Develop a fully integrated Software Defined Networking (SDN) test bed for Educational and Research Purposes

Amarasinghe A.A.D.K.P, Tharaka E.L.Y, Hashantha H.L.S

SUPERVISORS: Dr. Asitha Bandaranayke, Dr. Suneth Namal

Department of Computer Engineering, University of Peradeniya.

i. Abstract

Software defined networking (SDN) is a new approach of networking that revolutionized the traditional way of networking. It makes easy to designing, building, and managing networks by separates the networks control plane and forwarding plane. Centralizing the control of the network and network devices become very simple hardware units, SDN offers very easy way of defining policies of the network and flexible ways of control and monitor traffic flows. Objective of this project is to build a SDN Testbed for educational and research purposes using commodity embedded hardware (e.g.: PCs, servers, NetFPGA). The ultimate goal of this is make an educational tool which offers a visualization of traffic flows and several advanced network functionalities like network slicing, dynamic resource allocation and network function virtualization.

ii. Introduction

SDN is an alternative new paradigm to address the traditional networking paradigm problems like management issues, scalability issues. There are many researches undertaking many sub areas of SDN. Data center level there are many use cases of SDN networks. Programmers can create applications that perform actions on the underlying SDN network and can achieve flexibility, efficiency and most importantly resiliency for data centers.

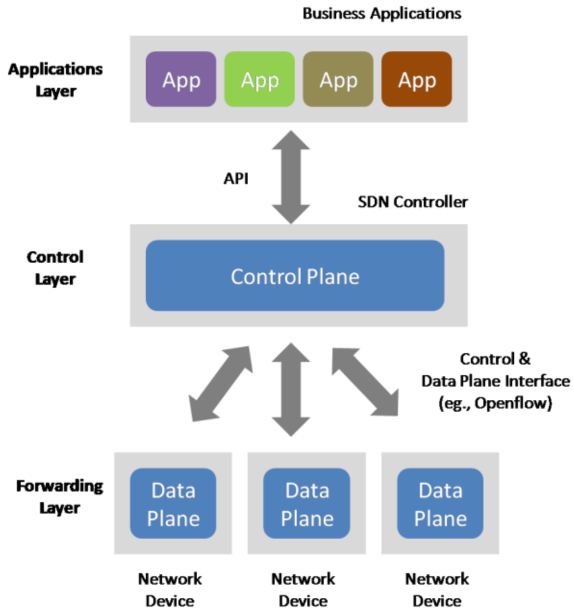
Main issue of the SDN network implementations is initial cost of the implementation. This lead to doing researches on this area discouraged. To address this problems SDN networks for research emerged. For example, OF@TEIN, GENNY. But at beginners level it is hard to do simple experiments in this kind of complex networks. Then a cost effective, easy to implement SDN network implementation is required. Also it would be better for beginners to get first hands on SDN network experience before going in to large complex networks.

This Project is to build an SDN Test bed for educational and research purposes in order to facilitate the comprehension about Programmable Network using commodity embedded hardware (e.g.: PCs, servers, NetFPGA). The ultimate goal here is to make cost effective SDN test bed for beginners which offers a visualization of traffic flows and several advanced network functionalities like network slicing, dynamic resource allocation and Network Functions Virtualization (NFV), that could be used as a teaching aid, as well as a research test bed.

iii. SDN Architecture

Traditionally, control plane and data plane of a traditional networking architecture were bind together. In OpenFlow standard which was created in 2008, was recognized as the first SDN architecture which defined how should control and data planes would be separated and communicate with each other [4].

SDN architecture divided into three separate sections,



Application Layer: In application layer SDN applications are running. SDN Applications communicate with the Control layer via well-defined application programming interface (APIs). In addition, the applications can build an abstracted view of the network by collecting information from the controller. These applications can perform networking management, analytics, security and business applications used to run large data center level.

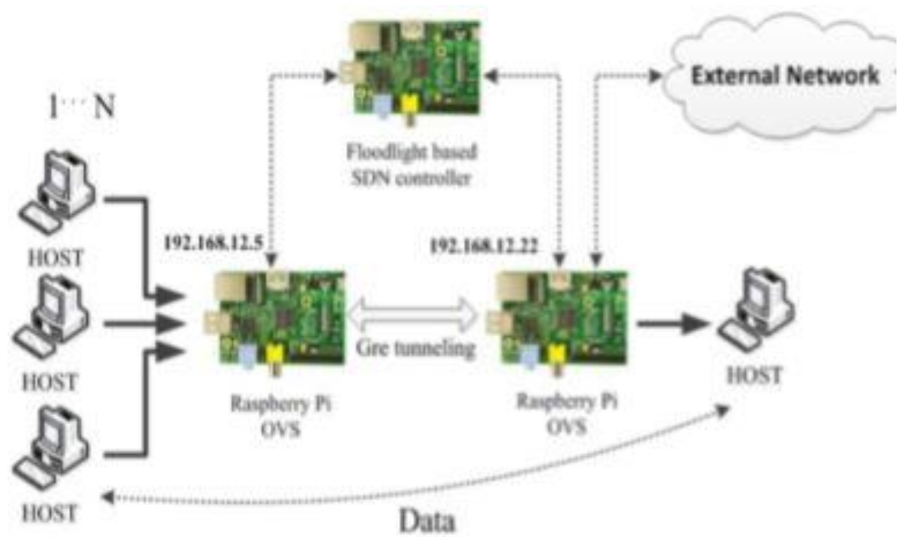
Control Layer: The SDN Controller does maintain communication between networking component and the network application. To do that controller has well defined API called North bound and south bound. North bound API enable applications to communicate with the controller and south bound API enable communication between controller and the network hardware. That defined controller as a logical entity that receives instructions or requirements from the SDN Application layer and relays them to the networking components. Also controller extracts information about the network from the hardware devices and communicates back to the SDN Applications with an abstract view of the network.

Forwarding Layer: The network devices which control the forwarding data and processing data for the network.

Open flow is just a part of the SDN stack called as a south bound protocol [4]. This enable communication between controller and the network devices.

iv. SDN Testbed Design Architectures

One approach of designing low cost SDN networks [3] described designed using Raspberry-PI computers, Virtual SDN switch (Open vSwitch) [6] and Floodlight SDN controller.



But Raspberry-PI's processing power and resources is not enough when virtual switch instance running. It causes the raspberry-PI nodes to fail. Then constant operation was not possible.

Also Raspberry PI consist of on Ethernet port. Multiple hosts should connect to SDN switches running on the Raspberry-PI. In this approach it used virtual Ethernet ports to do that. But it leads to performance of the system drops.

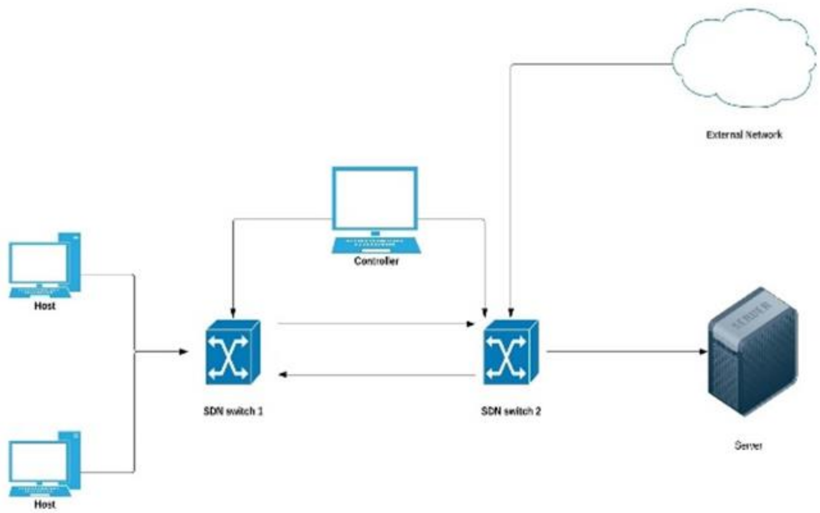
Same kind of approach described [2] with little differences. It used Raspberry PI, OpenvSwitch and Open Daylight Controller. It mentioned in its conclusion [2]

> ***"The tested performance statistics were much lower comparing to a network built on a Hardware OpenFlow switches like NetFPGA because of the speed limitations of USB to Ethernet converters and Raspberry Pi processor. But for non-performance critical SDN research and educational purposes statistics were in an acceptable range. In the future, The SDN testbed will be extended in to a full featured small-scale SDN offering additional features like traffic visualization, network slicing, dynamic resource allocation or network function virtualization"***

New approach expected to overcome this performance issues by using NetFPGA. Also expected to expand the testbed to full featured small-scale SDN testbed with additional features like visualizations and adding several advanced network functions.

v. New Design Approach

Our objective is creating an extended SDN test bed with these functionalities and create documentations and tutorials based hands on experience SDN learning platform for beginners. New approach included NetFPGA based SDN switch for performance increment.



Programming NetFPGA board as a SDN switch is a complex task. described a Verilog implementation of a SDN switch and source code can be found on a GitHub repository [5]. Source code compatible with NetFPGA 10G variation. The paper helped to understanding the underline implementation of the source code [1].

Selected NetFPGA was **NetFPGA-1G-CML** is a versatile, low cost network hardware development platform featuring a Xilinx® Kintex®-7 XC7K325T-1FFG676 FPGA and includes four Ethernet interfaces capable of negotiating up to 1 GB/s connections. 512 MB of 800 MHz DDR3 can support high-throughput packet buffering while 4.5 MB of QDRII+ can maintain low-latency access to high demand data, like routing tables. Rapid boot configuration is supported by a 128 MB BPI Flash, which is also available for non-volatile storage applications. The standard PCIe form factor supports high-speed x4 Gen 2 interfacing. The FMC carrier connector provides a convenient expansion interface for extending card functionality via Select I/O and GTX serial interfaces. The FMC connector can support SATA-II data rates for network storage applications, and can also be used to extend functionality via a wide variety of other cards designed for communication, measurement, and control.

There are several different SDN controllers which has different functionalities. In this approach it considered multiple SDN controllers like POX, Ryu, Trema, FloodLight, and OpenDaylight [7]. Here Controllers choose by it provided properties (like java interface provided default by the controller) and GUI functionality provided. Also tested all the different SDN controller by implementing them in a virtual network with Virtual machines. Among all of the controllers OpenDaylight was selected.

It is planned to use two host computers and one server for first basic implementation. On top of this network design applications developed. When requirement changed it is expected to further expand the network.

vi. Conclusion

SDN is a new field of research which rapidly growing. There are many interests and opportunities ahead in SDN. Leading university students toward that side will make a huge difference. To fulfill that intention, the first bottleneck is infrastructures to do experimentations. Also understanding core concepts of SDN paradigm properly is important. Instead of depend only on theoretical knowledge of SDN, Hands on experience and practical knowledge will helps to build well knowledged SDN experts. Building a platform to gain practical and theoretical knowledge will be a major step to that.

The low cost SDN testbed can be used for research and educational purposes. Also when designing such a testbed it is necessary to consider final system performance also. Then tradeoff between cost and performance is required. In this approach other than using Raspberry PI based SDN virtual switches, NetFPGA used to gaining the overall system performance. We employed OpenDaylight as the SDN controller framework. as opposed to the previous projects done in this area, we provide complete methodology and documentation for building a similar SDN testbed for interested third parties.

References

[1] Jad Naou, David Erickso, G. Adam Covington, Guido Appenzeller, Nick McKeown, “Implementing an OpenFlow Switch on the NetFPGA platform “

[2] Weerawardhana J.L.M.N.⇜, Chandimal N.J.A.W.†, Bandaranayake A.‡. “SDN Testbed for Undergraduate Education”, 2014

[3] Hyunmin Kim*, Jaebeom Kim**, Young-Bae Ko**. “Developing a Cost-Effective OpenFlow Testbed for Small-Scale Software Defined Networking”, 2014

[4] Nick McKeown-Stanford University, Tom Anderson-University of Washington, Hari Balakrishnan-MIT, Guru Parulkar-Stanford University, Larry Peterson-Princeton University, Jennifer Rexford-Princeton University, Scott Shenker-University of California, Jonathan Turner-Washington University, “OpenFlow: enabling innovation in campus networks”, 2008

[5] https://github.com/NetFPGA/NetFPGA-public/wiki/NetFPGA-10G-OpenFlow-Switch

[6] Simon Horman, “An Introduction to Open vSwitch”, LinuxCon Japan, Yokohama

[7] Rahamatullah Khondoker, Adel Zaalouk, Ronald Marx, Kpatcha Bayarou. “Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers”