Volume: 07 Issue: 02 Pages: 2690-2693 (2015) ISSN: 0975-0290

SDN and Mininet: Some Basic Concepts

Pooja
Department of Computer Science, Himachal Pradesh University, Shimla
Email: pooja.s3390@gmail.com
Manu Sood
Department of Computer Science, Himachal Pradesh University, Shimla
Email: soodm_67@yahoo.com

ABSTRACT
As computer network grow larger and more complex, there is a need for a new simple kind of approach to configure them. SDN has emerged as promising network architecture. It takes the control plane away from the individual nodes and centralize the network control by utilizing a flow based traffic management. Mininet is a cost effective and an efficient way to emulate and study SDN.This paper presents a study of programmable networks with basics of Mininet.
Keywords**: Architecture, Mininet, SDN, Traditional network.**

Date of Submission: July 02, 2015 Date of Acceptance: Aug 28, 2015

I.INTRODUCTION

Internet applications require networks to be fast, carry large amounts of traffic and to deploy a number of distinct, dynamic applications and services. Controlling and managing networks has become a highly complex activity and concepts like network virtualization and interchangeable data has added more difficulty to the network administrator. The traditional networks are inflexible. Once the forwarding policy has been defined, the only way to change it is by changing the configuration of all the connected devices. This is time consuming and limited on scalability, mobility and big data but, network are adapt to changes, updating after sometime.
In this context Software Defined Networking (SDN) is being looked upon as a way that has the power to change the traditional networking. In SDN architecture, the control and data planes are separated, controller is logically centralized and the underlying network infrastructure is abstracted from the applications [1].

II. TRADITIONAL NETWORK AND SDN

In the **traditional approach** to networking, most network functionality is implemented in a dedicated appliance; i.e., switch, router, application delivery controller which are configured manually [2]. Also, most of the functionality is implemented in hardware such as an ASIC (Application Specific Integrated Circuit). Data flow is controlled by routers and switches. A traditional network architecture as shown in Fig.1, consists of a data plane which carries data physically in form of packets from one node to the other by following certain protocols, control plane which consist of logic that devices are using therefore to forward packets over the network then comes Management plane which behave as an administrator [3]. The design principles behind them is not matching today's rapid data growth, bandwidth, speed, security and scalability issues. In research papers [4], [5] a comparative study of traditional networks and SDN has been done.

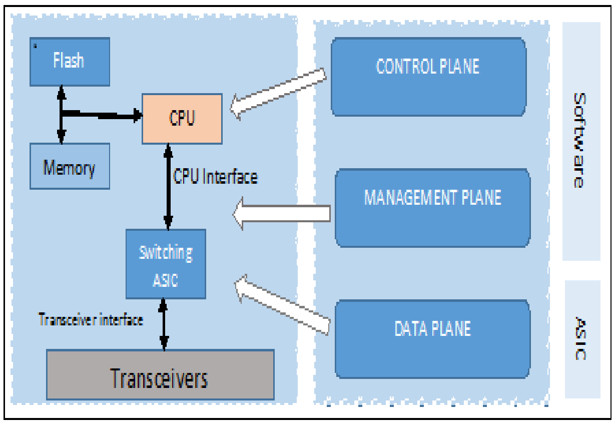


Fig 1. Traditional Switch Architecture [6]

In **Software Defined Networking (SDN)** [7] approach, the physical separation of the network control plane from the forwarding plane is done and where the control plane controls several devices. The Open Networking Foundation (ONF)is the group which is associated with the development and standardization of SDN.According to the ONF [8], "Software-Defined Networking (SDN) is an architecture that is dynamic, manageable, cost-effective and programmable making it ideal for the high-bandwidth, dynamic nature of today's applications". SDN is developed to enable simple programmatic control of the network data-path. Refer fig. 2 for more detailed SDN design concept. SDN focuses on four key features [9]:

- Separating control plane from the data plane.
- Centralized controller of network.
- Open interfaces between the devices in the control plane and the data plane.
- Programmability of the network by external applications.

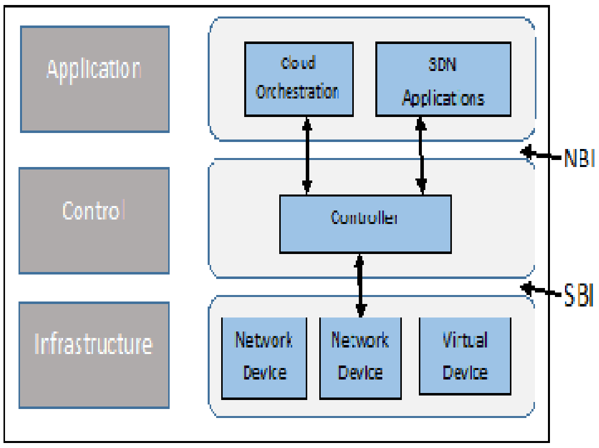


Fig 2: SDN Architecture [10]

The SDN architecture [10]separates the networking devices into data plane and control plane. Data plane perform forwarding packets fast and efficiently. It carries data packets from one port to another by following the rules which are programmed into device hardware. Control plane adds behavior into the device and it decides the logic to program the data plane. These two planes (data and control) communicate over a secure channel over a standard protocol called OpenFlow [5].Northbound Interfaces (NBI) are the interfaces between SDN Applications and SDN Controllers and Southbound Interface (SBI) are interfaces between SDN Controller and Network devices.

SDN supportsvirtualization [11] which makes it more scalable than traditional network. There are various controllers for SDN such as NOX [12], POX [3], Helios [13] and Floodlight [14]. The choice of particular controller depends upon the experience and knowledge of networks operators. SDN is fully programmable and whole intelligence lies in the centralized controller. Programming languages[2]like Frenetic, FML, Procera, Flog and Pyretic are used to code for the controllers.

III. MININET AND ITS BASICS

Mininet [15] is a network emulator which runs collection of end-hosts, switches, routers and links on a single Linux kernel. It is an approach of using operating system virtualization features, and processes toallow it to scale up to hundreds of nodes. Users can implement a new network feature or a new architecture, test it on large topologies with application traffic and then deploy the same code and test scripts into a real production network [16].

Mininet was created by a group of professors at Stanford University to be used as a tool to research and to teach network technologies. Now it is designed to easily create virtual SDN consisting of an Open-Flow controller, a flat Ethernet network of multiple Open-Flow enabled Ethernet, switches and multiple hosts connected to those switches. It has built-in functions that support using different types of controllers and switches [17].Fig. 3 shows a diagrammatical representation of emulating hardware using mininet.

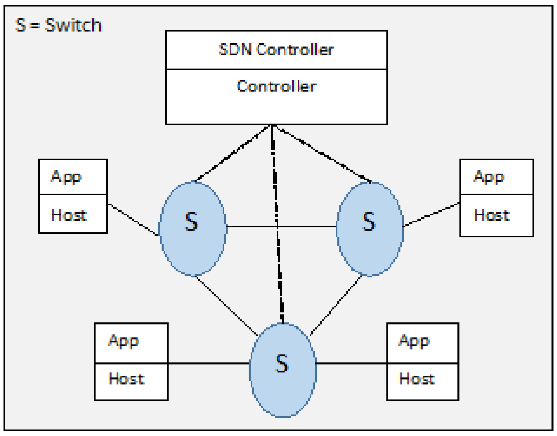


Fig.3: Emulating hardware network using Mininet [7]

Mininet can simulate SDN networks, can run a controller for experiments. Mininet by default includes OVCS controller and Open vSwitch. We can install the controller of our choice using command given below.

- sudo apt-get install POX/FloodLight

Mininet provides a number of commands [18]. Some useful Mininet commands are listed below:

- $ sudo mn

In order to run mininet as root we must use the sudo command to run the mininet. This command starts the minimal topology and enters the command line interface. It is the default topology and includes OpenFlow switch which is connected to two hosts and the OpenFlow controller.

- sudo mn –h

This command is used in order to see the help menu available in mininet.

- mininet> nodes

This command display the nodes available in the current network which are available for the mininet default minimal topology.

- mininet> dump

This command display the dump information about all nodes available in the current mininet network.

- mininet> h_1 ping h_2

This command tests the connectivity between host h_1 and h_2. This command will keep checking the connectivity between hosts until we stop the command.

- mininet> h_1 ping -c1 h_2

This command checks for the connection between host h_1 and h_2 for one packet.

- sudo mn –c

This command is used in order to clear mininet or previously used command.

Mininet contains five default topologies [19] such as minimal, single, reversed, linear and tree. Network controller is denoted by C_0, switches are denoted by S_1 to S_n and hosts are denoted by H_1 to H_n in the following figures is used to represent these topologies.

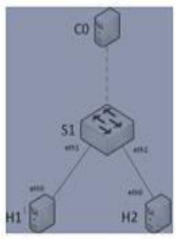


Fig.4 Minimal Topology

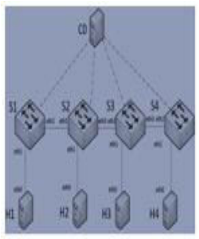


Fig.5 Linear Topology

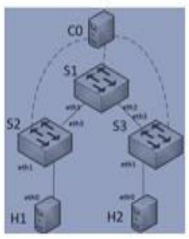
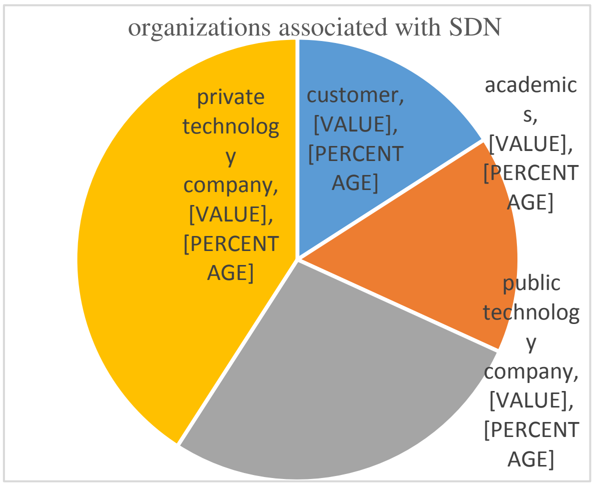


Fig.6 Tree Topology

To monitor the interactions between the controller and any switches in the network, we start Wireshark [20] and capture traffic on the Mininet. Wireshark must be started with superuser privileges so on the Mininet VM terminal. We set Wireshark to capture packet on the Mininet VM's loopback interface.

IV. ADAPTATION OF SDN

The adaptation path of SDN has gone from research networks to data centers and is now looking forward. In research networks SDN enables researchers to dynamically allocate resources from an existing network for their own use to test. In data centers SDN allows for the network to scale further more easily and security [21] in SDN is more flexible for a network where the hosts keep changing places.



It provides features which will revolutionize the future of networks like centralize control mechanism [22], cost efficiency, programmability, scalability , security, virtualization [2], cloud computing, reliability and efficient environment to support Big-Data [1].

In today's date Mininet is studied and researched in more than 100 universities around the world. The market report [23] summarizes the impact of SDN in industrial market. This impact of software-defined networking (SDN) will exceed $25B per annum by 2018, and could grow as high as $35B annually. The server virtualization market grew from nothing to nearly $1B annually in just over six years. Following pie chart depicts the various organization associated with SDN. Annual SDN purchases led by the Global 500 will grow from $20M in 2012 to more than $6B by 2018. Thisrapid emergence of SDN will heavily impact existing markets.

V. CONCLUSION

The traditional networking architecture can't accommodate today's requirements efficiently. Increase of mobile devices, virtualization, security, efficient Big-Data management and high quality of services has lead

SDN as a promising architecture. It enhances the scope of development in networking. Also, its impact is increasing at a very sharp rate day-by-day. SDN can be easily implemented and studied with mininet as it is open source network emulation software.

REFERENCES

[1] F. Alam, I. Katib and A. S. Alzahrani. "New Networking Era: Software Defined Networking". *International Journal of Advanced Research in Computer Science and Software Engineering*.Volume 3, Issue 11, November 2013

[2] D. Kumar and M. Sood. "Software Defined Networking: A Concept and Related Issues" in *Int. J. Advanced Networking and Applications*. Volume: 6 Issue: 2 Pages: 2233-2239 (2014).

[3] Pox. Available at: http://www.noxrepo.org/pox/about-pox.

[4] OpenFlow switch specification. Version 1.3. *ONF*. Available at: http://www.opennetworking.org/. 2012.

[5] Open Networking Foundation. "OpenFlow /Software Defined-networking (SDN)". Available at: http://www. www.opennetworking.org/.

[6] *IBM*. "IBM Systems and Technology Thought Leadership White Paper". 2012.

[7] An Introduction to Software Defined Networking. Whitepaper by *Citrix.*

[8] Software-Defined Networking: The New Norm for Networks. *ONF White Paper*.April 13, 2012

[9] B. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti. "A Survey of SDN- Past, Present and Future of Programmable Network" in *Communication Survey and Tutorials, IEEE.* Volume 16, Issue 3, pp. 1617-1634, Feb 13, 2014.

[10] HP OpenFlow and SDN Technical Overview**.** Technical Solution Guide. *White paper*. Version: 1. September 2013

[11] D. Drutskoy, E. Keller, and J. Rexford. "Scalable Network Virtualization in Software-Defined Networks". *IEEE Internet Computing*, 2013

[12] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. "Nox: towards an operating system for networks". *ACM SIGCOMM Computer Communication* Review, 38(3):105-110, 2008

[13] Helios by nec. Available at: http://www.nec.com/.

[14] Floodlight, an open SDN controller. Available at: http://foodlight.openowhub.org/.

[15] K. K. Sharma and M. Sood. "Mininet as a Container Based Emulator for Software Defined Networks" in *International Journal of Advanced Research in Computer Science and Software Engineering*. Volume 4, Issue 12, December 2014.

[16] B. Lantz, B. Heller, N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks". In *HotNets. ACM*, 2010.

[17] Introduction to Mininet at https://github.com/ mininet/mininet.wiki.git/

[18] Mininet Commands at http://mininet.org/

[19] Mininet Topologies at http://www.routereflector.com/2013/11/ mini net-as-an-sdn-testplatform

[20] Wireshark at https://www.wireshark.org/

[21] P. Porras, S. Shin and V. Yegneswaran. "A Security Enforcement Kernel for OpenFlow Networks".*HotSDN'12*, August 13, 2012.

[22] N. Feamster, J. Rexford, and E. Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". *ACM Queue*, Volume 11, Issue 12. 2013.

[23] SDN Market Sizing Report by *sdn central,* April 2013.

Biography

Pooja, received B.Tech degree in Computer Science from Himachal Pradesh University, Shimla in 2012. She is currently a M. Tech. candidate in the Department of Computer Science at the Himachal Pradesh University, Shimla. Her current research interest include computer networking and virtualization.

Manu Sood, currently working as Professor at Himachal Pradesh University. He received his M.Tech in Information Systems from Netaji Subhas Institute of Technology, New Delhi and Ph.D. from Faculty of technology, Delhi University, Delhi. His current research include SOA, Software Engg., MANETs, etc.