

Software-Defined Network - A New Approach

Avani P. Patel

Assistant Professor

*Department of Computer Engineering
GIDC Degree Engineering College, Navsari*

Vibhuti P. Patel

Assistant Professor

*Department of Computer Engineering
GIDC Degree Engineering College, Navsari*

Mihir A. Mishra

Assistant Professor

*Department of Computer Engineering
GIDC Degree Engineering College, Navsari*

Kushal D. Patel

Assistant Professor

*Department of Computer Engineering
GIDC Degree Engineering College, Navsari*

Pragnesh A. Patel

Assistant Professor

*Department of Computer Engineering
GIDC Degree Engineering College, Navsari*

Abstract

The fundamental traditional network is static in static thus it does not provide the user the facility to control the network as well as new innovation window becomes too narrow with respect to security, traffic control, cost efficient, automation, etc. Thus a new approach that is Software-Defined Network provides all this facilities and is programmable controlled by the user. This paper provides a review of the Software Defined Network and helps us in clearly understanding the new approach by studying the history of SDN and its applications in today's real life. It also helps in understanding the motive of the SDN.

Keywords: Open Flow, Dynamic, Scalability, Programmability, Agility, Multi-Tendency, Control Plane, Data Plane

I. INTRODUCTION

SDN is a framework that allows the network administrator to dynamically manage and control the network, traffic flow, services and packet handling using high level languages and APIs. Software Defined network (SDN) is an approach which allows the network administrator to manage network services through high level of abstraction. Software-Defined Networking is an architecture that is cost efficient, dynamic in nature; easily manageable making it ideal for high bandwidth. This architecture gives the control in the hand of the programmer by decoupling the network control and forwarding functions. The Open Flow protocol is a foundational element for building SDN solutions. The present network architecture consist of various hardware like routers, firewalls and switches and we can see all of them together as large complex system and thus a minute change at any level may cause network failure. Thus it is better to move on to such architecture that has a centralized control, easy to manage, dynamic and highly scalable.

II. COMPARISON OF SDN WITH TRADITIONAL NETWORK

In current network the data flow is controlled by switches and routers. Current network architecture is static if compared to SDN. Users don't have any control over packets. Generally this job is done by switches whereas in SDN, user can control his own traffic because of the centralize control mechanism. The basic difference between traditional and software defined network is that of handling the packets. In traditional network the way packet is to be handled is written in its firmware. Whereas in SDN it provides admin with granular control over the way switches handle data, giving them the ability to automatically prioritize or block certain types of packets. This allows greater efficiency as well as is less expensive as there is no need of expensive application specific network switches.

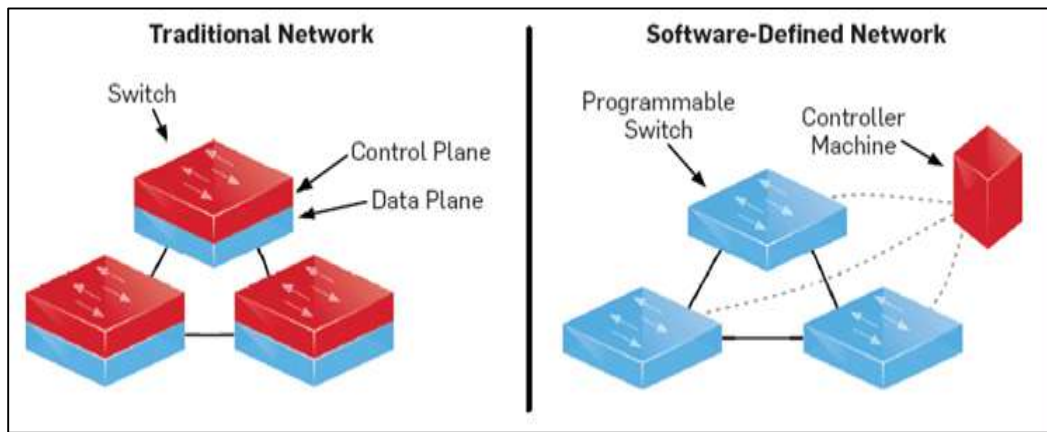


Fig. 1: Traditional Network v/s software Defined Network

The SDN architecture is:

- Directly programmable: Network control is directly programmable because it is decoupled from forwarding functions. Thus control is in the hands of the programmer.
- Agility: Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.
- Centrally controlled: Here the network is centralized this means the SDN maintains a global view of the network which appears as a single logical switch.
- Programmatically configured: the program does not depend on priority software so the network manager can configure, manage, secure and optimize network resources very quickly.
- Open standards-based and vendor-neutral: When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

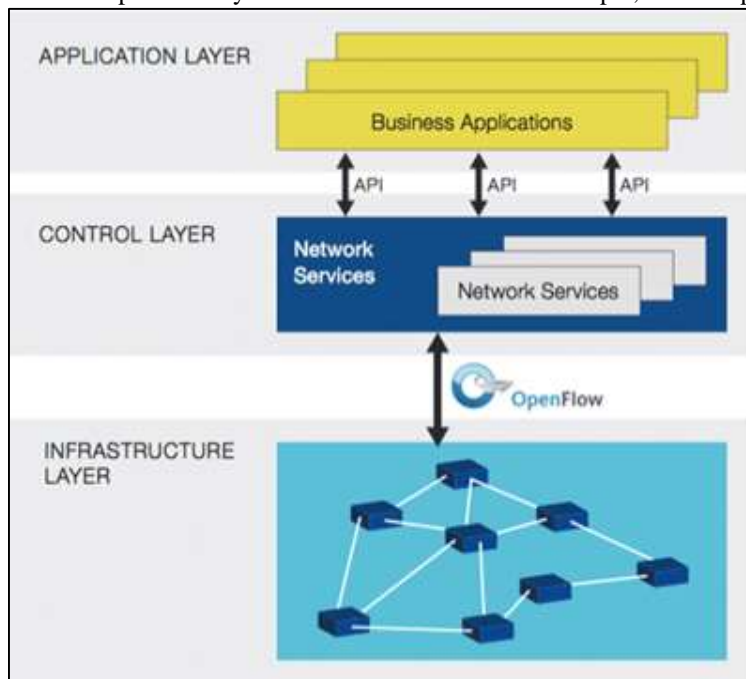


Fig. 2: SDN Architecture

III. NEED OF SOFTWARE DEFINED NETWORK

- 1) Virtualization: we should be able to use the network without worrying about its physical location and how it is organized.
- 2) Orchestration: the network should be able to manage and control thousands of devices with just one command.
- 3) Programmable: one should be able to control the behaviour as well as the network traffic.
- 4) Dynamic Scaling: size and quantity should be changeable as and when required.
- 5) Automation: It provides minimal manual
- 6) Visibility: Monitor resources, connectivity
- 7) Performance: It optimizes the resource utilization.

- It manages the Bandwidth.
 - Capacity optimization
 - Load balancing
 - Failure handling is fast.
- 8) Multi-tenancy: Controller must have complete control of topology, their addresses and routing.
- 9) Service Integration: It provides facility to place firewalls, Intrusion Detection System on the traffic paths according to our choice.
- 10) Openness: full choice of how the mechanism works.
- Modular plug-ins
 - Abstraction
 - Details should be hidden.
 - Define tasks by APIs and not by how it should be done.

IV. ARCHITECTURAL COMPONENTS

Following are the list of components of Software-Defined Network:

A. SDN Application (SDN App)

SDN Applications are programs that communicate directly and programmatically communicate their network requirement and their network behaviour to the SDN controller through a Network Bound Interface (NBI). In addition they may consume an abstracted view of the network for their internal decision making purposes. An SDN Application consist of Application Logic and one or more NBI Drivers.

B. SDN Controller

The SDN controller is the central control in charge of (i) translating the requirements from the SDN Application layer down to the SDN Data paths and (ii) providing the SDN Applications with an abstract view of the network. An SDN controller consists of SDN Central Logic, CDPI driver and one or more NBI Agents.

C. SDN Data Path

SDN Data path is a logical network that exposes visibility and control over the forwarding and data processing capabilities. It consists of traffic processing functions and traffic forwarding engines. These engines and functions may include simple forwarding between the data path's external interfaces or internal traffic processing or termination functions. One or more data paths may be managed as a single unit. An SDN Data path may also be defined across multiple physical network elements. This logical definition neither includes nor excludes implementation details like mapping of logical to physical, managing shared physical resources, interoperability with non-SDN Network, nor data processing functions that include OSI layer

D. SDN Control to Data-Plane Interface (CDPI)

The SDN CDPI is the interface defined between an SDN Controller and an SDN Data path, that provides at least control of all forwarding operations programmatically, control of reporting of statistics and notification of an event.

E. SDN Northbound Interfaces (NBI)

SDN NBIs are interfaces between SDN Applications and SDN Controllers and provide network views and direct expression of network behaviour and requirements. These interfaces are implemented in an open, vendor-neutral and interoperable way.

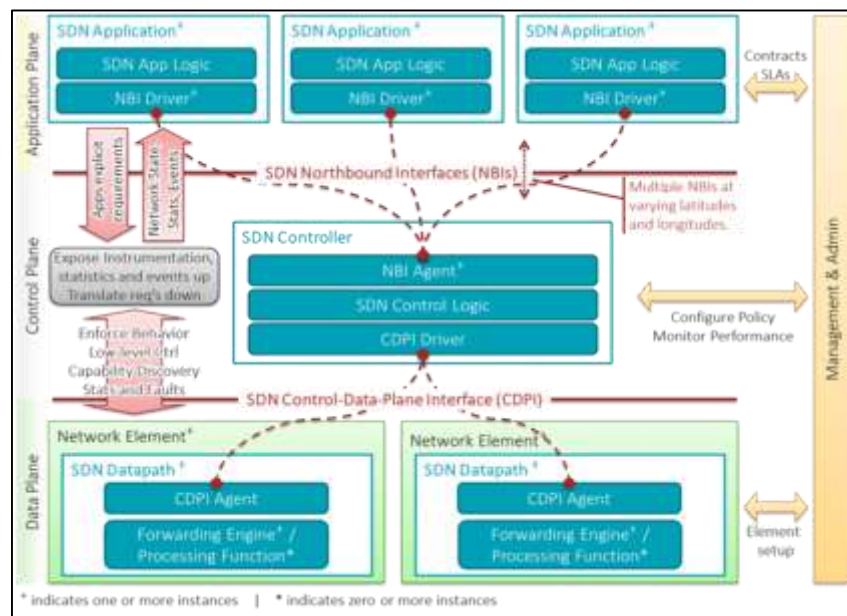


Fig. 3: A high level overview of software-Defined network Components

V. APPLICATION OF SOFTWARE DEFINED NETWORK

- Anywhere access to information via the cloud
- Network management
- Resource utilization
- Security
- Network Functions Virtualization (NFV)
- Network service chaining
- Faster innovation
- Bandwidth calendaring
- Network programmability
- Energy management

VI. CONCLUSION

The traditional network cannot satisfy the user's requirement efficiently. SDN provides high security, high automation, highly reliability and thus SDN is a more promising architecture. SDN also provides user a feeling of nearness to the network. SDN gives opportunities to implement different security strategies as it is capable of providing direct and fine-grained control over the network.

REFERENCES

- [1] K. Greene. 10 Emerging Technologies, TR10: Software Defined Networking, 2009. Available at:.
- [2] B. Lantz, B.Heller, and N. McKeown. A network in a laptop: rapid prototyping of Software-Defined Network. In Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, pages 19:1-19:6, New York, NY,USA,2010.
- [3] S. Costanzo, L. Galluccio, G. Morabito, S. Palazzo. "Software Defined Wireless Networks: Unbridling SDNs". European Workshop on Software Defined Networking, 2012.
- [4] <http://www.zdnet.com/10-key-questions-about-Software-Defined-networking-sdn-700001582>
- [5] <https://en.wikipedia.org/wiki/Software-defined-networking>
- [6] <https://www.opennetworking.org/sdn-resources/sdn-definition>
- [7] https://wiki.opendaylight.org/view/yang_Tools:Main
- [8] Vijay K. Garbani, M Scharf, T.V. Lakshman and V.Hilt. Bell Laboratories, Alcatel-Lucent " Abstracting network state in Software Defined Networks(SDN) for rendezvous services". "Communications (ICC), 2012 IEEE International Conference on Software Defined Networks. (2012).
- [9] <http://searchcloudprovider.techtarget.com/tip/Three-models-of-SDN-explained>
- [10] Fi'avio de OLiveria Silva, Jo'ao Henrique de Souza Pereir, Pedro Frosi Rosa, and Sergio Takeo Kofuji. "Enabling Future Internet Architecture research and Engineering by using software Defined Networking." European Workshop on Software Defined Networking, 2012.
- [11] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," SIGCOMM Comput. Commun. Rev., vol. 35, no. 5, p. 41–54, Oct. 2005. [Online]. Available: <http://doi.acm.org/10.1145/1096536.1096541>
- [12] Furqan Alam, Iyad Katib and Saeed Alzahrani, "New Networking Era: Software Defined Networking", Indian Journal Of Advanced research in Computer Science and Software Engineering, volume 3, Issue 11,November 2013.