IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Conference on Electronics & Telecommunication Engineering 2018 (CETE-2018)

Thakur College of Engineering and Technology, Mumbai

Vol. 5, Special Issue 3, February 2018

# Implementing SDN to replace traditional networking paradigm to overcome limitations of traditional network

**Ashish Yadav[1], Pralhad Konnur[2], Sachin Singh [3], Kalawati Patil [4]**

Department of Electronics and Telecommunication

Thakur College of Engineering and Technology, Mumbai, India

**Abstract:** Traditional networks are vertically integrated and if any problem arises, it is required to troubleshoot every networking device. This affects the flexibility and halts the step to evolution. Suppose if a new economic venture is in planning to be established, a new networking system is provided. Due to lack of agility and flexibility, they cannot be upgraded instantly as they have distributed control plane. This introduces to a very intelligent network system termed as Software Defined Networking. SDN is an entry and evolutionary networking model that can modify the limitation of current network layout. The controller used here is Open Daylight which is the brain in such a network while Mini net emulation software provides the simulation switches. Here is an attempt towards integrating a proposed application with SDN system for greater observation and monitoring of a network.

**Keywords:** Network applications, Northbound interface, NOS, Mini net, Open Daylight.

## I. INTRODUCTION

Since a decade, it is found that learning and implementing traditional networking infrastructure has many limitations. India is country where every person requires internet connection, so, to speed up things, we require a totally different system where all the end goals are achieved [1]. Currently, SDN is deployed by giant multinationals such as Facebook, Google, AT&T and Verizon which has tremendously helped them to increase their growth [2]. It is predicted that by 2026, SDN may be efficiently installed and developed in India which is a real requirement now.

The following are the flaws of the current traditional networks which the plan to work is on:

1. Difficult to optimize:

Network operators find it difficult to introduce latest and fresh services that generate revenue and improve their infrastructures which are extremely expensive: data centres, networks (wide area), and enterprise area networks [1], [2].

2. Known problems:

Networks are continuously having serious problems with manageability, security, robustness and mobility. These issues have not been studied efficiently so far [1].

3. Capital costs:

Network capital costs are increasing in extremely large numbers and cost of operations are growing and putting tremendous pressure on network operators [1], [2].

4. Difficult to customize:

Even third parties and vendors are unable to provide customized cost effective solutions to address their customers' problems [3].

## II. OBJECTIVE

The basic design of SDN is capable of the utilization of modularity based mostly typical abstractions [1], quite the same as formal code engineering ways as shown in Fig 1. A typical network which is software defined mostly divides processes like virtual setup, resource distribution, congestion prioritization and traffic forwarding within the underlying hardware in 3 basic layers, namely, application, control, and information planes [3]. Every element of the planes has

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

Conference on Electronics & Telecommunication Engineering 2018 (CETE-2018)

Thakur College of Engineering and Technology, Mumbai

Vol. 5, Special Issue 3, February 2018

well outlined boundaries, a particular role, and conjointly relevant application programmable interfaces (APIs) to speak with neighboring planes [1]. A striking difference between the present distributed inspection of individual components and also the centralized network design is shown in Figure 1. The key bits of the framework entail the following: (i)

2 Implementing SDN to replace traditional networking paradigm to overcome limitations of traditional network

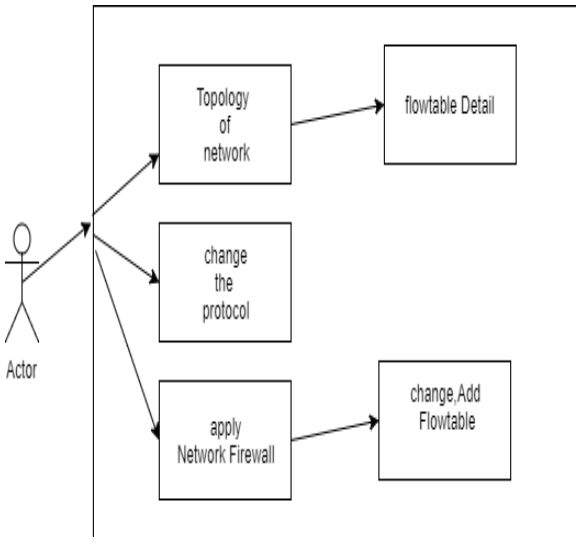
Data (forwarding) plane: The information plane could be a set of network parts which might be switches, routers, virtual networking instrumentation, firewalls, and then forth [2]. The only preliminary reason of information plane is to forward network traffic as with efficiency as supported with a definite set of forwarding rules schooled by the plane (management) [3]. Via SDN, the networking hardware is literally made dumb by removing forwarding intelligence and isolated configuration per network element and moving these functionalities to the management plane. Data transfer between information and management planes is achieved by southbound APIs. [1]. Now, the servers act as a southward communication protocol of selection complimented by a number of vendors also such as ONF [1].

(ii) Control plane: Management plane is liable for creating choices on however traffic would move from one explicit node to a different supported user application necessity and human action ensuing network policies to the information plane [3]. The central part of a sway plane is that an SDN controller interprets individual application necessities and business objectives like the necessity for traffic prioritizing, access management, information measure inspection and Quality of Service that area unit communicated to information plane elements [2],[3]. Supporting the dimensions of the network, there is a possibility that quite one SDN controller for extra redundancy. By introducing network programmability through the management plane, it becomes potential to control flow tables in individual parts supported network performance and repair necessities [1]. The controller offers a transparent and centralized read of the underlying network giving a suitable network monitoring tool to fine tune performance [1].

(iii) Application plane: Application plane includes network specific and business applications. Associate abstract scan of the underlying network is presented to applications via northbound application interface. The extent of abstraction would possibly embrace network parameters like delay, throughput, and convenience descriptors giving the applications a wider scan of the network [4]. Applications reciprocally request property between end nodes and once applications or network services communicate these wants to the SDN controller, it correspondingly configures individual network elements at intervals the knowledge plane for economical traffic forwarding [2],[3].

## III. FLOWCHART

Proposed application will be working with the help of API available on OpenDaylight website. This will be a single application which serve multiple software defined networking application including changing network protocol, showing present network with all the flow table detail, provision of implementing firewall security to each host individually.



IARJSET

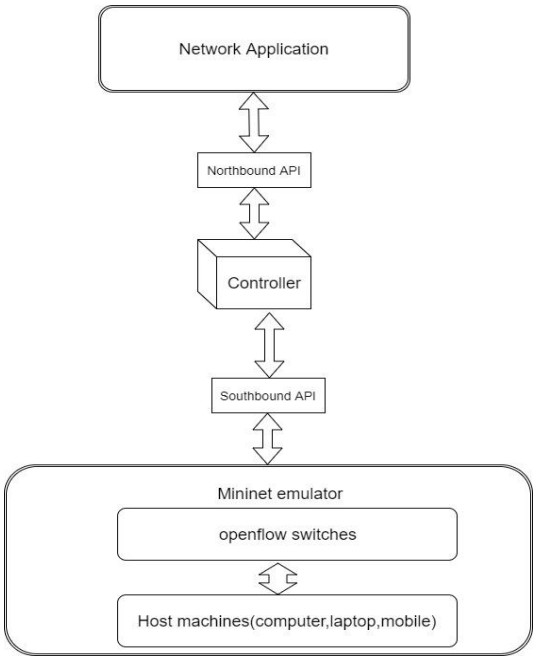
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Conference on Electronics & Telecommunication Engineering 2018 (CETE-2018)

Thakur College of Engineering and Technology, Mumbai

Vol. 5, Special Issue 3, February 2018




Proposed application will be working with the help of API available on OpenDaylight website. This will be a single application which serve multiple software defined networking application including changing network protocol, showing present network with all the flow table detail, provision of implementing firewall security to each host individually. Fig 6 shows the architecture of system in which our proposed application works. Controller has many predefined southbound API to interact with forwarding devices and host machines. This southbound api can be provoked by calling northbound API .Nothbound API can be called in many ways, but we are using REST Method to execute this API.

**Working of Application:** Proposed application is going to be operating with the assistance of API accessible on OpenDaylight web site. This may be one application that serves multiple code outlined networking application as well as dynamical network protocol, showing the network with all the flow table detail, provision of implementing firewall security to every host on an individual basis.

Controller manages the traffic (network flows) by manipulating the flow table at switches.

When packet arrives at switch, match the header fields with flow entries during a flow table.

If any entry matches, performs indicated actions and update the counters.

If it doesn't match, switch asks controller by sending a message with the packet header.

## IV. METHODOLGY

The main pillars of a SDN:

Controllers: The “brains” of the network, SDN Controllers supply a centralized read of the network, and change network directors to follow the underlying systems (like switches and routers) however the forwarding plane ought to handle network traffic [2].

Southbound APIs: Software-defined networking uses south APIs to relay info to the switches and routers “below.” OpenFlow, thought of the primary commonplace in SDN, was the first south API and remains mutually of the foremost common protocols [1]. Despite some considering OpenFlow and SDN to be one within the same, OpenFlow is simply one piece of the larger SDN landscape [1].

Northbound APIs: This type of network system uses north APIs to communicates with the applications and business logic. These facilitate network directors to programmatically form traffic and deploy services [1].

Mininet simulation software: Mininet could be a network simulator that creates a network of virtual hosts, switches, controllers, and links [1]. Mininet hosts run commonplace UNIX network computer code, and its switches support OpenFlow for extremely versatile custom routing and Software-Defined Networking [2],[3].

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

Conference on Electronics & Telecommunication Engineering 2018 (CETE-2018)

Thakur College of Engineering and Technology, Mumbai

Vol. 5, Special Issue 3, February 2018

## WORKING OF SDN USING MININET AND OPENDAYLIGHT

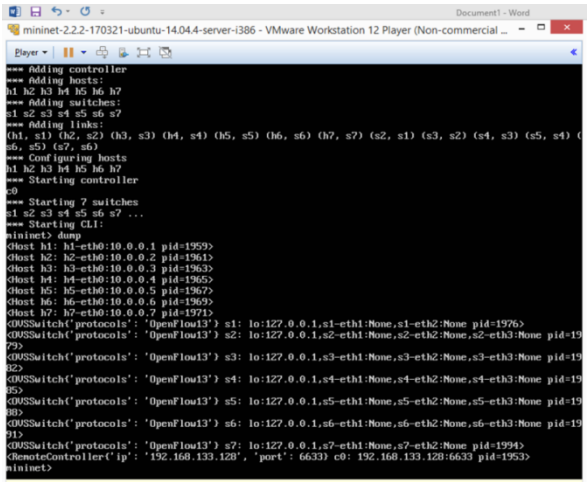



Fig 2. Assignment of different switches to the controller

First a network of OpenFlow switches and host is connected to a number of other switches depending upon the requirement. In Mininet software, we can see that switches are connected with each other and also host as seen in Fig 2. Now the basic network has been formed but to gain the advantage of software defined network, we have to control all the switches via a central controller [3]. For this sole purpose, OpenDaylight controller which is open source is used for connecting the Mininet environment and the controller. Configuration of controller IP address is done to use the Opendaylight server [3],[4]. After connecting Mininet and Opendaylight, we can confirm the connection by seeing Mininet environment in Opendaylight controller software. For this web server feature of controller is implemented [1]. This website works with the help of rest of the APIs. In topology section of controller website we can see the topology of Mininet environment [1],[2].

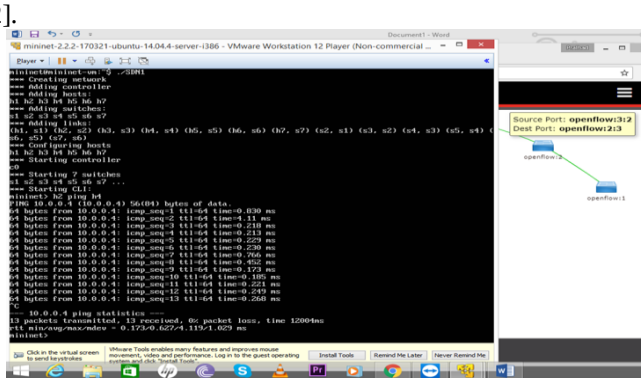



Fig 3. Representation of pinging speed statistics

From the above Fig 3. it is noticed that host h2 pings host h4. During the first trial, it takes 0.830 ms for the packet to reach host h4 from h2. In the following trials, this time frame goes on reducing.

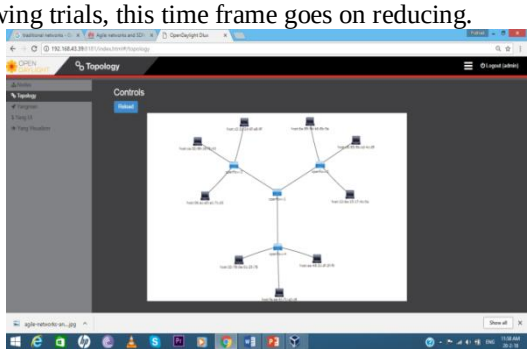



Fig 4. Topology of mininet network

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

Conference on Electronics & Telecommunication Engineering 2018 (CETE-2018)

Thakur College of Engineering and Technology, Mumbai

Vol. 5, Special Issue 3, February 2018

| Per Table | Per Flow | Per Port | Per Queue |
|---|---|---|---|
| Active Entries | Received Packets | Received Packets | Transmit Packets |
| Packet Lookups | Received Bytes | Transmitted Packets | Transmit Bytes |
| Packet Matches | Duration (Secs) | Received Bytes | Transmit overrun errors |
| | Duration (nanosecs) | Transmitted Bytes | |
| | | Receive Drops | |
| | | Transmit Drops | |
| | | Receive Errors | |
| | | Transmit Errors | |
| | | Receive Frame Alignment Errors | |
| | | Receive Overrun erorrs | |
| | | Receive CRC Errors | |
| | | Collisions | |

Fig 5. Basics stats table

Fig 5 explains the basic statistics table based on OpenFlow. It provides counter for incoming flows or packets. Information can be retrieved to the control plane.

## V. EXPECTED OUTCOME

(1) Monitoring and observation of network underline the controller by proposed application.
(2) Changing protocol of network and designing network accordingly by proposed application.
(3) Applying network firewall for each host in the network by proposed system.

## VI. RESULT AND DISCUSSION

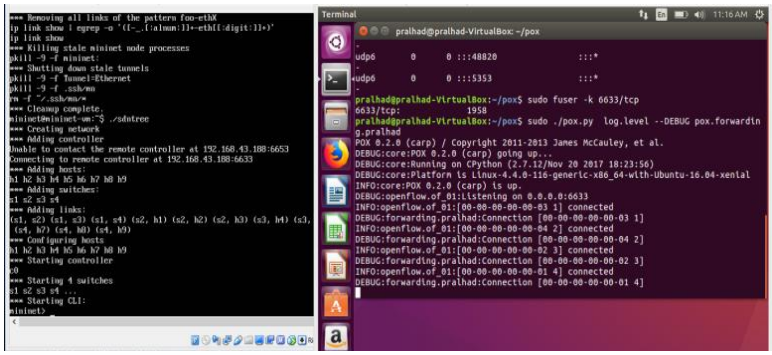
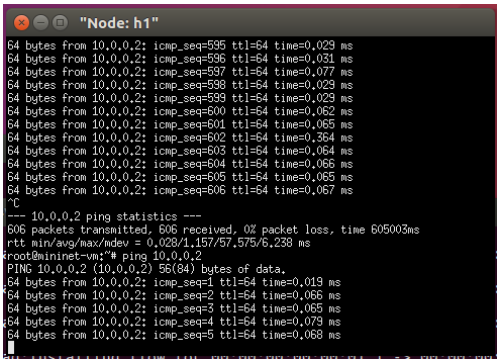



**Fig 6. Connection of switches with POX controller**

Here, the connection of a network comprising of four switches has been established with the POX controller.
The right side consists of the POX controller whereas the left side consists of the switches.
The results that have been achieved are illustrated below.



IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Conference on Electronics & Telecommunication Engineering 2018 (CETE-2018)

Thakur College of Engineering and Technology, Mumbai

Vol. 5, Special Issue 3, February 2018

Fig 7. Connection between host 1 and host 2

Here, host h1 terminal (10.0.0.1) pings destination host h2 at 10.0.0.2.

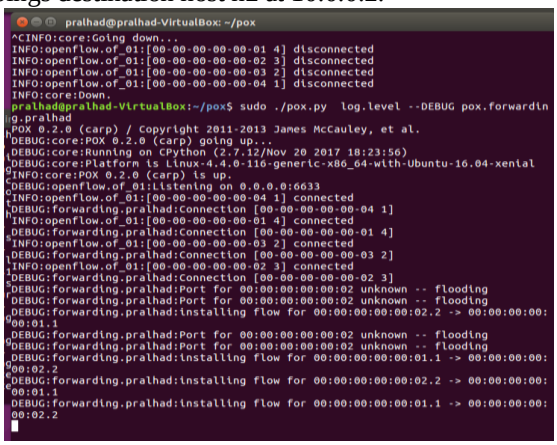



Fig 8. Installation of flow tables

Now, POX controller is installing the flow tables in the necessary switches i.e. switch s1 (00.00.00.00.00.01).
For this, the concept of flooding has been used and developed in the algorithm present in pralhad.py python file.

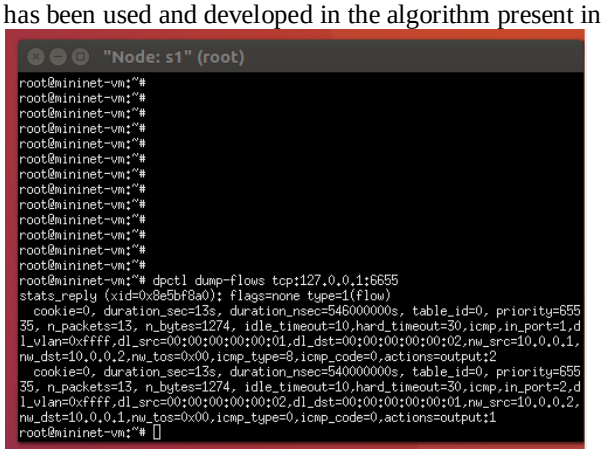



Fig 9. Packet details and host prediction

Here the output of logic of the algorithm can be clearly seen in switch 1 terminal.
In traditional networks, after the connection has been terminated, the flow tables are not dumped. This proves to be a major disadvantage as huge memory and storage is used. But the algorithm which has been developed retains the flow tables. This helps in the future prediction of destination hosts by sniffing the packet details.

While the POX controller was used to observe the minute details of the packets, the ODL controller is used for studying the GUI of the network

Then, the ODL GUI is opened as shown in the below figure.

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Conference on Electronics & Telecommunication Engineering 2018 (CETE-2018)

Thakur College of Engineering and Technology, Mumbai

Vol. 5, Special Issue 3, February 2018

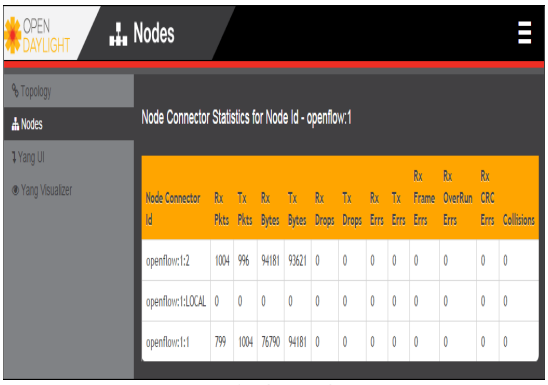



Fig 10. ODL GUI

## VII. CONCLUSION

Managing and controlling a large network has always been an issue for enterprise companies. This issue can almost be eliminated by the use of SDN. By the means of SDN, the agility and flexibility of a complex network can be easily cracked. The current study has been reflected in this paper.

## REFERENCES

[1] Fernando M.V. Ramos, "Software Defined Networking: A Course Survey", IEEE, Vol. 103, No. 1, January 2015.A Survey of Man In The Middle Attacks-by Senior Member, IEEE, Nicola Dragoni, and Viktor Lesyk.

[2] Bruno Nunes Astuto, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks", IEEE Communications Society, 2014, 16 (3), pp.1617 - 1634.Preventing and defending against Layer 2 attacks Understanding,.-by Yusuf Bhaiji.

[3] Wenfeng Xia, "A Survey on Software-Defined Networking", IEEE Communications Surveys & Tutorials (Volume: 17, Issue: 1, Firstquarter 2015).

[4] Manar Jammal," Software defined networking: State of the art and research challenges", Computer Networks,Volume 72, 29 October 2014, Pages 74-98.

[5] Harvey Newman, "High Speed Scientific Data Transfers usingSoftware Defined Networking", Austin, Texas—November 15, 2015.