Building SDN framework using OpenDaylight Controller

Prakruthi S[1], Usha K Patil[2]

[1]*MTech Student, Computer Science & Engineering, GSSSIETW, Mysuru, India,*
[2]*Assistent Professor, Dept of CSE, GSSSIETW, Mysuru, India,*

Corresponding Email: **[1]prakruthigowda24@gmail.com**

ABSTRACT

The demand for more effective networking environment is required in today's immutable infrastructure to handle with rapidly changing requirements of the users. Therefore (SDN) Software Defined Networking was brought in around 2005 to reconstruct current network to have accelerated innovation, consolidate administration and programmability by separating the data plane and control planes. This paper deals with developing Software Defined Networking framework by using the OpenDaylight controller. This runs over a SDN controller based on OpenFlow to program the application which can be built on the software, without any dedicated hardware

INTRODUCTION

OpenDaylight (ODL) is standard [1] platform for open SDN, which can be for any scope and size of network and permit network services over a range of hardware in a network. Many networks are designed in such a way that it can fulfil the needs and requirement of that point in time.

Software defined networking is the one which is capable of improving current network by easily accepting the changes as the user requirements. This is because OpenDaylight is capable of dealing with network challenges by allowing for constructing the universal platform so that it can set up in several ways. Standard OpenDaylight incorporates open source, open APIs to hand over the SDN platform, which prepare the network for highly adoptable, programmable, and intelligent.

The network is portrayed through model-driven method for performing some functions and achieving outcomes by OpenDaylight controller. In order to solve complex problems OpenDaylight contribute data structures in familiar massaging framework and data store, and by allowing elegant services to be designed and combined together.

In SDN platform ODL supports extensive collection of protocol for classic as well developing network and enhances the programmability of current networks. For example SDN platform supports OpenFlow, Table Type patterns- extension of OpenFlow, and also other protocol like BGP/PCEP, CAPWAP and NETCONF. Along with this it also consolidates with Open Vswitch and open stack through OVSDB Integration plan. Following are some of the other multiprotocol support provided by the ODL.

- Easily set up into current or greenfield network
- There are huge number of protocol support from NETCONF to OpenFlow protocol
- The ODL allows for endless innovations since it is open source model.
- Ability to install required functions and protocols that user needs.
- To overcome complicated issues, ODL allows incorporating different protocols and many services.
- Computable model of ODL environment allows to influences the new designs.

By broadcasting similar API structures, the Software Defined Networking programmability can be managed and developer can build abstractions of south or north over the controller.

RELATED WORK

1: Design and Implementation of SDN Controller Platform

According to this paper the [3] author uses the open sources controller and idea of SDN to implement four-tier architecture of SDN and control platform. Application interface, message acceptance and other services are obtained and managed by establishing the connection with SDN. Author also targets to execute device administrative module by implementing message transformation layer, and event sharing module.

2. An Implementation Model and Solutions for Stepwise Introduction of SDN

In this paper, writer [4] tells step wise implementation of software defined networking which runs on the controller along with quagga (routing software) and SDN switch. The controller is a one which is capable of handling and controlling information evenly throughout the SDN by the higher level layer of the controller and able to solve complex problems that rises during implementation SDN without losing benefits of SDN.

PROPOSED WORK

The OpenDaylight controller [2] is software of java virtual machine and it can be installed in any operating system and any platform which supports java. To implement SDN, OpenDaylight controller uses following tools:

MAVEN: To form automation ODL make use of Maven. Project Object Model (pom.xml) is used by Maven to write dependencies in bundles and also start and load the bundles.

OSGI: This structure is used for binding bundles for transaction of information, it allow for actively loading JAR file packages and bundles as back-end system by OpenDaylight.

JAVA interface: This interface is a main path for distinct bundles to demonstrate experience of unique state. Java interface is used for forming patterns, event accepting and requirements.

REST APIs: Northbound APIs are rest APIs like passive routing, topology controller, flow programmer, and host tracker.

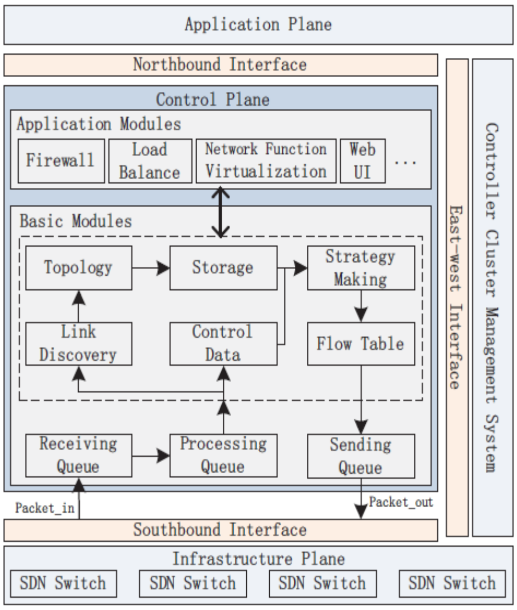

Figure 1: Framework of the SDN controller.

Applications use northbound APIs which are exposed by the controller. The northbound APIs are supported by duplex REST and OSGi framework. The applications which operate on same address spaces are supported by OSGi where as application which operates on different address spaces are supported by REST APIs on the controller.

In the network algorithms, business logic endure in applications, these applications collect network agility; logical analysis is carried out by executing some algorithm and coordinates the new rules all over the network. There are many plugins which are supported by numerous protocols and southbound APIs example like BGP-LS, OpenFlow and so on. Service Abstraction Layer (SAL) is associated actively to this modules and display the function to northbound APIs are written.

Controller and network devices consist of protocols, the requests from protocol is fulfilled by SAL. This also brings security to applications when OpenFlow and other protocols emerge. Topology manager is responsible for controlling and storing the data, which is required by the controller in order to control the devices by their reachability and capability. The topology database is generated by the factors like switch manager, APR handler, device manager, and host tracker by the topology manager.

Networking in SDN

The operating systems in the network assemble the information using application programming interface and contribute a hypothetical model of system topology and forwarding plane is manipulated and given to the controller in order to control the application. Elasticity and measurability are supported to have an entire awareness of the network to develop flow administration by the controller.

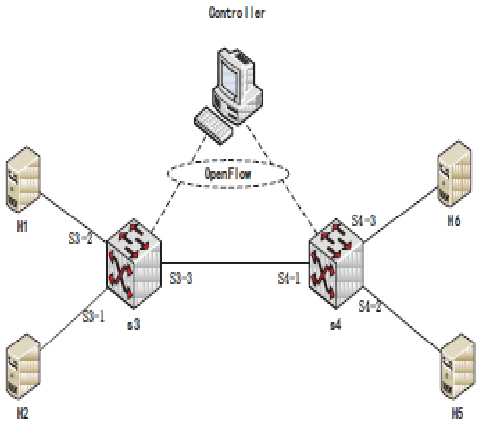

Figure 2: SDN network topology

As shown in the above figure, [5] when the packet is received from the sender at the switch, the flow rules are checked by the switch from the SDN caches. If corresponding entry is found the flow is executed with the relevant instructions and then send to the receiver otherwise packets are send back to the controller through secure channel. Effectively and responsibly the controller can update, add and delete the flows using southbound APIs. The controller performs and adds new entry to routing table and for each of the flow path along the corresponding switches. When the correct port receives the packets from the switch it is then forwards the packet to destination.

CONCLUSION

SDN is a protected phrase which is enclosing different kind of technologies intended to construct the network as active and elastic in latest data center used as storage infrastructure and virtualized server. OpenDaylight controller is open source project for advancing SDN by incorporating several tools, APIs and protocols by the controller. This paper tells about using OpenDaylight and building the SDN framework. And also tells how traffic flows from the sender to the receiver.

This project can be enhanced by analysing the performance of the traffic flow by considering the performance matrices like type of network, packet count, time and much more.

REFERENCES

[1]. “Overview of OpenDaylight”, https://www.opendaylight.org/platform-overview-beryllium, Mar 2016.

[2]. “OpenDaylight User Guide”, by Beryllium https://drive.google.com/file/, Feb. 2016.

[3]. Yi, Lu, et al. "Design and implementation of SDN controller platfom." Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on. IEEE, 2015.

[4]. Nakayama, Hiroki, et al. "An implementation model and solutions for stepwise introduction of SDN." Network Operations and Management Symposium IEEE, 2014.

[5]. Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." Communications Magazine, IEEE 2013.