ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

The Rise of Software-Defined Networking (SDN): A Paradigm Shift in Cloud Data Centers

Mohit Mittal

Uttar Pradesh Technical University, Lucknow, Uttar Pradesh, India

ABSTRACT: Software Defined Networking (SDN) is a novel network paradigm that intelligently manages the network's resources. The global view of a network is achieved through the separation of the data plane and network control. The abstraction of the network to its application and services is facilitated by the logically centralized control plane. SDN offers the potential to enhance network performance by utilizing the information exchanged between various layers in the network architecture and feedback control, which is facilitated by a global view of the network. SDN enables the network administrator to configure the network from a single point using software control. The entire network can be configured through programming and is dynamically optimized based on the network status. OpenFlow-based SDN technologies offer improved orchestration of network resources, are adaptable to evolving business requirements, and reduce operational complexities. The Network Function Virtualization (NFV) paradigm virtualizes the networking functions, similar to the hypervisors used for compute functions, and implements the entire suite of networking services as Virtual Machines (VMs). This article provides a comprehensive examination of the influence of Software-Defined Networks on data centers that are enabled by Cloud Computing. This article provides a performance comparison of ONOS and ODL data controllers on a variety of parameters, including throughput, jitter, and loss.

KEYWORDS: SDN, Data Centers, OpenFlow, Network Function Virtualization (NFV), SDN Controllers (ONOS, Open Daylight**)**

I. INTRODUCTION

The magnitude of the Internet is astonishing. Cloud computing has transformed the IT departments and ICT infrastructures of many firms by enabling the fast creation, hosting, and maintenance of corporate applications. It eliminated several costs, including startup, expansion, and operational expenses. Users saw extended wait times for time-sensitive services due to network congestion, over-provisioning, and routing complexity resulting from the proliferation of digital services and the growing number of connected devices and computers. Cloud service providers significantly falter in delivering context-aware, instantaneous, mobile, and immutable services to end users [1] [2].

In conventional networks, the management and implementation of infrastructure and services is a continuous process involving modifications to various network configurations, often using proprietary interfaces. The intricate and extensive protocols OSPF, BGP, and EGP govern packet forwarding. Implementing complex rules in conventional IP networking designs is difficult owing to the lack of appropriate network abstractions and a uniform representation of the global network state.

The network's design, as opposed to its programming, renders the deployment of new services slow and intricate [3][4]. A paradigm shift and new beginning in networking, "Software-Defined Networking (SDN)" eliminates control information from forwarding devices such as switches and routers [5][6]. By decoupling the control function from forwarding devices, SDN has transformed the management of large-scale networks. Prior to the advent of software-defined networking (SDN), manufacturers of individual switches provided the software for the control plane. This application executed many distributed algorithms, including the spanning tree protocol, to determine the routing laws and network design. This approach was connected with many drawbacks: Initially, there was a rise in complexity and a need for interoperability across manufacturers inside the distributed control plane, and these suppliers often had

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

divergent perspectives. Secondly, since network administrators were required to comply with the vendor-designated list of approved features, customization and innovation were almost unattainable. Third, a strong partnership with a certain vendor was essential for the flawless operation of all components. The equipment's total cost of ownership was excessive, necessitating significant investment in training and certification for network administrators to assure correct operation.

Software-Defined Networking (SDN) has enabled data plane forwarding devices to surmount several challenges, including those related to multi-tenant data centers, as well as concerns about flexibility, programmability, dynamic policies, upgrades, and innovation. Moreover, by reducing entry barriers, it facilitated the participation of new providers in the market. Enabling SDN transforms the formerly manual and resource-heavy process of dynamically delivering network services and enforcing regulations into a programmable and autonomous system, significantly enhancing manageability [7]. Data centers and wide area networks (WANs) have effectively used software-defined networking (SDN) due to its extensive endorsement in both academia and industry [8][9]. Businesses and operators are assessing SDN, while scholars are doing study on it. Numerous leading global organizations, such as Google Inc., are interlinking their numerous data centers worldwide via an internal backbone network based on software-defined networking. The utilization of the connections increased from 30–40% to almost 100%, significantly enhancing network performance as a consequence [9]. This notable achievement was facilitated by SDN's improved management of the programming and transmission of network flows.

The challenges of transparency and flexibility have been exacerbated by the rise of geographically distributed data centers in recent years. In response to the growing demand for datacenter workloads, the datacenter network must rapidly adapt to provide increased processing capacity. Moreover, owing to the erratic characteristics of datacenter workloads, operators needed to promptly identify network connections responsible for bottlenecks and reroute data traffic to circumvent costly overprovisioning.

Consequently, network operators were prepared to abandon these protocol implementations in favor of more manageable alternatives, such as designs that tackle the challenges of transparency and flexibility. Software-Defined Networking (SDN) proposes a centralized server, known as a controller, to entirely decouple the control plane from network switches, supplanting rigid and opaque network protocols. In the data plane, switches just relay packets according to directives from the controller.

The switches may alert the controller upon the arrival of certain packets, flow counters, and other relevant data. In such instances, the controller often instructs the switches on their actions. Figure 1 illustrates the distinction between Software-Defined Networking and traditional networking by demonstrating the integration of the control plane and switches into a single centralized controller, as opposed to depending on disparate, rigid protocols [10].

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

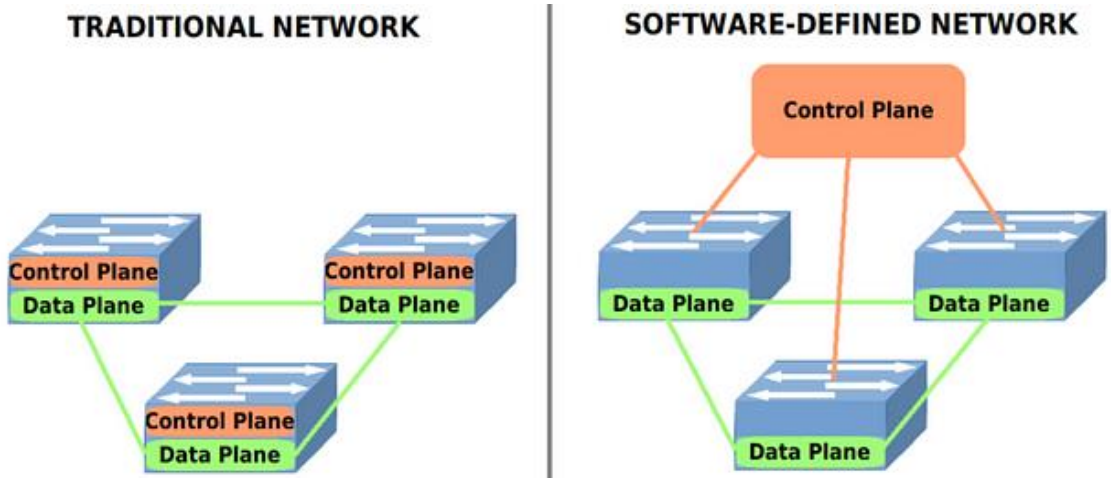

Figure 1: Traditional Network V/S Software-Defined Network

The data plane and the control plane are the two primary components of traditional networking apparatus. The control plane determines the device's capacity to forward received packets. In accordance with decisions made by the control plane, these packets are sent by the data plane. Increased velocity is required on the data plane, while enhanced programmability is essential on the control plane. In response to these varied requirements, the innovative idea of integrating all switches' control planes onto a single server has arisen, while the data plane remains on the networking device. The network's hardware does not govern its operation; rather, the software, protocol, and state function independently. Similar to x86 devices, this maintains a separation between software, operating systems, and hardware. Software-Defined Networking (SDN) centers on the concept of decoupling the control and data planes [10].

This article gives a thorough study of the impact of Software-Defined Networks on data centers powered by Cloud Computing. This article compares the performance of ONOS and ODL data controllers across a range of characteristics, including throughput, jitter, and loss.

II. SDN AND NFV

2.1SDN ARCHITECTURE OVERVIEW

Investigations into the researcher's SDN methodology have shown encouraging outcomes in cloud data centers. The emergence of data centers has facilitated the development of SDN. The control of the network under SDN may be modified dynamically. Open source systems may be automated with software-defined networking (SDN). Facilitated by SDN, commodity hardware simplifies programming significantly. SDN has three independent tiers: Application, Control, and Data. Abstraction and flexibility may be achieved by decoupling control plane functions from fundamental packet forwarding nodes. Secure Data Transfer (SDN) is a critical network technology currently in use.

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

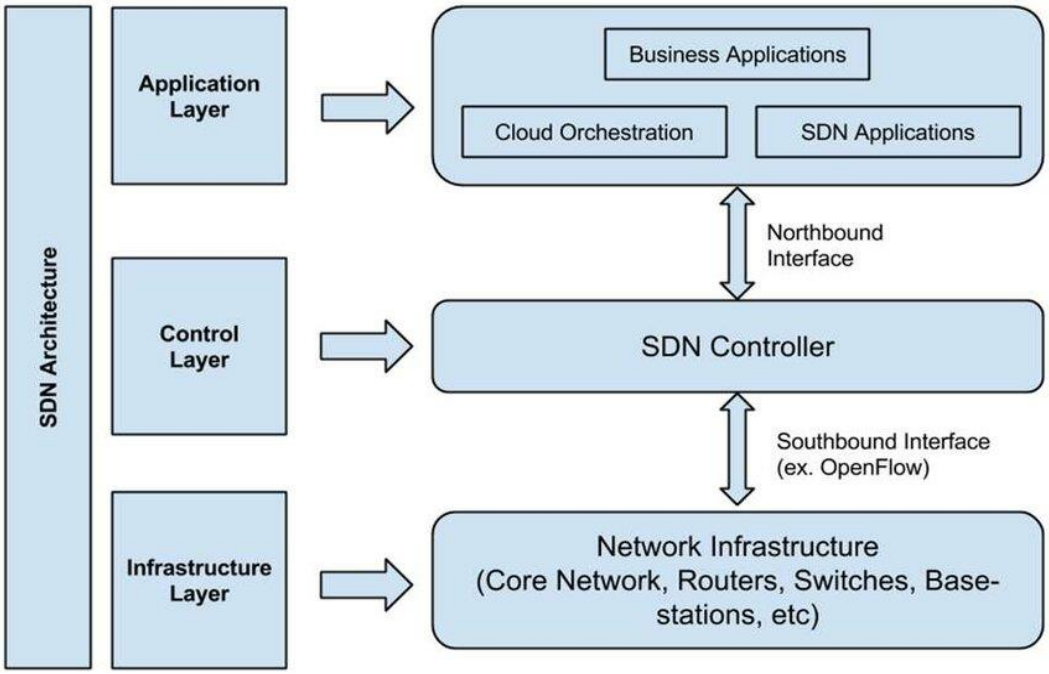

Figure 2: SDN Architecture

Figure 2 illustrates the open architecture of Software-Defined Networking (SDN) proposed by the Open Networking Foundation (ONF). The composition consists of infrastructure, control, and application layers. Through Software-Defined Networking (SDN), the network administrator may configure the whole network using software controls from a centralized place. The whole network may be modified via programming and dynamic optimization based on network conditions. Software-defined networking (SDN) enhances network performance by simplifying information exchange across various network layers and offering an overarching view of the network. Enhanced orchestration of network resources, improved adaptability of networks to evolving business requirements, and less operational complexity are all advantages of software-defined networking (SDN) systems using OpenFlow.

2.2 NFV

Telecommunications sector participants may improve service quality by strategically positioning "Network Functions/Services Chain (NF/SFC)" inside data centers and coordinating them across extensive interconnected networks. A network function (NF) takes packets at the input port(s) of a network element and transmits them via the output port(s) of the same element, perhaps applying specific changes. According to IETF RFC 3234, "Network Functions (NFs) are classified into two primary categories: (i) routing/forwarding and (ii) all other functions within the network layer and above." Internet Protocol routers execute the routing and forwarding functions of network functions and maintain current statistics counters. Route selection functions provide many benefits, including less latency, by opting for a certain path, such as the shortest route, between a source and a destination. These fundamental operations maintain a compact network layer, providing an extra advantage. As this functionality must be executed by all devices along a designated source-destination path, these qualities are necessary. Thus, the IP layer serves just as the basis for several additional protocols.

Frames facilitate the transmission of IP packets across many network layers. The "renowned hourglass model," seen in Figure 1.4 of IETF RFC 3234, posits that IP is situated above all other protocols. Due of the Internet's immense ubiquity and rapid evolution, basic NFs were inadequate. Consequently, network managers sought methods to enhance

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

network performance, isolation, and security via the integration of novel network services. Middleboxes, the secondary category of network functions, arose as a result. These advanced functions meticulously analyze and modify packet structures, rewriting fields across the whole header and, in some cases, even analyzing and editing the packet content.

III. LITERATURE SURVEY

Firewalls, load balancers, routers, and switches provide communication among different network components. To enable network devices to respond to various network events, administrators must formulate and enforce rules. These labor-intensive jobs often require physical effort and a small toolkit. Thus, establishing and managing networks, along with optimizing their performance, are arduous and susceptible to errors. Network providers must also address the problem of "internet ossification" [11]. The Internet, with its extensive and intricate deployment base, has become a vital infrastructure. The interdependence of Internet protocols and physical infrastructures renders upgrades to either a formidable or unfeasible endeavor. A suggestion has been made that network programmability [12] might resolve these issues. Software-defined networking (SDN) is an evolutionary paradigm in networking, facilitating the segregation of the control plane from the data plane. Consequently, the design and management of the network are simplified, resulting in enhanced flexibility and agility. In software-defined networking (SDN), the control plane governs the network's intelligence and decision-making, and the data plane manages traffic forwarding. Package Processors (P4), OpenFlow (OF), and ForCES are exemplars of protocol-agnostic programming frameworks. The Open Networking Foundation (ONF) enhanced the SDN and OF protocols. Software-defined networking (SDN) offers several potential applications across various networks, ranging from data centers (DCs) to wireless 5G networks. This investment has prompted a sector-wide transition to software-defined networking (SDN). Projections suggest that the SDN market will grow at a compound annual growth rate of 26.8% from 2018 to 2023.

Conventional IP networks, also referred to as legacy networks, are widely used despite the rapid advancement of software-defined networking (SDN). More intricate and extensive SDN implementations need the synchronized efforts of several distributed control systems, unlike the first SDN models that assume a single controller manages the network. Initiatives are underway to enhance comprehension of the control plane in its entirety. Controller placement, localization, and network state consistency are distributed Software-Defined Networking solutions reliant on the preceding network domain [13-15].

It must address more issues than only SDN security. Although major firms like as Google have embraced SDN early and the industry is thriving, small and medium-sized enterprises remain apprehensive about it. As SDN matures and achieves broad implementation, it becomes an appealing target. Moreover, in contrast to traditional networks, SDN creates additional vulnerabilities for intruders [16].

While additional obstacles to SDN adoption exist, technological issues remain predominant. The company's prospective growth and profitability are at risk. When current network infrastructures operate well, corporations are reluctant to invest in new technology and retrain their technical personnel. There are no established methods used in production to facilitate the transition to a new network. The move from conventional networks to Software-Defined Networking (SDN) heightens the probability of service failures and breaches of Service Level Agreements (SLAs) [17].

The control plane makes decisions (e.g., TTL updates, CRC verification, QoS queuing, etc.), whereas the data plane executes and transmits messages based on those decisions. The data plane of hSDN networks may support both traditional Ethernet switches and Layer 3 (L3) routers, contingent upon the deployment strategy. The data plane of SDN consists of hardware components that do not communicate data with one other. The controller regulates network traffic by establishing connections to SDN-enabled devices using a Southbound Interface (SBI). The present configuration includes SDN controllers that support OpenFlow, ForCES, and several other SBI protocols. SNMP and NETCONF may be used for low latency and large bandwidth, despite not being its intended purpose [17]. Hybrid data planes may be activated by default in some control protocols as well.

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

A legacy agent may also be used to update an SDN node. Tilmans and Visicchio [18] propose an SDN backup architecture that relies on IGP. Meanwhile, until the establishment of a permanent channel, the controller utilizes the OF protocol. To provide backup paths, the legacy agent uses the OSPF protocol. Upon the occurrence of a failure, agents utilizing the Inter Gateway Protocol (IGP) rely on local routing data to promptly re-establish network connectivity. As a result, network disturbances are rectified more swiftly than in a network that just employs SDN.

Caesar et al. [19] provide instances of implementing the controller-based paradigm inside a BGP context. They advocate for an RCP, or "routing control platform," a hardware device that operates akin to a central controller. Every router in the network is linked via an iBGP connection. After RCP has determined a route for each router inside the domain, it may disseminate that route across the network via iBGP. Due to its scalability and rapid response capability, RCP is well suited for several applications in the event of a network connection failure.

Numerous studies indicate that hybrid SDN networks may be susceptible to security breaches owing to the presence of numerous control planes. A hybrid network upgrade may result in inconsistent forwarding. Acquiring statistics from a legacy switch over an unencrypted connection may reveal a security vulnerability in an SDN controller's system. Employing auto-configuration methodologies in hSDN networks may assist in assessing the danger of ARP-based spoofing attacks, which represent an additional vulnerability in hSDN networks [20].

The integration of conventional networking with Software-Defined Networking (SDN) presents inherent security vulnerabilities associated with both networking paradigms. The following section will delineate and classify the many vulnerabilities of both traditional and SDN-enabled networks, offering a thorough overview of the attack surface of hybrid systems. Network security may be jeopardized by end-host malware and defective software-defined networking (SDN) applications. Software-defined networking applications (SDNs) may disturb standard network operations and monitoring by using the controller's northbound interface. Due to the increased probability of software defects in proprietary systems, they provide the greatest risk to the network administrator. There exists a possibility that some SDN software inside third-party repositories may include malicious malware that might jeopardize the network's integrity.

Two instances of software-defined networking (SDN) controllers that facilitate several applications in sharing network traffic, such as for load balancing, monitoring, network address translation, and packet filtering, are OnOS and Open Daylight. Inadequate coordination of the forwarding rules provided by these applications may result in unforeseen traffic routing or security vulnerabilities [21].
The objective of denial-of-service (DoS) attacks is to disrupt the normal operation of a targeted network or computer system by using hacked devices to generate harmful data. The use of the SDN controller in hybrid systems offers distinct advantages for safeguarding against such attacks. To safeguard against denial-of-service attacks, the controller may programmatically enforce traffic filtering rules on its managed SDN devices. Denial-of-service attacks may aim at either SDN controllers or switches.

IV. PERFORMANCE ANALYSIS OF OPENFLOW CONTROLLERS

4.1 OpenFlow Protocol

OpenFlow is a northbound communication protocol used in software-defined networking (SDN) design. It lets network devices connect to the forwarding plane. It is possible to add a flow rule to the flow table of OpenFlow-enabled switches by talking to the sending data elements in the infrastructure layer. This lets the SDN controller set up many different network services, such as L2 switching, L3 switching (routing), load sharing, traffic tracking, and a lot more.
The OpenFlow switch standard (www.opennetworking.org) is where you can find the rules for OpenFlow Logical switch. There are a group table and one or more flow tables in the OpenFlow logical switch that make it easier to look up and send packets. When the switch and processor talk to each other outside of band, they can do so over a secure connection. The OpenFlow parts can be seen in Figure 3.

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

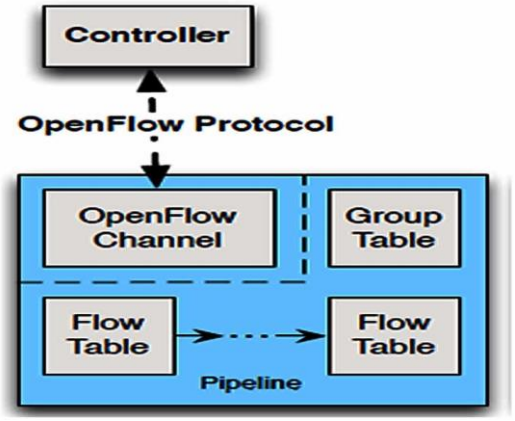

Figure 3 OpenFlow Components

The OpenFlow switch protocol enables the SDN controller to alter flow tables in two manners: in reaction to incoming packets and in anticipation. Figure 4 illustrates that each flow table inside the switch has a collection of flow entries. These elements include match fields, priority, counters, instructions, timeouts, cookies, and flags that may be assigned to the matched packets.

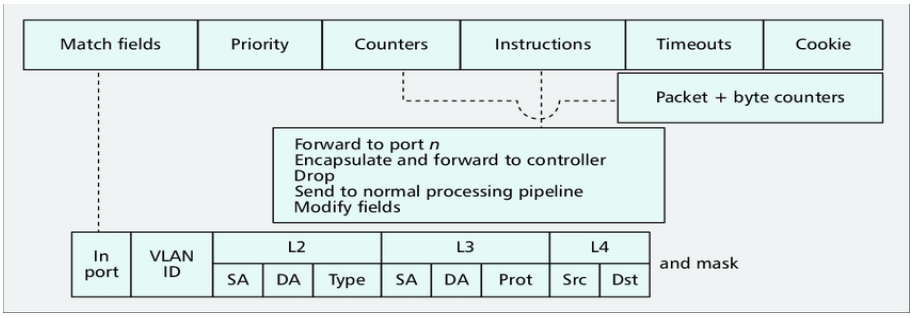

Figure 4: Flow Entry Fields of Flow Table in OpenFlow Switch

When a packet aligns with a flow entry, the flow entry will provide a series of instructions that, when executed, will modify the packet, action set, and/or pipeline processing of the flow. The action set delineates the procedures for routing, modifying, and processing packets in a pipeline for the specified tables.

The segregation of the control and forwarding planes is crucial in software-defined networking (SDN). Routers and switches constitute the forwarding plane, whereas the control plane determines the routing of packets. The initial capital needed to establish a Software Defined Network (SDN) is substantial. A hybrid SDN architecture may use existing network equipment. Programming in hybrid SDN deployments is confined to the SDN domains. Hybrid SDN is compatible with several legacy protocols, not only OpenFlow. The traditional network maintains control plane forwarding protocols, but the controller may use hybrid SDN models to direct packets to legacy switches. Hybrid networks may mitigate controller failure by using the established convolutional method. Hybrid SDN amalgamates the advantages of both pure SDN and convolutional switches.

The ability of SDN to decouple the control plane from the forwarding plane facilitates a more streamlined network design. An SDN controller is a crucial element for the proper operation of a software-defined network (SDN).

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

Additionally, the list includes three additional controllers: Ryu, Floodlight, OpenDaylight, and POX. The open-source POX controller facilitates rapid development and experimentation. The Python interface simplifies the process of topology discovery. POX's design enables connectivity to software-defined networking (SDN) switches using the OpenFlow or OVSDB protocols. Mobile device applications for network management and control may be developed using the open-source software-defined networking controller Ryu. The Mininet emulator serves as a testing platform for the Hybrid SDN. Mininet emulates a genuine network by running actual kernel, switch, and application code on a lightweight virtualization platform. Minor network. The study considers several network topologies and components. The findings are beneficial for hybrid SDN controllers with varying quality of service demands.

4.2 Mininet

A software-defined network (SDN) controller and the network emulator Mininet may operate together to build a virtual network. The network testbed on Mininet may be used to test OpenFlow applications. Because of the assistance it provides, it enables several developers to work on the same topology concurrently. With its included command line interface (CLI), testing and debugging of the network may be done easily. Building and exploring networks is made easier with Python API support. Mininet outperforms its forerunners in terms of speed, scalability, and bandwidth provision. It simplifies the process of creating a functional model of the actual network for programmers. Mininet eliminates the need to use an actual network for testing topologies that are complicated. By using the Mininet, one may personalize the rules for packet forwarding. Mininet offers OVS switches and controllers for sale. Mininet makes it possible to install additional software, such the D-ITF package. Every one of the Mininet's virtual hosts is affected by the fact that it only has one Linux kernel.

4.3 Experimental Set UP

Figures 6 and 7 demonstrate the results of evaluating the controllers' performance using two different topologies: linear and tree. The Mininet virtual machine is operated on Ubuntu 14.04 Lts. Connected to the SDN controller, the SDN switches already have flow tables set up. The test included recording the outcomes of packets transported between conventional and SDN switches.

.

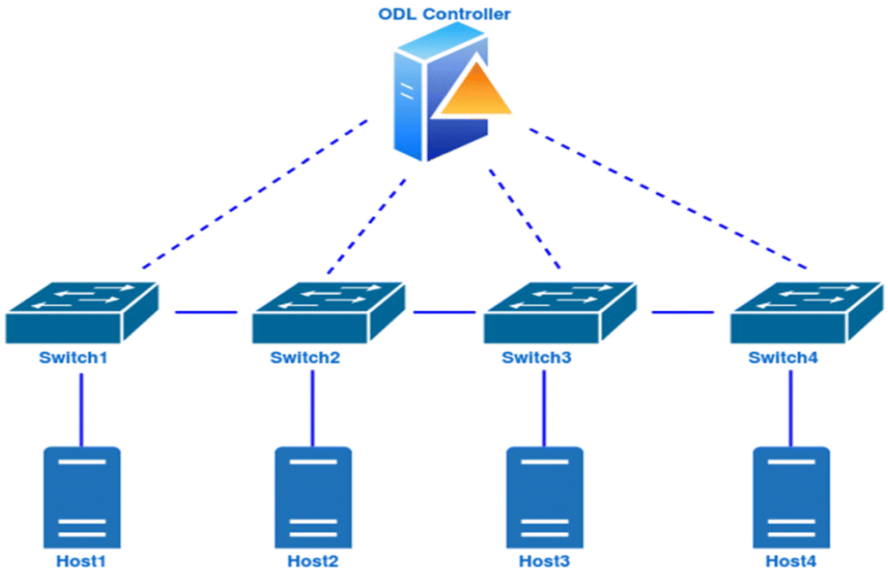

Figure 5: Linear Topology

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

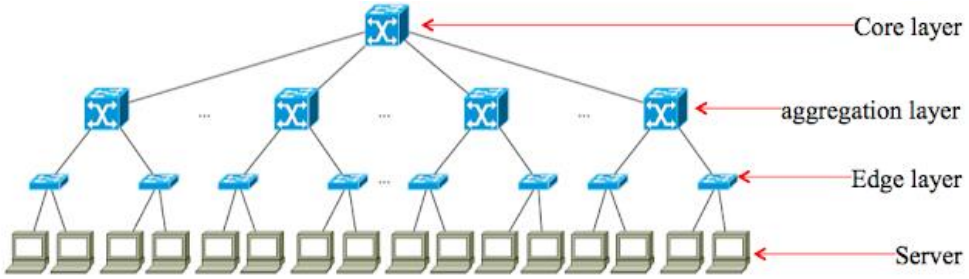

Figure 6: Fat Tree Topologies

The architectural design of the fat tree topology is illustrated in Figure 7. Software-defined networking switches occupy levels three and four of the design, while conventional switches occupy levels two and three. The desired results were achieved by transferring packets from traditional switches to SDN switches. The Open Day Light (ODL) and ONOS controllers were utilized to operate the topology during the test. There were a few switches that were configured as traditional switches and a few more that were OpenFlow-enabled in each architecture.
The following are the QoS network parameters:

- Latency: Time in milliseconds that it takes a packet to travel from its origin to its destination over a network.
- Jitter: Latency fluctuation.
- Loss: The percentage of packets that was lost in transit. The packet loss ratio is a percentage ofthe total number of packets sent and received.
- Throughput: The amount of data a network can transport in a given amount of time.

Table 1 and table 2 displays the results. For testing, 20 GB of packets from conventional switches were transmitted to SDN switches through UDP.

Table 1: Comparison between ONOS and ODL controller For Linear tree Topology

Controller	Time Taken in seconds	Amount of data Transferred In Gb	Jitter in ms	Total data grams	Loss Count	Loss Percentage
ODL	862	20	0.025	14608732	124	0.00085
ONOS	862	20	0.017	14608732	56	0.00032

Table 2: Comparison between ODL and ONOS controller For Fat tree Topology

Controller	Time Taken in seconds	Amount of data Transferred In Gb	Jitter in ms	Total data grams	Loss Count	Loss Percentage
ODL	1226	20	0.60	14608732	7304366	50
ONOS	1203	20	0.21	14608732	292174	48

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

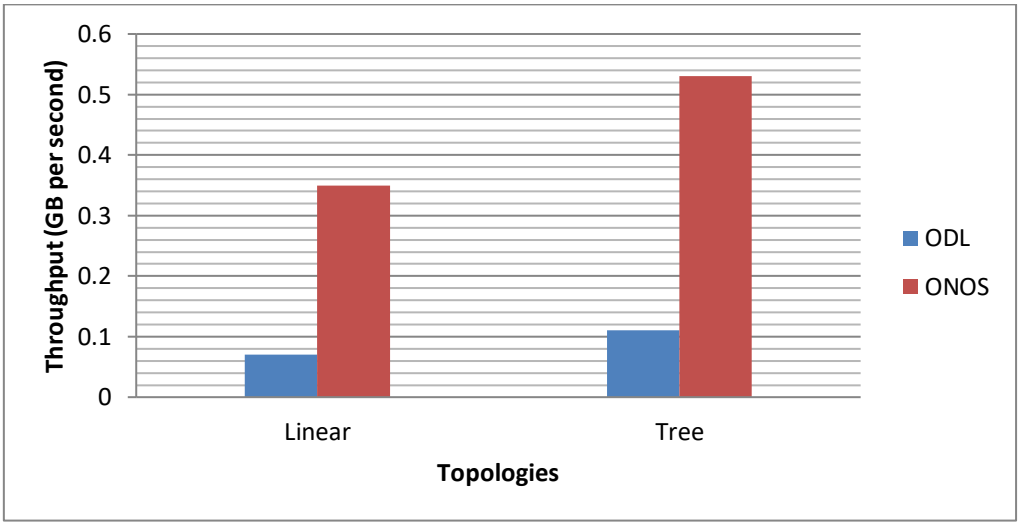

Figure 7: Throughput Comparison

From table 1 and 2 , it is clear that ONOS is better than ODL as far as the jitter and loss parameters are concerned. From figure 8, it is clear that the throughput of ONOS is higher than ODL for both linear and tree topology.

V. CONCLUSION

SDN allows the network administrator to manage the network from a centralized location using software management. The whole network may be programmed for configuration and is dynamically optimized according to its state. OpenFlow-based SDN solutions enhance network resource orchestration, respond to changing business needs, and diminish operational difficulties. The Network Function Virtualization (NFV) paradigm virtualized networking functions, akin to hypervisors for computational tasks, and deploys the whole array of networking services as Virtual Machines (VMs). This article offers a thorough analysis of the impact of Software-Defined Networks on data centers facilitated by Cloud Computing. This article presents a comparative analysis of the performance of ONOS and ODL data controllers across many characteristics, such as throughput, jitter, and packet loss. It is evident that ONOS surpasses ODL regarding jitter and loss characteristics. The throughput of ONOS surpasses that of ODL in both linear and tree topologies.

REFERENCES

[1] Xia, W.F., Foh, C.H., Xie, H.Y., Niyato, D., Wen, Y.G., "A Survey on Software-Defined Networking," (In Submission) IEEE Journal of Communications Surveys and Tutorials, May 2013.

[2] N. McKeown, T. Anderson, H. Balakrishnan, et al., "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69-74, 2010.DOI: 10.1145/1770399.1770416

[3] Mendonça, M.,Nunes Astuto, B., Nguyen, X.N., Obraczka, K., Turletti, T., "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," (In Submission) Networking and Telecommunication, HAL, INRIA, June 2013.

[4] T. Benson, A. Akella, and M. Zhang, "Network Traffic Characteristics of Data Centers in the Wild," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 266-277, 2010. DOI: 10.1145/1736821.1736843

[5] Duan, Q., Yan, Y.H., Vasilakos, A.V., "A Survey on Service-Oriented Network Virtualization toward Convergence of Networking and Cloud Computing," IEEE Transactions on Network and Service Management, vol.9, no.4, pp.373–392, December 2012

ISSN(Online) : 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)|Impact Factor: 1.672|

Vol. 2, Issue 8, August 2013

| DOI:10.15680/IJIRSET.2013.0208078 |

[6] Khan, A., Zugenmaier, A., Jurca, D., Kellerer, W., "Network Virtualization: a Hypervisor for the Internet?" IEEE Communications Magazine, vol.50, no.1, pp.136–143, January 2012
[7] Macapuna, C.A.B., Rothenberg, C.E., Magalhaes, M.F., "In-Packet Bloom Filter-Based Data-Center Networking with Distributed OpenFlow Controllers, "IEEE 2010GLOBECOM Workshops, pp.584– 588, 6–10 December 2010.
[8] Thanh, N.H.,Nam, P.N.,Truong, T.-H.,Hung, N.T.,Doanh, L.K., Pries, R., "Enabling Experiments for Energy-Efficient Data-Center Networks on an OpenFlow-Based Platform,"Proceedings,2012 Fourth International Conference on Communications and Electronics (ICCE), pp.239–244, 1–3 August 2012
[9] Fang, S.,Yu, Y., Foh, C.H., Aung, K.M.M., "A Loss-Free Multipathing Solution for Data Center Network Using Software-Defined Networking Approach," IEE
[10] Pries, R., Jarschel, M., Goll, S., "On the Usability of OpenFlow in Data Center Environments,"Proceedings,2012 IEEE International Conference on Communications (ICC), pp.5533–5537, 10–15 June 2012.
[11] J.S. Turner, D.E. Taylor, Diversifying the internet, in: GLOBECOM '05. IEEE Global Telecommunications Conference, 2005, vol. 2, pages 6. pp. –760.
[12] Jacobus Van der Merwe, Charles Kalmanek, Network programmability is the answer, in:Workshop on Programmable Routers for the Extensible Services of Tomorrow (PRESTO2007), Princeton, NJ, 2007.
[13] Stefan Schmid, JukkaSuomela, Exploiting locality in distributed SDN control, in:Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in SoftwareDefined Networking, 2013, pp. 121–126.
[14] Yury Jimenez, Cristina Cervello-Pastor, Aurelio J. Garcia, On the controller placement fordesigning a distributed SDN control layer, in: 2014 IFIP Networking Conference, IEEE,2014, pp. 1–9.
[15]Sushant Jain, Alok Kumar, SubhasreeMandal, JoonOng, Leon Poutievski, Arjun Singh,SubbaiahVenkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, UrsHolzle, StephenStuart, Amin Vahdat, B4: Experience with a globallydeployed software defined wan, in:Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13,2013, pp. 3–14, http://dx.doi.org/10.1145/2486001.2486019
[16]SakirSezer, Sandra Scott-Hayward, PushpinderKaurChouhan, Barbara Fraser, DavidLake, Jim Finnegan, NielViljoen, Marc Miller, NavneetRao, Are we ready for SDN?Implementation challenges for software-defined networks, IEEE Commun. Mag. 51 (7)(2013) 36–43.
[17] R. Enns, M. Bjorklund, J. Schoenwaelder, A. Bierman, Network Configuration Protocol(NETCONF), 6241, IETF, 2011, http://www.rfc-editor.org/rfc/rfc6241.txt.
[18] Olivier Tilmans, Stefano Vissicchio, IGP-as-a-backup for robust SDN networks, in: 10thInternational Conference on Network and Service Management (CNSM) and Workshop,2014, pp. 127–135, http://dx.doi.org/10.1109/CNSM.2014.7014149.
[19] Matthew Caesar, Donald Caldwell, Nick Feamster, Jennifer Rexford, AmanShaikh,
Jacobus van der Merwe, Design and implementation of a routing control platform, in:Proceedings of the 2nd Conference on Symposium on Networked Systems Design &Implementation-Volume 2, USENIX Association, 2005, pp. 15–28,http://dx.doi.org/10.5555/1251203.1251205.
[20] PankajBerde, MatteoGerola, Jonathan Hart, Yuta Higuchi, Masayoshi Kobayashi, ToshioKoide, Bob Lantz, Brian O'Connor, PavlinRadoslavov, William Snow, Guru Parulkar,ONOS: Towards an open, distributed SDN OS, in: Proceedings of the Third Workshop onHot Topics in Software Defined Networking, HotSDN '14, Association for ComputingMachinery, New York, NY, USA, 2014, pp. 1–6,http://dx.doi.org/10.1145/2620728.2620744.
[21]Jan Medved, Robert Varga, Anton Tkacik, Ken Gray, Opendaylight: Towards a model driven SDN controller architecture, in: Proceeding of IEEE International Symposium on aWorld of Wireless, Mobile and Multimedia Networks 2014, 2014, pp. 1–6,http://dx.doi.org/10.1109/WoWMoM.2014.6918985.