

4) Service Plane: It provides middle-box services to improve performance, security, etc. It is not required for small networks. It may consist of load balancers, proxy- servers, firewalls, etc.

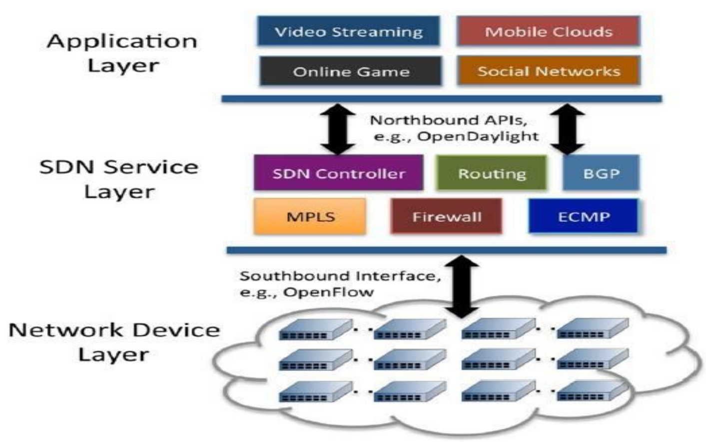
IV. SDN ARCHITECTURE

The core and most attractive idea of SDN is to decouple control planes from physical switches. Software-defined control planes such as SDN controllers run on servers and centralize the management of network devices and data flows.

OpenFlow is the first standard protocol of SDN. OpenFlow defines the southbound interface between software control planes and forwarding planes in SDN architecture. In recent years, OpenFlow-based SDN have been merged with other promising technologies such as cloud computing, mobile clouds, and network virtualization.

A typical SDN architecture is shown in Figure. The underlying layer contains hardware devices such as OpenFlow-enabled switches. SDN service layer provides software-defined network functions including routing, firewall, multiprotocol label switching, equal cost multi-path, and etc.

Underlying switches connect SDN controllers through a northbound interface defined by OpenFlow. SDN controllers such as Open daylight, Ryu, and Floodlight provide APIs for upper layer applications.



V. ARCHITECTURAL COMPONENTS

A. The following list defines and explains the architectural component

1) *SDN Application (SDN App):* SDN Applications are the basic programs that explicitly, directly, and programmatically communicate their network requirements and desired network behavior to the SDN Controller via a northbound interface (NBI). In addition, they provide us with an abstracted view of the network for their internal decision making purposes.SDN

Application consists of one SDN Application Logic and one or more NBI Drivers. SDN Applications provide another layer of abstracted network control, thus offering one or more higher-level NBIs through respective NBI agents.

2) *SDN Controller*: The SDN Controller is a logically centralized entity in charge of

translating the requirements from the SDN Application layer down to the SDN Data paths, and

providing the SDN Applications plus an abstract view of the network (which may include statistics and events).

SDN Controller consists of various types of the SDN Control Logic, the NBI Agents and the Control to Data-Plane Interface (CDPI) driver of one or more in number. It is defined 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 Northbound Interfaces:* SDN NBIs refers to the 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). One most important value of SDN lies in the fact that the expectation that these interfaces are implemented in a vendor-neutral, an open and interoperable way.
4) *SDN Southbound Interfaces:* The SDN SBI is the interface defined between an SDN Controller and network devices, which provides at least

a) programmatic control of all forwarding operations,
b) capabilities advertisement,
c) statistics reporting, and
d) event notification. One value of SDN lies in the expectation that the SBI is implemented in an open, vendor-neutral and interoperable way.

VI. ADVANTAGES OF SDN ARCHITECTURE

Some advantages of SDN-based network architecture are:

A. Network operators will be able to configure network devices using soft- ware programs, rather than type configuration commands on thousands of devices manually
B. SDN controllers provide a better global view of network topology. It also accelerates the updates of network functions.
C. With SDN technology, it is possible to replace expensive and high throughput switches with large numbers of commoditized switches. Such infrastructure is capable of supporting higher throughput and more flexible flows with less capital and operational expenditure.

VII. SDN FLOW FORWARDING

OpenFlow uses TCAM tables to route packet sequences (flows). If flows arrive at a switch, a flow table lookup is performed. Depending on the flow table implementation this is done in a software flow table if a vSwitch is used or in an ASIC if it's implemented in hardware. In the case when no matching flow is found a request to the controller for further instructions is sent. This is handled in one of three different modes. In reactive mode the controller acts after these requests and creates and installs a rule in the flow table for the corresponding packet if necessary. In case of proactive mode all the controller populates their flow table entries for all possible traffic matches possible for this switch in advance. This mode can be compared with typical routing table entries today, where all static entries are installed ahead of time. Following this criteria, no request is sent to the controller as all incoming flows will find a matching entry. A major advantage in proactive mode is that all packets are forwarded in line rate (considering all flow table entries in TCAM) and no delay is added. The third mode, hybrid mode, follows the flexibility of a reactive mode for a set of traffic and the low-latency forwarding (proactive mode) for the rest of the traffic.

VIII. ASPECTS

A. The two aspects are

1) Number of Controller
2) Position of each controllers

Pastly,is extended with resilience aspects

Nextly,is done with load balancing among Controllers and inter controller latencies.

Presently, these both are bundled together.

IX. CONCLUSION

Due to its innovation potential, SDN is seen as one key technology to enable and operate next generation networks. However, the term SDN currently exist, leading to a fragmented view.

As we studied, Software defined networking (SDN) has emerged as a new paradigm that shifts network control from distributed protocols to a logically centralized control plane.

Lastly, this paper reflects better understanding of SDN.

X. ACKNOWLEDGMENT

We would like to acknowledge the contribution of all the people who have helped in reviewing this paper. We would like to give sincere thanks to Mr. Sumit kumar sar for his guidance and support throughout this paper. We would also like to thank our families and friends who supported us in the course of writing this paper.

REFERENCES

[1] M. Jarschel, T. Zinner, T. Hoßfeld, P. Tran-Gia, and W. Kellerer, “Interfaces, Attributes, and Use Cases: A Compass for SDN,” IEEE Communications Magazine, 2014.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” SIGCOMM CCR, 2008.

[3] A. Tootoonchian and Y. Ganjali, “HyperFlow: a Distributed Control Plane for OpenFlow,” in INM/WREN’10, Berkeley, CA, USA, 2010.

[4] B. Heller, R. Sherwood, and N. McKeown, “The Controller Placement Problem,” in HotSDN ’12, New York, NY, USA, 2012.

[5] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran- Gia, “Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks,” in 25th International Teletraffic Congress (ITC), 2013.

[6] D. Hock, M. Hartmann, S. Gebert, T. Zinner, and P. Tran-Gia, “POCOPLC: Enabling Dynamic Pareto-Optimal Resilient Controller Placement in SDN Networks,” INFOCOM, Toronto, Canada, 2014.

[7] Stanislav Lange, Steffen Gebert, Thomas Zinner, Phuoc Tran-Gia, David Hocky, Michael Jarschelz, and Marco Hoffmannz,’’ Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks,’’ University of Wurzburg, Institute of Computer Science, Chair of Communication Networks, Wurzburg, Germany,2015.

.