Software Defined Networking with Floodlight Controller

Vidya B. Harkal
S. K. N. College of Engineering,
Pune – 411046

A. A.Deshmukh
S. K. N .College of Engineering,
Pune – 411046

ABSTRACT
Software defined networking (SDN) is combine approach of connection oriented technologies and routing overlay technology presenting new open and programmable network. Decoupling control and data plane from network devices and bring control plane in logically centralized SDN controller which can be then act as network operating system.

This paper is to present an OpenFlow controller – Floodlight Controller. The Floodlight Controller realizes a set of common functionalities to control and inquire an OpenFlow network, while applications on top of it realize different features to solve different user needs over the network. Floodlight consists of controller modules that implement core network services a software defined network would expose to applications, and application modules that implement solutions for different purposes.

Keywords
Software defined network, floodlight controller

1. INTRODUCTION

1.1. Software Defined Networking
This is an era of internet which highly dependent on network for performance. In this, infrastructure has main two components such as simple, vendor neutral & future proof hardware and flexible software. These things are considered as properties of ideal network [1], but exception is legacy network which does not satisfy any of the above goals. In legacy infrastructure decision making functionality is available inside the device for that control plane & data plane work together and make complex and hard to network manager. Software defined networking is programmatic operator network interface, which helps to address a wide verity of operator requirements without changing any of the lower level requirements of the network [2]

Benefits of Software Defined Networking

- Programmability
- Openness
- Centralized control
- Abstraction & Virtualization
- Rapid Innovation

1.2. Software Defined Networking (SDN) Architecture
Most of the networking device has control and data plane working on same device. Only control available to network administrator is from the network management plane, which is used to operate and configure each node separately. The static nature of current network devices do not allow detailed control plane configuration because of this drawback SDN comes into the picture. It provides open user controlled management of the forwarding hardware of a network element. Network hardware devices keep their switching fabric and handover their intelligence to the controller .SDN decouple control and data plane and brings control plane logically centralized, This network operation system is known as software defined networking controller which helps to control ,change ,manage network behaviour dynamically through software interference[3].

Fig. 1 demonstrates the different layers and components of SDN architecture.

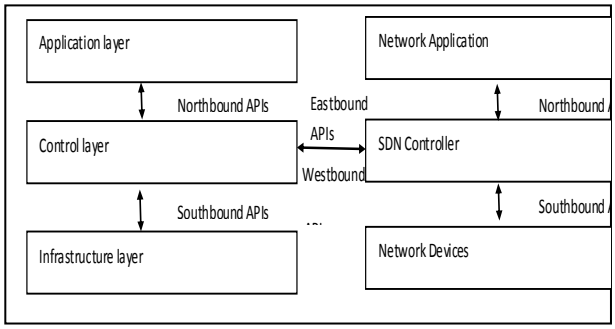

Fig.1 Architecture of SDN

As shown in Fig.1 SDN architecture has below mentioned layers -

- **Infrastructure layer**

This layer includes network devices such as routers, switches, middle boxes, firewall etc. Flow based forwarding carried out through these network devices.

- **Control layer**

Network operating system is a part of Control layer which control low level infrastructure resources which are logically centralized controller. This layer present abstract view over infrastructure layer which enabled administrator to apply custom polices over lower layer of SDN controller which act as brain of network.

SDN controller broadly divided into two categories:

1. Open source single instance controller
2. Closed source distributed controller

- **Application layer**

This layer act as platform on top of the various network applications. This layer provides services managed by applications such as adaptive routing, network monitoring, security management, network virtualization etc.

- **Interfaces**

SDN architecture consist of different interfaces such as

1. Northbound interfaces

The northbound application programming interface (APIs) is Interface between the software module of controller platform layer and application layer is northbound interface, it includes

RESTfull APIs, adhoc APIs, NVP NBAPI, SDMN API.

2. Southbound interfaces

The southbound application programming interface is interface between control layer and infrastructure layer it includes openflow, OVSDB, L2, L3 agent

3. Eastbound – Westbound Interface

This controller used for multi controller based architecture eastbound westbound interface where it used to communicate between controllers.

1.3. Wireless SDN

Wi-Fi network have several unique properties over the wired network. SDN was first applied to wired environment and now wireless SDN is new opportunity for research. In SDWN architecture wireless termination point [WTPs] refer to physical devices coincide with Access point [APs] form infrastructure layer .WTPs communicate with remote SDWN controller through secure channel.

SDWN architecture has two different models [4]:

1. Centralized wireless network [centralized SDWN]
2. Distributed wireless network [Distributed SDWN]

Main functionalities in wireless environment are managing wireless access point, user verification, mobility and handoff management, security etc.

2. LITERATURE REVIEW

SDN has long history it introduced in mid-1990 and still new approaches and improvements are going on for making it open and programmable. There are many open controller used to improve SDN performance and provide security to SDN. Nick feamster et.al[5] all in their paper explained timeline history of SDN in three section active network (mid 1990-2000), control data separation (2001-2007), open flow and network operating system (2007-2010). Shah et all explained in [6]the detailed architecture four open flow controller namely NOX, Becan, Maestro and floodlight, they provided detailed evolution of these 4 controller, results of this evolution used as a key guideline for architecture to improve existing or new architecture plane . Porras et.al. [7] focused on security issue of SDN by explaining security enhanced-version of floodlight, this is known as SE-floodlight with security enforcement kernel. Wallner et.al. [8] addressed issues like quality of services of SDN, what is importance of QoS in SDN and how to managed QoS services through SDN. They provide solution as Queue based classification in OVS. Riggio et.al.[9] proposed programming abstract for wireless SDN it includes different types of abstractions like Light Virtual Access Point (LVAP) abstraction, Resource Pool abstraction, Channel Quality and Interference Map abstraction, Port abstraction etc. Below table (Table No. 1) shows some of the open source controller available which used in SDN

Table 1. Open Source SDN Controllers [16]

Controller	Created by	OpenFlow version	Description
NOX	Nicira	1.0, 1.3	Multithreaded, asynchronous, event based programmable model by Nicira .its first controller .
POX	Nicira	1.0	Python based modular controller .
Beacon	Stanford university	1.0.1	Multithreaded event based , cross platform java based controller
Maestro	Rice university	1.0	High performance using multithreaded java based controller
Floodlight	Big Switch Networks	1.0	Java based Modular architecture
Floodlight-plus	Big Switch Networks	1.3	New version of floodlight for supporting OF 1.3
Ryu	NTT Labs	1.4	Component based, python based controller
Open Daylight	Linux Foundation	1.0 , 1.3	Modular java based

3. SYSTEM MODEL AND ITS DESIGN GOALS

3.1. System Model

Fig. 2 demonstrates the system model, which consists of 3 components' such as Application layer of SDN, SDN Controller and Infrastructure layer.

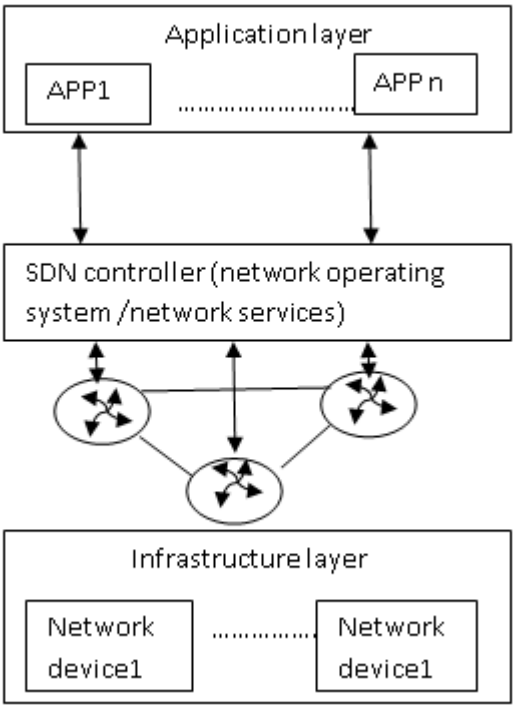

Fig. 2 System Model

SDN Controller in nothing but the control layer of the architecture which work as a centralized controller. This controller allows infrastructure admin to apply custom policies over the control layer in short controller is act as a brain of a network.

3.2. Design Goals

This proposed SDN controller service should achieve below mentioned design objectives:

- High availability - Implementing high availability by providing cluster of controllers.
- Reliability - Reliability of SDN controller relies on active-standby mode by controller node, secure connection between controller and switch nodes, multi-controllers based on openflow.
- Scalability - Scalability of SDN controller relies on node upgrading without service interruption and unique node upgrade in the distribute systems without any influence on the whole system.
- Improve overall performance of SDN

4. PROPOSED SYSTEM

Proposed system is consisting of implementing floodlight controller in SDN to improve its performance by overcoming the current gaps it has. Floodlight has a sync module, which is designed to allow for multiple floodlight instances to work collaboratively and share/sync state between each other. It isn't integrated into the controller core though but the ISyncService module is missing which helps to share information between controllers in a cluster in order to maintain state between the controllers. ISyncService is a Floodlight service provided by the sync module for implementing high availability amongst a cluster of Floodlight controllers. The idea is that module can become IStoreClients and IStoreListeners in order to notify or be notified upon a state change of the controller cluster. Having a robust and well-tested sync module would add a lot of value to Floodlight.

Floodlight control is the back bone of this system; the Floodlight Open SDN Controller is an enterprise-class, Apache-licensed, Java-based OpenFlow Controller. It is supported by a community of developers including a number of engineers from Big Switch Networks. Fig. 3 demonstrates the 'modular architecture' is used to describe the architecture of Floodlight Controller.

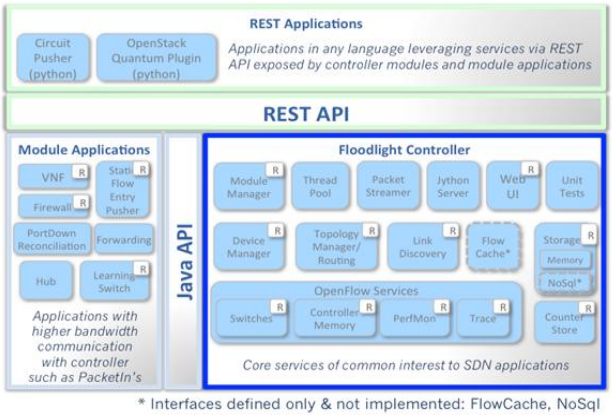

Fig. 3 Architecture of Floodlight Controller [8]

4.1 System Architecture

OFSwitchManager is a Floodlight module designed to manage all the OpenFlow switches connected to the Floodlight controller. It can be used to get references to and interact with switches such as send OFMessages like OFFlowMods and OFPacketOuts. Modules can leverage the OFSwitchManager by requesting a reference to the IOFSwitchService.

Services Provided -

- IOFSwitchService

Service Dependencies –

- IFloodlightProviderService
- IDebugEventService
- IDebugCounterService
- ISyncService

Fig. 4 shows the system architecture of the proposed system which has clusters of controllers which helps to achieve high availability, for this new service ISyncServiceis introduced.

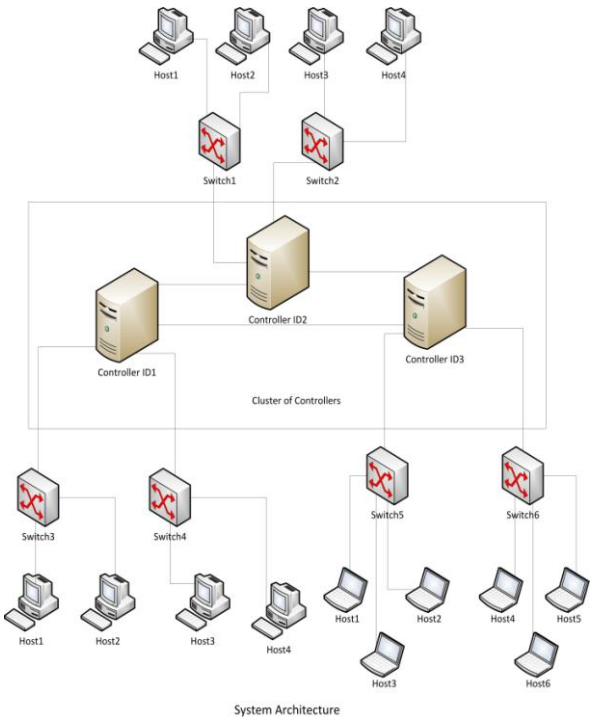

Fig. 4 System Architecture

- ISyncService

ISyncService is a Floodlight service provided by the sync module for implementing high availability amongst a cluster of Floodlight controllers. This is a module designed to maintain a database that is effectively mirrored across all participating controllers.

ISyncService is implemented as a singleton module that maintains the sync state for a given controller. Each controller will have a separate ISyncService module that communicates with other controllers' ISyncService modules.

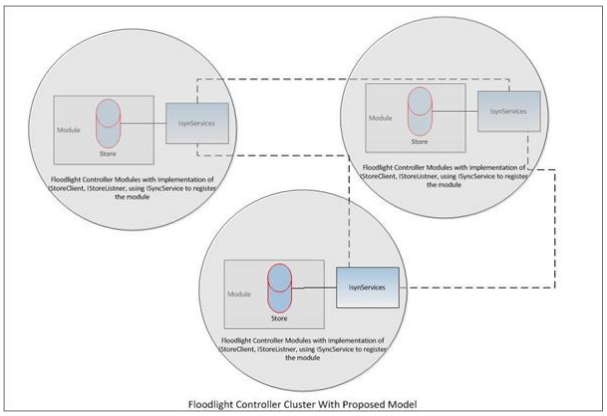

Fig.5 Positioning of ISyncService

4.2 Mathematical Model

SDN-based network there are flow level and packet level services .where packet consider as basic unit of services for packet level based system and flow between controller and switch as basic unit for flow level based system.

Considering both packet-level and flow-level arrival processes in the network implies unique arrival and service characteristics and requirements. [13]

Packet enter in system goes to controller at least ones. For two node in system (controller and switch or controller -controller) forwarding queue of the type M/GI/1and feedback queue of delay loss type M/GI/1 – S can be consider for both type of system. Bases on that performance measure can be calculated. [14]

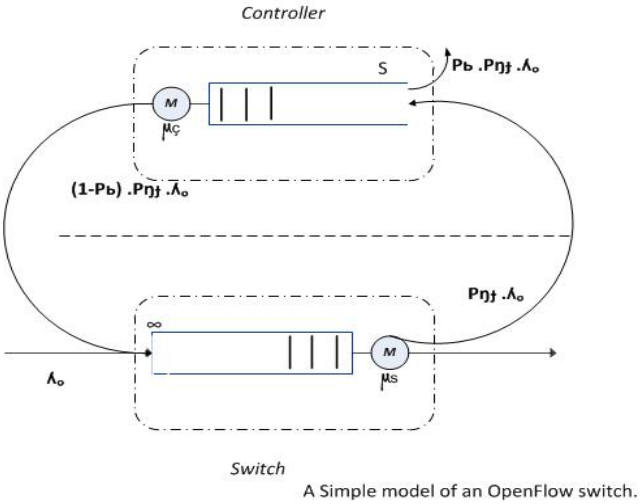

Fig. 5 A Simple model of an OpenFlow switch[14]

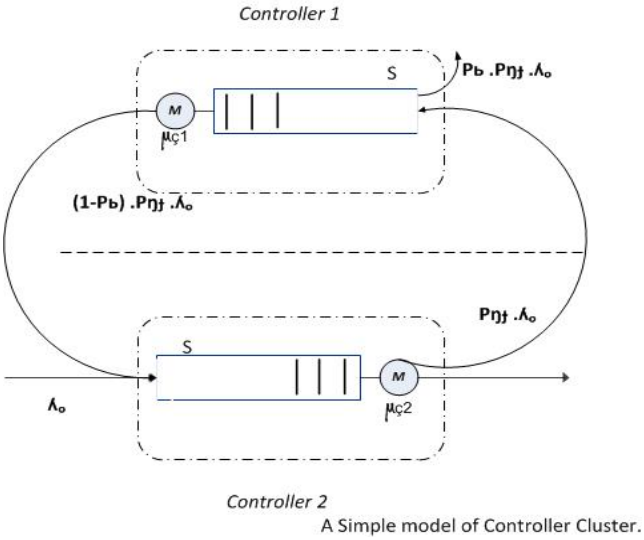

Fig. 6 A Simple model of controller cluster[14]

5. CONCLUSIONS AND FUTURE SCOPE

Software Defined Networking provides a fast, cross-hardware, and above all, inexpensive option for implementing network management solutions. This approach of using floodlight demonstrates the potential of SDN; however, Floodlight also required critical new features before is truly useful in a production environment. So the proposed system overcomes drawbacks of Floodlight such as very limited configuration management and lacks a high availability mechanism. Also if controller fails it can quickly paralyze the entire network. New service introduced provides functionality to floodlight to make it easy to take first steps and develop some expertise in SDNs to overcome its challenges.

This implementation can be modified so that it can re-use by existing or modules implemented in future to able to sync there data in cluster based topology of floodlight controllers

6. REFERENCES

[1] Douglas comer (ed.) 2014 "Internetworking with TCP/IP : principles, protocols, and architecture", (6th ed.) vol 1, Pearson education.

[2] Paul goransson ,chuck black(ed.), 2014,"software defined networks a comprehensive approach ", Elsevier publication

[3] Wenfeng xia, yonggang wen, , chuan heng foh, dusit niyato, and haiyong xie ,et.al., 2014, "A survey on software-defined networking ", in IEEE communications surveys & tutorials.

[4] S. rowshanrad , s.namvarasl, v. abdi, M. hajizadeh, M. keshtgary ,et.al., "a survey on sdn, the future of networking ", in journal of advanced computer science & technology.

[5] Nick feamster, jennifer rexford, ellen zegura,et.al., 2014.," The road to sdn: an intellectual history of programmable", ACM .

[6] Shah, s.a. faiz, j. ; farooq, m. ; shafi, a. ,et.al.," An architectural evaluation of sdn controllers ", in communications (icc), 2013 IEEE international conference on Budapest.

[7] Steven cheung, martin fong, Phillip Porras, Keith skinner & dr. vinod yegneswaran ,et.al.,"Securing the software-defined network control layer ", in proceedings of the 2015 network and distributed system security symposium (ndss). san diego,

[8] Ryan Wallner, Robert Cannistra ,et.al.,An SDN approach: quality of service using big switch's floodlightopen-sourcecontroller "

[9] Riggio, r. Gomez, k.m. ; Rasheed, t. ; Schulz-zander ,et.al.,"programming software-defined wireless networks" in IEEE network and service management (cnsm), 2014 10th international conference on, rio

[10] Roberto Riggio, Mahesh K. Marina, Julius Schulz–Zander,Slawomir Kuklinski, and Tinku Rasheed,et.al., "Programming Abstractions for Software–Defined Wireless Networks" IEEE Transactions on Network and Science Management,

[11] Natasha gude, Teemu Koponen,Justin Pettit,Ben Pfaff,Martín Casado,Nick Mckeown,Scott Shenker, et.al. ,2008 ," NOX: towards an operating system for networks. "

[12] David Erickson ,et.al., 2013, "The beacon openflow controller " in hotsdn '13 acm new york, ny, usa

[13] S .kaur , Japinder Singh and Navtej Singh Ghumman, et.al .,2014," network programmability using pox controller ", in ICCCS

[14] Jarschel, Michael & Oechsner, Simon & Schlosser, Daniel & Pries, Rastin & Goll, Sebastian & Tran-Gia, Phuoc ,et.al.,(2011), "Modeling and Performance Evaluation of an OpenFlow Architecture", Proceedings of the 23rd International Teletraffic Congress (ITC '11), pp1-7.

[15] Azodolmolky, Siamak & Nejabati, Reza & Pazouki, Maryam & Simeonidou, Dimitra, et.al., (2013) "An Analytical Model for Software Defined Networking: A Network Calculus-based Approach", IEEE Globecom.

[16] Yosr jarraya, taous madi, and mourad debbabi, et.al.,2014," a survey and a layered taxonomy of software-defined networking " in IEEE communication surveys & tutorials.