Mininet Implementation of SDN Towards Network Softwarization

[1] T Hari Krishna, [2]C Rajabhushanam

[1] [2]Department of Computer Science and Engineering Bharath University Chennai, India

[1]harikrishna.timmana@gmail.com, [2]rajabhushanamc.cse@bharathuniv.ac.in

Abstract: Internet has grown abruptly from last decade. Existing Internet resources like infrastructure, connection policies and management tools are not effective and difficult to manage. To decouple control plane from routing plane Software-Defined Networking (SDN) concept is evolved. This paper revisits a) SDN brief history, definition b) SDN Layered architecture c) Mininet overview, and installation basics d) Simulating the SDN using Mininet emulator with sample experiments . This paper provides overview SDN research& tools towards network sofwarization.

Keywords: Software Defined Networking, Definition, Architecture, Mininet, MiniEdit).

I. INTRODUCTION

Internet usage is increased vastly from last decade. Resources and users are connected to internet also increased. Existing network infrastructure and routing techniques are not sufficient to provide services to connected users. Adding and removing the internet resources are complex and time consuming; leads to lack of scalability. NEs (Network Elements) are vendor specific and expensive to maintain. In 2004 SANE[1] and Ethane [2] proposed an architecture that has a centralized controller to define and manage security policies of the network which is evaluated as OF/SDN[3]. Openflow protocol was published in 2009 by adding SDN programmability[4][5]. The Open Networking Foundation (ONF) has formed in 2011 to standardize, commercialize and promote the Openflow[6]. SDN is proposed[7] years back, but not yet implemented completely and the research is towards network softwarization. It is required SDN basics to understand the SDN features, architecture and implementing tools. In this paper we revisited the SDN definition in Section II. The architecture of SDN is briefed in Section III. Mininet overview, installation steps and sample experiments are presented in Section IV & V.

II. DEFINITION

The Open Networking Foundation (ONF) has published SDN Definition[8] in "Software-Defined Networking: The New Norm for Networks," as "Software defined networking (SDN) is an emerging network architecture where network control is decoupled from forwarding and is directly programmable. Network intelligence is centralized in software-based SDN controllers, which maintain a global view of the network". SDN key features are a) Decoupling the control plane from the data plane b)A centralized controller and view of the network c) Programmability of the network elements by external applications

III. ARCHITECTURE

In this section, we describe the SDN architecture[9] and its primary components. ONF has evaluated, SDN architecture that separates the network control(i.e. control plane) from forwarding functions(i.e. data plane). This enables the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. In such an architecture, the infrastructure devices become simply forwarding engines that process incoming packets based on a set of rules generated on the fly by a (or a set of) controller at the control layer according to some predefined program logic. Fig.1 shows the overview of SDN architecture
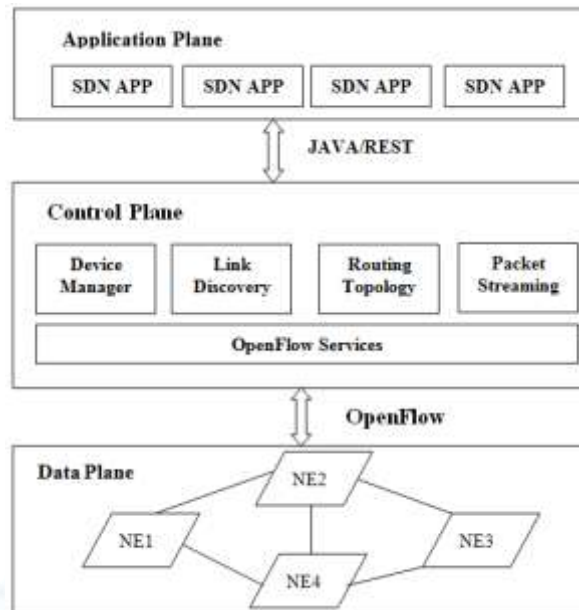
- SDN Layerd Architecure

   The controller generally runs on a remote commodity server and communicates over a secure connection with the forwarding elements using a set of standardized commands. ONF presents in [9][10] a high-level architecture for SDN that is vertically split into three main functional layers:

Data Plane

This layer handles the data packets and applying actions to them, based on rules that is programmed into lookup tables. It is also known as the data layer [11] or forwarding plane. It contains mainly of Forwarding Elements (FEs) including physical and virtual switches accessible through an open interface. Control plane provides input to data plane about the data path to where the data need to be sent.

Control Plane

This layer performs identifying and programming action to data plane. It contains a set of software-based SDN controllers providing a consolidated control functionality through open APIs to manage the network forwarding actions via an open interface. Control plane operates at a lower speed than the data plane and the controller API's are independent of core hardware.

A SDN controller interacts with these three layers through three open interfaces:

a) Southbound: This communication interface allows the controller to interact with the forwarding elements in the data plane. ONF [6] has identified OpenFlow protocol is an a foundational element for structuring SDN solutions

b) Northbound: This interface enables the programmability of the controllers by exposing universal network abstraction data models and other functionalities within the controllers for use by applications at the application layer. It is more considered as a software API than a protocol that allows programming and managing the network. At the time of writing this paper, there is no standardization effort yet from the ONF side who is encouraging innovative proposals from various controllers developers. According to the ONF, different levels of abstractions as latitudes and different use cases as longitudes have to be characterized, which may lead to more than a single northbound interface to serve all use cases and environments. Among the various proposals, various vendors are offering a REpresentational State Transfer (REST)-based APIs [12] to provide a programmable interface to their controller to be used by the business applications.

East/Westbound: This interface is used to gain high availability[13] by allowing communication among groups or associations of controllers to synchronized status.

Application Plane

This layer is called as Management plane which of the end-user applications that consume the SDN communications and network services [14].This is used to configure and monitor the network device by using shell , command line interface or web services. Network visualization and security business applications [15] examples for end-user applications.

There are multiple SDN implementations available namely NOX[16] and POX [17] by Nicira, Floodlight[18] by BigSwit and Open Daylight[19] by multiple Vendors. Open Flow is the standardized protocol at South bound Controller interface to communicate with NEs. There is no standardized protocol for North bound controller. Feature based comparison of open source controllers are mentioned in Appendix Table. 1.

IV. MININET

Mininet[20] is a network emulator that constructs a network of virtual hosts, switches, controllers, and links. It enables research, development, learning, prototyping, testing, debugging, and any other tasks that could benefit from having a complete on a computer. It makes easy to create complex and customized network topologies to work multiple concurrent users independently. It provides Command Line Interface (CLI) support for creating and testing the test beds. It also contains MiniEdit tool that supports designing and implementing the network through user interface edit has been completed, the paper is ready for the template. Duplicate the template file

V. RELETED WORK

We worked with Mininet 2.2.1[21] on Ubuntu16.04 running with Intel Core i7-4700MQ CPU processor at a 2.40GHz x 8, 5.7 GB of Memory. We used VMware[22] to install Ubuntu 16.4 on windows system. Mininet is a open source tool, source code can be retrieved using below git URL
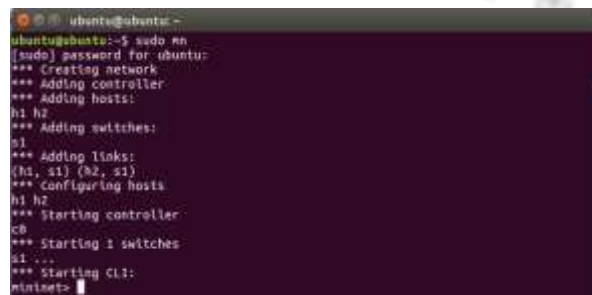
$ git clone

BELOW COMMAND USED TO INSTALL MININET ON UBUNTU.

$ ~/mininet/util/install.sh -a

TO VERIFY THE INSTALLATION WE USED BELOW COMMAND

$ sudo mn --test pingall

Output of the "mn" command is as shown in Fig.2

- Mininet "mn" command output

Following experiments[23][24] helps to understand the basics of Mininet.

First Experiment

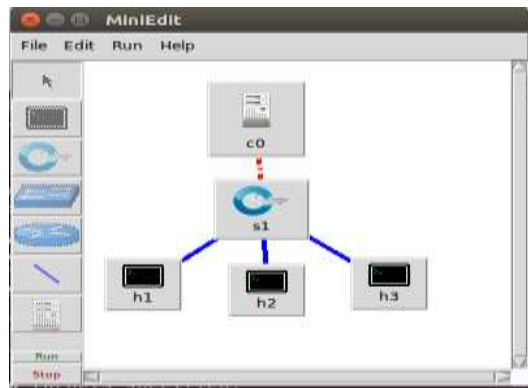One controller c0 connected to three hosts namely h0, h1, and h2 through a switch s1.

Fig.3 MiniEdit single switch network

The following CLI command is used for network creation.

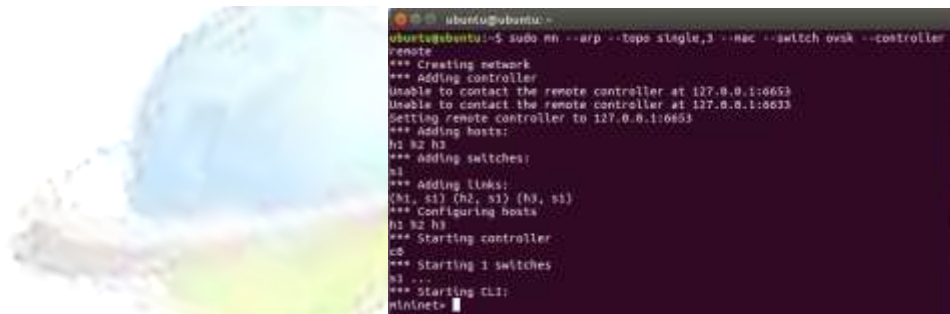$ sudo mn --arp --topo single,3 --mac --switch ovsk --controller remote

Fig4. Mininet single switch network

Second Experiment

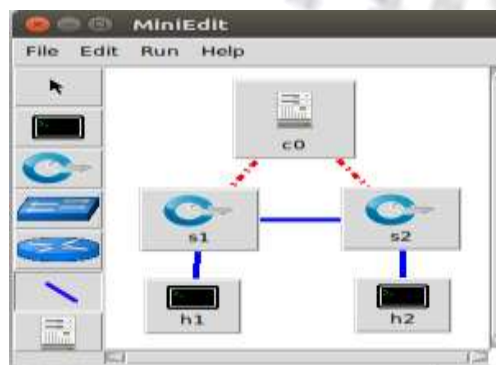One controller (c0) connected to two hosts namely h0, h1, and h2 through two linear switches(S1, S2).

Fig.5 MiniEdit linear switch network

The following CLI command is used for network creation.

$ sudo mn --topo linear --switch ovsk --controller remote

FIG.6 MININET LINEAR SWITCH NETWORK

VI. CONCLUSION

SDN is an evolving architecture that decouples network control from data plane.. SDN is proposed years back, but not yet implemented completely and the research is towards network softwarization. Through this paper we revisited the SDN definition, architecture of SDN, Mininet overview, installation steps and sample experiments. This helps the researchers to understand the SDN and its tools towards Network softwarization[25].

VII. FUTURE WORK

In this paper we discussed SDN and Mininet basics. In Future we may construct the complex networks with hybrid topologies to do SDN research towards Network softwarization.

References

[1] M. Casado et al, "SANE: A protection architecture for enterprise networks", USENIX Security, 2006
[2] M. Casado et al, "Ethane: Taking Control of the Enterprise", Sigcomm 2007
[3] A. T. Campbell et al., "A survey of programmable networks," SIGCOMM Comput. Commun. Rev., vol. 29, no. 2, pp. 7–23, Apr. 1999
[4] ONF, "Software-defined networking: The new norm for networks,"Open Networking Foundation, Tech. Rep., April 2012.[Online].Available:https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf
[5] Zaigham Mahmood, Cloud Computing: Challenges, Limitations and R&D Solutions" Springer, Oct 20, 2014
[6] N. McKeown et al, "OpenFlow: Enabling Innovation in Campus Networks", white paper, 2008
[7] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," Communications Surveys Tutorials, IEEE, vol. 16, no. 3, pp. 1617–1634, Third 2014.
[8] Opennetworking.org, "Software-Defined Networking (SDN) Definition, 2017. [Online]. Available: https://www.opennetworking.org/sdn-resources/sdn-definition.
[9] Big Switch Networks. (2012, Oct.). The Open SDN Architecture. [Online]. Available: http://www.bigswitch.com/sites/default/Þles/ sdn_overview.pdf.
[10] RFC 7426"Software-Defined Networking (SDN): Layers and Architecture Terminology"[Online]. Available: https://tools.ietf.org/html/rfc7426
[11] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," Communications Surveys Tutorials, IEEE, vol. PP, no. 99, pp. 1–1, 2014
[12] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Univ. California, Irvine, CA, USA, 2000
[13] "Realizing the Power of SDN With HP Virtual Application Networks," Hewlett-Packard Development Co., Cupertino, CA, USA, Tech. Rep., 4AA4-3871ENW, Oct. 2012.
[14] Stefano Vissicchio et al."Opportunities and Research Challenges of Hybrid Software Defined Networks" IETF Proceedings,[Online]. Available: https://www.ietf.org/proceedings/89/slides/slides-89-sdnrg-0.pdf
[15] "SDN security considerations in the data center,"Open Networking Foundation, Palo Alto, CA, USA, Solution Brief, Oct. 2013.

[16] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," Comp.Comm. Rev., 2008.

[17] M. McCauley, "POX," 2012. [Online]. Available: http://www.noxrepo.org/

[18] FloodLight," Project Floodlight ", [Online] Available: www.projectfloodlight.org/floodlight

[19] OpenDaylight,"Open Source SDN Platform" [Online] Available:https://www.opendaylight.org/

[20] GitHub (2017) "Mininet: Rapid Prototyping for Software Defined Networks" [Online]. Available: https://github.com/mininet/mininet

[21] SDN Hub. (2016, April 20). Useful Mininet setup. [Online]. Available: http://sdnhub.org/resources/useful-mininet-setups/

[22] VMware, Inc., "VMware NSX Virtualization Platform," 2013.[Online].Available:https://www.vmware.com/products/nsx/

[23] F. Keti and S. Askar. "Emulation of Software Defined Networks Using Mininet in Different Simulation Environments," in 6th International Conference on Intelligent Systems, Modelling and Simulation, 2015.

[24] Te-Yuan Huang et.al,"Teaching Computer Networking with Mininet"Wi-Fi: HHonors, SigComm2014[Online]. Available: http://conferences.sigcomm.org/sigcomm/2014/doc/slides/mininet-intro.pdf

IEEE SDN Initiative http://sdn.ieee.org