Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018)
IEEE Xplore Compliant - Part Number:CFP18J06-ART, ISBN:978-1-5386-0807-4; DVD Part Number:CFP18J06DVD, ISBN:978-1-5386-0806-7

Software Defined Network: Future of Networking

Arpita Prajapati, Achyut Sakadasariya, Jitisha Patel *Computer Engineering and Information Technology Department, CGPIT, Bardoli*, India.
arpita.prajapati03@gmail.com, achyut.sakadasariya@utu.ac.in, jitisha.patel@utu.ac.in,

***Abstract*—The digital world we live in has been lead by the evolution of Internet, which has created revolution. Today almost everything is connected and accessible from anywhere, everywhere. Unfortunately, the traditional IP network system is still complex, not easily manageable and difficult to reconfigure in case of any change or fault. Software Defined Networking (SDN) is an emerging paradigm separates the network's control logic from the underlying routers and switches, promoting logical centralization of network control and introducing the ability to program the network. It has become the focus in the current information and communication technology area because of its flexibility and programmability. SDN abstracts low-level network functionalities to simplify network management. The OpenFlow protocol implements the SDN concept by abstracting network elements. It offers flexible and scalable functionality for network by decoupling the network control from forwarding devices. SDN uses a REST API (Representational State Transfer) for communication between controller and another application.**

***Index Terms*—Software defined networking, network virtualization, network operating systems, OpenFlow, REST API**

I. INTRODUCTION

Since networks have been growing huge in size and requirements, moving around hardware switches has become a burden. Even manual setting up of individual software switches has become a complicated and error-prone task for companies running strongly virtualized environments along with their large networks. Automatic reconfiguration and response mechanisms are virtually non-existent in current IP networks. Enforcing the required policies in such a dynamic environment is therefore highly challenging. Moreover, current networks are also vertically integrated meaning the control plane (that decides how to handle network traffic) and the data plane (that forwards traffic according to the decisions made by the control plane) are packed inside the networking devices. This reduces flexibility and does not allow innovation and evolution of the networking infrastructure. This is where Software Defined Network (SDN) comes to shorten and ease the game. Software-Defined Network (SDN) [1] [2] is a new approach to the current world of networking, that gives hope to overcome the limitations of current network infrastructures. It removes the vertical integration by separating the network's control logic (the control plane) from the traffic forwarders (underlying routers and switches in the data plane). This separation of the control and data planes makes network switches, a simple forwarding devices and the control logic is implemented in a logically centralized controller, also known network operating system, simplifying policy enforcement and network reconfiguration and evolution [3]. In this paper, II section is about traditional v/s SDN , III section is about literature survey. In this section we overview about SDN architecture and their components. In next section we show the comparison of simulation tools for SDN implementation.

II. TRADITIONAL v/s SDN

Traditional networking uses a distributed model for the control plane. Protocols like ARP, STP, OSPF, EIGRP, BGP and other run separately on each network device [4]. These network devices communicate with each other but there is no central device that has an overview or that controls the entire network.

SDN uses a centralized model for the control plane. SDN controller resides in control plane and switches are in data plane. SDN controller is responsible for feeding the data plane with the information of its control plane. Protocols like OpenFlow, BGP (Border Gateway Protocol), OVSDB (Open vSwitch Database Management Protocol, XMPP (Extensible Messaging and Presence Protocol), NETCONF, MPLS-TP (Multiprotocol Label Switching-Transport profile) [5].

The control plane (where and how to forward the packet), and the data plane (the physical forwarding of the packet onto the wire). Figure 1 shows that traditional network is the type of network that is more prone to failure due to multiple disconnected brains (controller) not working together with one another. In such a way that is easily scriptable and configurable in comparison of network such as SDN that is open and connectable through flows to central controllers on the software layer.

If both the planes are placed in one physical device then two processes have to share the resources, increasing the traffic load and burden on CPU and memory. Figure1 shows that separation of data plane and control plane in SDN. By separating these processes and having a dedicated server, controlling and monitoring the control plane and network's ability to make efficient routing decisions become easy and also enable network to properly configure with less traffic

load.

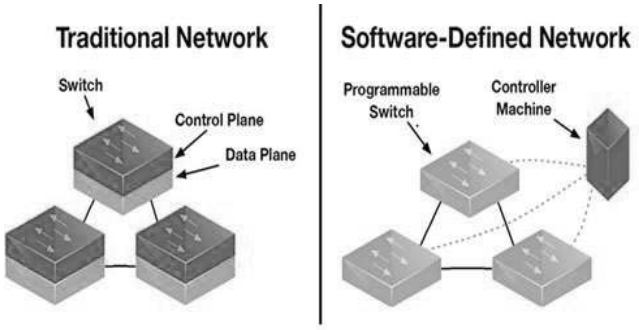

Figure 1: Traditional network v/s SDN [6]

III. LITERATURE SURVEY

A vastly increment of devices in internet, management and configuration have become complex and time consuming. Also it is difficult to access the high bandwidth, extendibility and flexibility when networking devices are vertically integrated. To overcome this issue, one must break the vertical integration of networking devices. It is called Software Defined Network (SDN). Inspired by the words of Marc Andreesen, ‘‘software is eating the world’’, SDN and virtualization are poised to be the solutions that overcome the challenges like extendibility, flexibility and agility of network. The main idea behind SDN is to separate the forwarding/data plane from the control plane while providing programmability on the control plane. SDN focuses on centralized solution, where a single control entity has a global view of the network. This simplifies the implementation of control logic, management and configuration.

A. SDN Architecture

Figure 2 shows the SDN architecture, where 3 layers: 1) Data layer, 2) Control layer and 3) Application layer are communicating with the help of interfaces. Layers communicate using Southbound interface and Northbound Interface. Southbound interface is between data layer and control layer and Northbound interface is between control layer and application layer. Detailed description about all components of SDN architecture is as below:

Data Layer

The data layer incorporates the resources that deal directly with customer traffic, along with the necessary supporting resources to ensure proper virtualization, connectivity, security, availability and quality [7]. It is composed of various networking equipment which forms underlying network to forward network traffic [8]. It is responsible for handling data packets based on instructions which are given by controller. Actions performed in data plane are forwarding, modifying and dropping the packet [9].

Control Layer

The control layer is responsible for making decision on how packets should be forwarded by one or more devices [8]. It is an intelligent logic for SDN controller where all implementation logic is done. SDN controller is for managing the network, so it must have control logic for switching, routing, firewall security rules, clustering, etc [9].

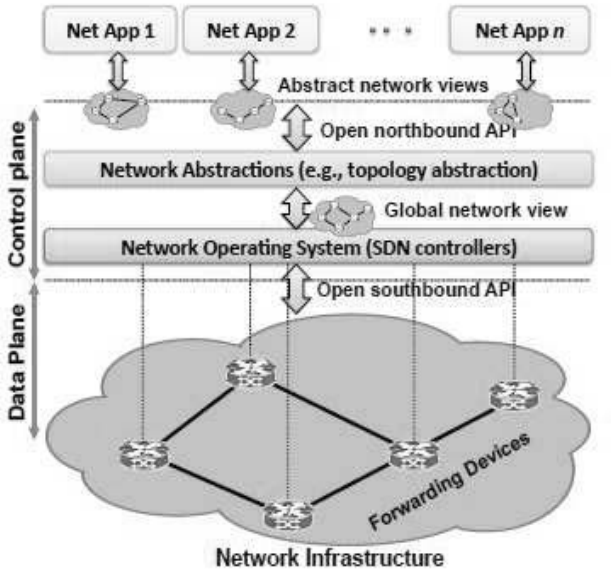

Figure 2: SDN Architecture [10]

Control layer functionality:

- Topology discovery and maintenance
- Packet route selection and instantiation
- Path failover mechanisms

There are different types of controllers which are used by control plane for different features as shown table 1.

Table 1: Controller’s features [11]

Controller name	NOX	POX	RYU	Floodlight	OpenDay-light
Language	C++	Python	Python	Java	Java
Performance	High	Low	Low	High	High
Distributed	No	No	Yes	Yes	Yes
OpenFlow support version	1.0	1.0	1.0-1.4	1.0	1.3
Cloud support	No	No	Yes	Yes	Yes
Learning	Medi-um	Easy	Mediu -m	Hard	Hard

Application Layer

The application layer comprises one or more applications, each of which has exclusive control of a set of resources exposed by one or more SDN controllers [8]. Applications and services that use the services from control plane with the help of Northbound interface [7]. It is an open area for developing innovative applications by using network information about network topology, statics, network state, etc. There are number of applications developed with use of network information like those who relates to network automation, network management, network configuration and security [9].

Southbound Interface

Southbound interfaces are APIs that enables the communication between control plane and data plane. OpenFlow protocol is well known protocol for southbound interface. Southbound interface provides (i) programmatic control of all forwarding operations, (ii) capabilities advertisement, (iii) statistics reporting, and (iv) event notification [7].

OpenFlow is not the only available southbound interface for SDN; there are also other southbound interfaces such as Open vSwitch database management protocol (OVSDB), forwarding and control element separation (ForCES), protocol oblivious forwarding (POF), OpFlex control protocol, OpenFlow Config (OF-Config), OpenState, revised OpenFlow library (ROFL), hardware abstraction layer (HAL) and Programmable Abstraction of Datapath (PAD) [12].

Northbound Interface

It is meant for communication with upper, Application layer and would be in general realized through REST APIs of SDN controllers [9]. It provides abstract network views and enables direct expression of network behavior and requirements. This may occur at any level of abstraction (latitude) and across different sets of functionality (longitude).Open and standard northbound interfaces are crucial to promote application portability and interoperability among the different control platforms. It is also used to integrate the SDN Controller with automation stacks, such as Puppet, Chef, SaltStack, Ansible and CFEngine, as well as orchestration platforms, such as OpenStack, VMware's vCloudDirector or the open source CloudStack [13].

IV. TOOLS FOR IMPLEMENTATION

Several simulation tools such as the OMNET++ and Mininet have been developed to evaluate the performance of SDN. The other tools for simulation are Ns-3 and Estinet. These tools have their own features. The comparison between different simulations tools have been shown in Table 2 [13].

OMNET++ and Mininet are suitable for designing, building, and testing and provide practical feedback when developing real world systems. OMNET++ offers useful facilities for prototyping and simulating SDN application. Mininet is an emulation platform for the functional testing of OpenFlow protocol and SDN application. OMNET++ and NS-3 cannot support real controller. Mininet is less scalable compare to NS-3, Estinet and OMNET++. Estinet combine all the functionality and overcome the problem of real controller and emulation. Estinet is also capable for simulation to large number of switches. Estinet generates the accurate result and results are repeatable.

Table 2: Comparison of Simulation Tools [13]

	NS-3	**OMNET ++**	**Mininet**	**Estinet**
Supporting Simulation	Yes	Yes	No	Yes
Supporting Emulation	No	No	Yes	Yes
Ability to use real controller	No	No	Yes	Yes
Result repeatable	Yes	Yes	No	Yes
Performa-nce result correctness	No spanning tree protocol & no real controller	No real controll-er	Performance depend on resources	Yes
GUI support	Monitoring only, Configurati-on by C++	Configua ration, Monitori -ng	Monitoring only, Configurati-on by Python	Configurati-on, Monitoring

V. CONCLUSION AND FUTURE SCOPE

Incremental growth of internet is hard to manageable and configurable by traditional network. Based on functionality like scalability, flexibility, easy to configure the network, SDN is more preferable compare to traditional network. SDN is most promising network for performance where large numbers of devices are there and heavy load of network is there.SDN aims to simplify the network with the using of centralized architecture for switching and routing operations.

Most of the functions are done by SDN but major concern about security is there. Security of network is as much as important for devices. Security problem is in SDN because of centralized architecture. So, this will overcome by stronger architecture.

VI. REFERENCES

[1] N. Mckeown, "How SDN will Shape Networking", October 2011. [Online]. Available: http://www.youtube.com/watch?v=c9-K5OqYgA.

[2] S. Schenker, "The Future of Networking, and the Past of Protocols", October 2011. [Online]. Available: http://www.youtube.com/watch?v=YHeyuD89n1Y4

[3] H. Kim and N. Feamster, "Improving network management with software defined networking", Communications Magazine, IEEE, vol. 51, no. 2, pp.114–119, 2013.

[4] Rene Molenaar, "Introduction to SDN (Software Defined Networking)", 2013-2017.[Online].Available:https://networklessons.com/cisco/ccna routing switching-icnd2-200 105/introduction-to-sdn-software-defined-networking.

[5] Michelle McNickle, "Five SDN protocols other than OpenFlow", 2014. [Online].Available:http://searchsdn.techtarget.com/news/2240227714/Five-SDN-protocols other-than-OpenFlow.

[6] Software Defined Network, 2017. [Online].Available: https://www.datacomm.co.id/en/telco/sdn/.

[7] Stuart Bailey, Deepak Bansal, Linda Dunbar, Dave Hood, Zoltán Lajos Kis, Ben MackCrane, Jeff Maguire, Dan Malek, David Meyer, Manuel Paul, Sibylle Schaller, Fabian Schneider, Rob Sherwood, Johann Tonsing, Tina Tsou, Eve Varma, "SDN Architecture Overview", Open Networking Foundation,Version 1.0, December 12, 2013.

[8] Malcolm Betts, Nigel Davis, Rob Dolin, Paul Doolan, Steve Fratini, Dave Hood, Mandar Joshi, Kam Lam, Ben Mack-Crane, Scott Mansfield, Pascal Menezes, Sriram Natarajan, Manuel Paul, Makan Pourzandi, Jennifer, Jonathan Sadler, Sibylle Schaller, Fabian Schneider, Silvio Shefer, Peter Smith, Jean Tourhilles, Tina Tsou, Eve Varma, Maarten Vissers, Marc Woolward, Sachs Zhang Dacheng, “SDN Architecture Overview”, Open Networking Foundation, Issue 1, June 2014.

[9] Himanshu Arora, “Software Defined Networking (SDN) – Architecture and role of OpenFlow”, June 2017. [Online]. Available:https://www.howtoforge.com/tutorial/software-defined-networking-sdn architecture-and-role-of-openflow/.

[10]Diego Kreutz, M. V.Ramos,Verissimo,Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig: "SDN-A comprehensive survey.", IEEE, 2014.

[11]Igor Godanj, Kresimir Nenadic, Kresimir Romic: “Simple Example of Software Defined Network”, IEEE, 2016.

[12]Rahim Masoudi, Ali Ghaffari: “Software Defined Networks: A survey”, Journal of Network and Computer Applications, 2016.

[13]SDx Central, ”What are the Northbound API”, 2012-2017. [Online].Available:https://www.sdxcentral.com/sdn/definitions/north-bound-interfaces-api/.

.