ISSN 2347 - 3983
Volume 10. No.3, March 2022
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter011032022.pdf
https://doi.org/10.30534/ijeter/2022/011032022

Dynamic Flow Reduction Scheme Using Two Tags Multi protocol Label Switching (MPLS) in Software Define Network

Adeniji Oluwashola David
University of Ibadan, Nigeria , od.adeniji@ui.edu.ng, sholaniji@yahoo.com[2]

Received Date February 04, 2022 Accepted Date : February 25, 2022 Published Date : March 07, 2022

ABSTRACT

Software-Defined Networking (SDN) is a technique that breaks the vertical integration of networks by separating the networks' control logic from the underlying routers and switches that forward the traffic. The performance of SDN network nodes may be degraded due to flow entries that are frequently updated and stored in the network elements. The control plane is flooded by the configuration messages thereby resulting in a trade-off between the number of configuration messages and number of permanent flow entries in the network.The focus of the study is to develop a scheme that will reduce the number of flow entries of IPv6 packets using two multiprotocol label switching tags in Software Defined Networks. An experimental testbed was developed using openvswitch, Ryu controller, OpenFlow version 1.3, IPERF3 and Wireshark. A Mininet kernel was built on Ubuntu Linux 18.04.1 operating system. The IPv6 Hybrid Permanent Flow (HPF) scheme and Non-Permanent Flow (NPF) was used to reduce the number of permanent flow entries and the number of configuration messages from the controller to the Ternary Content Addressable Memory (TCAM) of the Openswitch.The Result shows that IPv6 HPF scheme had a reduction of 63% as against NPF in the network for the dependency of switch per region while 33% reduction in flow entries of the network for the dependency of host per switch. The result can be used by Internet Service Providers (ISPs) for real-time network flow that will provide handling of an unpredictable surge of big-data transmission across network backbone.

Key words : Open Flow, Ternary Content Addressable Memory (TCAM), Flow Entries, Software Defined Network, Multi- Protocol Label Switching.

1. INTRODUCTION

In the year 2020 and future years, 70% of the world population would be dominated by mobile users [1]. Big data is everywhere and it can help organizations and any industry in many different ways. Nowadays there is so much data that existing hardware and software are not able to deal with. These data are created at such a high speed and has become too complex and dynamic to process, store, analyze, transmit and manage with traditional data tools and network devices. [2]. For instance, Facebook has 1.038 billion active users per day [3] with attendant network packets becoming plainly tremendous. Big data transmission cannot be accomplished without accurate network control due to enormous data and complicated computation. Hence, data transmission on the Internet requires a software defined network for granular policy routing. Internet Service Providers (ISPs) have devised a means of tackling network challenges that emanate from the routing of the packet traffic which occurs in their network as a result of Big Data with the use of Software Defined Networks (SDN). Current networks are vertically integrated. The control plane (that decides how to handle network traffic) and the data plane (that forwards traffic according to the decisions made by the control plane) are bundled inside the networking devices, reducing flexibility and hindering innovation and evolution of the networking infrastructure. Also, the transition from IPv4 to IPv6, started more than a decade, due to the inertia of current IP networks.). SDN has advantages from the global perspective of the network and it can extremely facilitate big data transmission, storage, and processing. For instance, dynamic resource allocation to big data applications to meet service level agreements can improve the performance of SDN-based data centers, compared to traditional data centers. In addition, the central controller can acquire information from multiple layers, share them among different layers so that the network performance can be improved. SDN uses OpenFlow protocol to instruct the network elements on the stepwise processing of packets transiting the network by installing a flow entry in the TCAM of each switch on the forwarding path for each pair of source and destination packet. The performance of the SDN network nodes may be degraded since flow entries are frequently updated, and huge transmission packet requests demands for high number of flow entries to be installed in the

openvswitch. More processing is needed in the network controller when the network size is enlarged since incoming packets are forwarded as decisions from the controller.

This paper is organized as follows: In section 1: Introduction on software defined networking architecture, section 2 literature review, Section 3 Research Methodology and Section 4 Result for discussions. Section 5: Provides the limitations associated with the analysis of the use of two MPLS label in software defined networking and in Section 6 the conclusion of the paper was presented.

2. LITERATURE REVIEW

First, Software Defined Network breaks the vertical integration by separating the network's control logic (the control plane) from the underlying routers and switches that forward the traffic (the data plane) as the simplified view of this architecture is shown in Figure 1. Second, with the separation of the control and data planes, network switches become simple forwarding devices and the control logic is implemented in a logically centralized controller (or network operating system), simplifying policy enforcement and network reconfiguration and evolution.

The separation of the control plane and the data plane can be realized by means of a well-defined programming interface between the switches and the SDN controller. The controller exercises direct control over the state in the data plane elements via this well-defined application programming interface (API). The most notable example of such an API is OpenFlow [4]. An OpenFlow switch has one or more tables of packet handling rules (flow table). Each rule matches a subset of the traffic and performs certain actions (dropping, forwarding, modifying, etc.) on the traffic. Depending on the rules installed by a controller application, an OpenFlow switch can be instructed by the controller to behave like a router, switch, firewall, or perform other roles (e.g., load balancer, traffic shaper, and in general those of a middlebox). An important consequence of the SDN principles is the separation of concerns introduced between the definition of network policies, their implementation in switching hardware, and the forwarding of traffic. This separation is key to the desired flexibility, breaking the network control problem into tractable pieces, and making it easier to create and introduce new abstractions in networking. This abtraction can simplify the network management and facilitate network evolution for innovation. However *there are no adequate provision for quality of service (QoS) in OpenFlow using Flow Label to reduce bits required as a field to match packets in internet protocol six (IPv6) [5]*.

2.1. Internet Protocol Version Six (IPv6)

The trillions of new IPv6 addresses will meet the internet demand for the foreseeable future.This communication protocol allows mobile device users to move from one network to another, while maintaining a permanent Internet Protocol address. The dual role played by Internet Protocol (IP) addresses imposes some restrictions during mobility, because when a terminal moves from one network (IP subnet) to another, it will maintain the IP address of the node that is associated with in order not to change the identifier in the upper layers during ongoing sessions. However, the resource management of Multihoming in nested mobile network raises new issues in the host mobility of ipv6 network.in [6]. Different methods have been employed to secure and protect the shared and sensitive data. *However, the* significant roles of encryption algorithms are numerous and essential in information security[7] ***in*** Comparative Study of Symmetric Cryptography Mechanism . The prediction of incoming attacks is achieved in a timely manner which enables security professionals to install defense systems in order to reduce the possibility of such attacks in *Zero Day Attack Prediction with Parameter Setting Using Bi Direction Recurrent Neural Network in Cyber Security*. [8]. Because attacks are different from legitimate user' activities, these differences can be easily spotted by these systems.

2.2. Packets Flow Entry

A packet is sent to ask the instruction to the controller if the packet does not match any flow in the flow table present on each node in the transmission path. There are two typical conventional schemes for flow table design. First, static flow entries are permanently stored in the flow table of each switch. These flow entries are called Permanent Flow Entries. Path computation for each pair of source and destination is prepared in advance[9.10]. The request of path setting from the controller is unnecessary. However, each switch requires a large number of flow entries The size of ternary content-addressable memory (TCAM) in the switch is limited, this scheme cannot support a large network size. Second, flow entries are required to be installed only when the first packet of a new request arrives at a switch..a request arrives at a switch, the switch sends an inquiry as a packet_in to the controller.The concept of Using Neural Network for Intrusion Detection in the configuration [11]. The controller computes a path for the request and replies a packet_out back to the switch to install flow entries into the flow table. The reply packet_out will be called a Configuration Message hereafter.Hierarchical Cluster Head Election Using Exponential Decay Function Prediction [12]. A flow entry that is installed from the configuration message is called a Non-Permanent Flow Entry. A large network size cannot be supported by this scheme due to a CPU overloaded status in the controller. In addition, the control plane is flooded by the configuration messages. it is a trade-off between the number of configuration messages and the number of permanent flow entries. The network.applications and architectural design choices," Future Internet[13]. The challenge, therefore, is how to reduce the number of required IPv6 permanent flow entries while keeping a low number of configuration messages in the network because of the limited memory size of the forwarding plane nodes.

SDN abstracts the control functions from the switches or routers as against the traditional network, rendering the data plane devices as a merely forwarding agent [14, 15].A set of security threats, such as unauthorized access and data leakage, are enumerated and mapped against each of the five SDN layers and interfaces.[16]. It is of very low standard and quality, little or no integrity, very easy to forge in, [17].*The tradeoff between the two protocols can provide a significant impact on the networks.in [18]*.It has proven to be the required technology to make networking more scalable, dynamic and removes the data plane devices vendor proprietary [19].

3. METHODOLOGY

The developed IPv6 flow reduction scheme called a IPv6 hybrid permanent flow (IPv6 HPF) scheme, adopts a concept of two MPLS tags, outer and inner tags, and employs both permanent and non-permanent flow entries. The permanent flow entries are permanently installed in every switch before starting a network operation. If a packet does not match any flow entry, the switch requests a configuration message, as a non-permanent flow entry, from the Ryu controller. In this scheme, the flow table is redesigned to support two MPLS tags.The developed Model define the Permanent Flow entries and the Sum of entries in all switches as follows below:.

The Permanent Flow entries = Sum of entries in all switches.----------[1]

The Sum of entries in all switches are defined below:

Sum of entries for local region communication over all switches is given by $\sum_{i\varepsilon s} \sum_{\Sigma s} p_i^j$

Sum entries for inter region $\sum_{i\varepsilon s} \sum_{\kappa\varepsilon\kappa} h_i^k$

Sum of entries for local switch over all switches is given by: $\sum_{i\varepsilon s} \ell_1 + v + u$ $\sum_{i\varepsilon s} ei$ adds a flow entry to each edge switch to pop the outer tag destined within its region. Also V adds a flow entry to each edge switch which are dynamic to pop the inner tag destined within its switch V = |s| and U adds flow entries to each switch to forward the packet to its Hosts u = H x |s|. The overall number of Permanent Flow, entries is given $\sum_{i\varepsilon s} \sum_{j\Sigma s} p_i^j + \sum_{i\varepsilon s} \sum_{\kappa\varepsilon\kappa} h_i^k + \sum_{i\varepsilon s} \ell_1 + v + u$ ----------[2]

Since v and u are constant, therefore they do not affect the optimization result. So, U and V are omitted in the formulation of IPv6 Hybridized Permanent Flow problem.The logic in IPv6 Hybridized Permanent Flow (HPF) problem is to minimize the number of permanent flows entries.The constraints to this formula are; (a) each switch is in one and only one region.(b) switches that share the same region connect to each other through path within the same region and (c) switches which had a link to another region are Edge switches. The constraints to this formula $F_s^k + F_j^k + F_i^k < 2 + \forall k \varepsilon K, s, d i \varepsilon S, r_{nd}^i = 1$ ----------[3]

To define the decision variables use p_i^j and h_i^k. where p_i^j and h_i^k are minimized to Zero by Objective Function in the cases they are not forced to 1.

The components of the network was configured using Python programming language. Wireshark was used to extract data from the network.The implemented IPv6 HPF scheme was modelled via the Mininet simulator. The topology in figure 3 below adequately illustrated the Openvswitch and the connected host functionality link. Each Openvswitch is connected to the Ryu controller and a host is also connected to a corresponding switch. In the topology, the data source IPv6, destination IPv6, inner tag, outer tag and output port of the related scenario was stored in the controller database called MPLS tag table.*Figure 1. shows the Testbed Network Topology* .

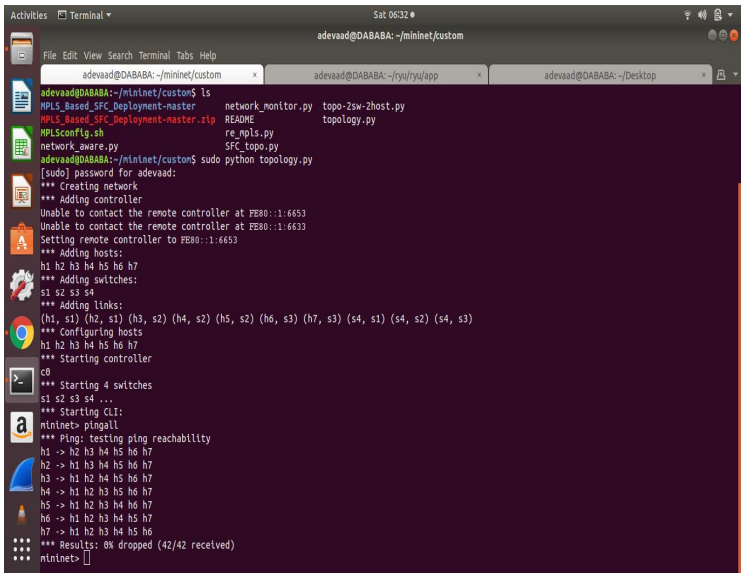


Figure 1: Testbed Network Topology

The topology was created during the experiment which consist of eight switches and eight hosts. The topology was simulated using the command line interface and the result was captured . The controller (Ryu) has an IPv6 loopback address of FE80::1 and it is listening to a port number 6653. A ping6 all request was also sent from various host on the network to each other. The Ryu controller communicates with the switches in the topology using an OpenFlow API. Ryu-manager is the inbuilt mechanism to run all custom build application for the controller. Figure 2 shows the output of the Topology.

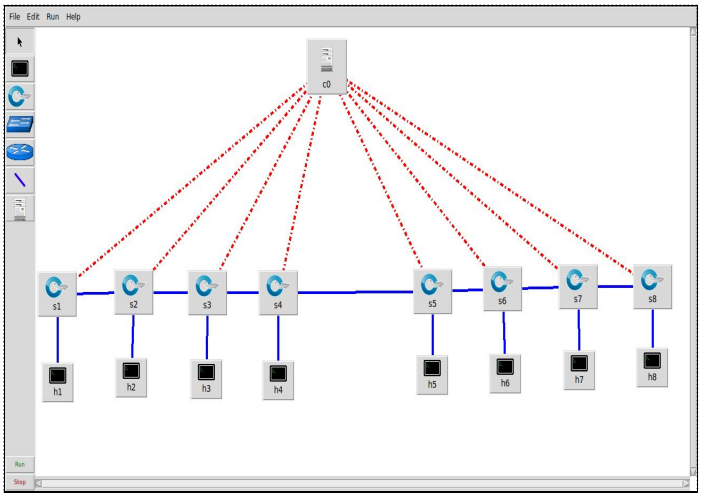


Figure 2: Test bed Topology Flow for the Packet flow

The configuration of MPLS application designed to instruct the controller on how to append MPLS tags on the packet_in requests was depitched in Figure 3.The custom build application is written in python and hence run using a python command.

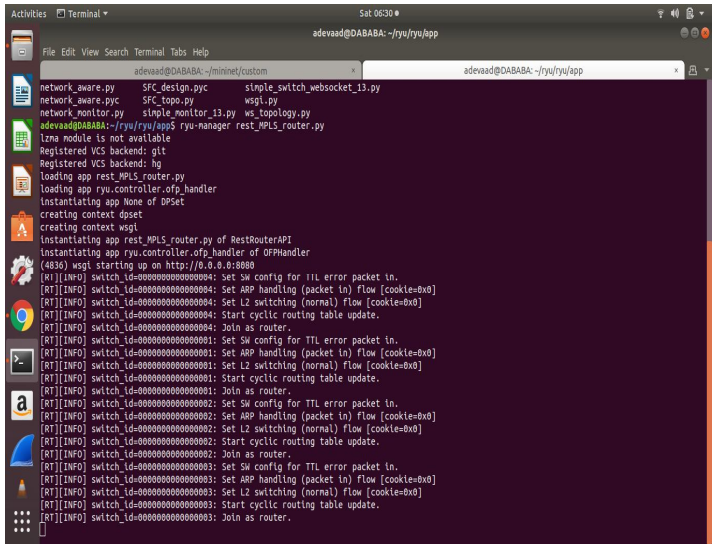


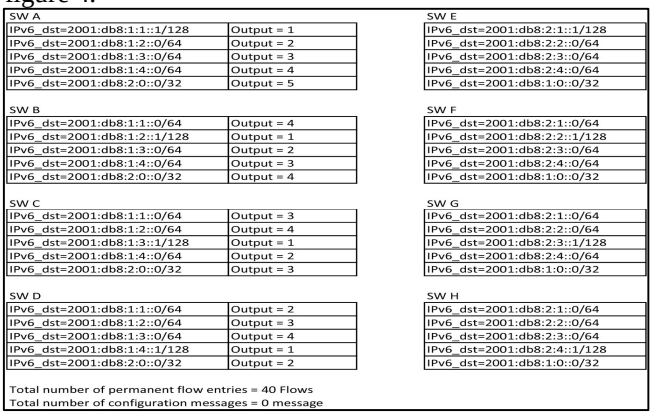
Figure 3. Configuration of MPLS router and switches

Table 1: Characteristics of IPv6 SDN Experimental Test Bed Component

MODEL	CPU/SPEED	RAM	OS	PROTOCOL	EMULATOR
G820 with Processor Intel (R) Core (TM) i5	CPU @ 2.50GHz 2.50GHz,	16.00 GB (15.9GB usable)	Linux Ubuntu 18.04.1 LTS with 32bits	Floodlight 2.2.1 and OpenVswich 2.9.2.	MININET version 2.2

4. RESULT AND DISCUSSION

The total number of permanent flows in the network becomes 40 using ping6 in the network topology as discussed in fig 1. There are three requests, which are (H1 ping6 H2), (H3 ping6 H4), and (H5 ping6 H6) and vice versa. The connection between hosts H1 and H2 is on the path H1, swC, swA, swE, swF, and H2. The connection between hosts H3 and H4 is on the path H3, B, A, E, H4. The connection between hosts H5 and H6 is on the path H5, G, H, H6. No configuration message is required in this case. Flow table in each switch is shown in figure 4.



SW A

IPv6_dst=2001:db8:1:1::1/128	Output = 1
IPv6_dst=2001:db8:1:2::0/64	Output = 2
IPv6_dst=2001:db8:1:3::0/64	Output = 3
IPv6_dst=2001:db8:1:4::0/64	Output = 4
IPv6_dst=2001:db8:2:0::0/32	Output = 5

SW B

IPv6_dst=2001:db8:1:1::0/64	Output = 4
IPv6_dst=2001:db8:1:2::1/128	Output = 1
IPv6_dst=2001:db8:1:3::0/64	Output = 2
IPv6_dst=2001:db8:1:4::0/64	Output = 3
IPv6_dst=2001:db8:2:0::0/32	Output = 4

SW C

IPv6_dst=2001:db8:1:1::0/64	Output = 3
IPv6_dst=2001:db8:1:2::0/64	Output = 4
IPv6_dst=2001:db8:1:3::1/128	Output = 1
IPv6_dst=2001:db8:1:4::0/64	Output = 2
IPv6_dst=2001:db8:2:0::0/32	Output = 3

SW D

IPv6_dst=2001:db8:1:1::0/64	Output = 2
IPv6_dst=2001:db8:1:2::0/64	Output = 3
IPv6_dst=2001:db8:1:3::0/64	Output = 4
IPv6_dst=2001:db8:1:4::1/128	Output = 1
IPv6_dst=2001:db8:2:0::0/32	Output = 2

SW E

IPv6_dst=2001:db8:2:1::1/128
IPv6_dst=2001:db8:2:2::0/64
IPv6_dst=2001:db8:2:3::0/64
IPv6_dst=2001:db8:2:4::0/64
IPv6_dst=2001:db8:1:0::0/32

SW F

IPv6_dst=2001:db8:2:1::0/64
IPv6_dst=2001:db8:2:2::1/128
IPv6_dst=2001:db8:2:3::0/64
IPv6_dst=2001:db8:2:4::0/64
IPv6_dst=2001:db8:1:0::0/32

SW G

IPv6_dst=2001:db8:2:1::0/64
IPv6_dst=2001:db8:2:2::0/64
IPv6_dst=2001:db8:2:3::1/128
IPv6_dst=2001:db8:2:4::0/64
IPv6_dst=2001:db8:1:0::0/32

SW H

IPv6_dst=2001:db8:2:1::0/64
IPv6_dst=2001:db8:2:2::0/64
IPv6_dst=2001:db8:2:3::0/64
IPv6_dst=2001:db8:2:4::1/128
IPv6_dst=2001:db8:1:0::0/32

Total number of permanent flow entries = 40 Flows
Total number of configuration messages = 0 message

Figure 4: Flow entries for HPF scheme

The flow table in each switch in Figure 4.shows the matching destination of IP address. An output port is specified as an action for each flow entry. The number of permanent flows in each switch is five (5).The IPv6 hybrid permanent flow (HPF) scheme, adopts a concept of two MPLS tags, outer and inner tags. The IPv6 HPF scheme divides switches in the network into multiple groups. Each group is called a region. Each region has an individual region ID number, which is specified in the outer tag to indicate the destination region. A switch in the same region has an individual switch ID number, which is specified in the inner tag to indicate the destination switch. Switches are able to have the same switch ID number if they are in the different region. At least one of the switches in each region functions as an inter-region switch to connect the region with other regions. IPv6 Hybrid Permanent flow scheme is presented in figure 5.

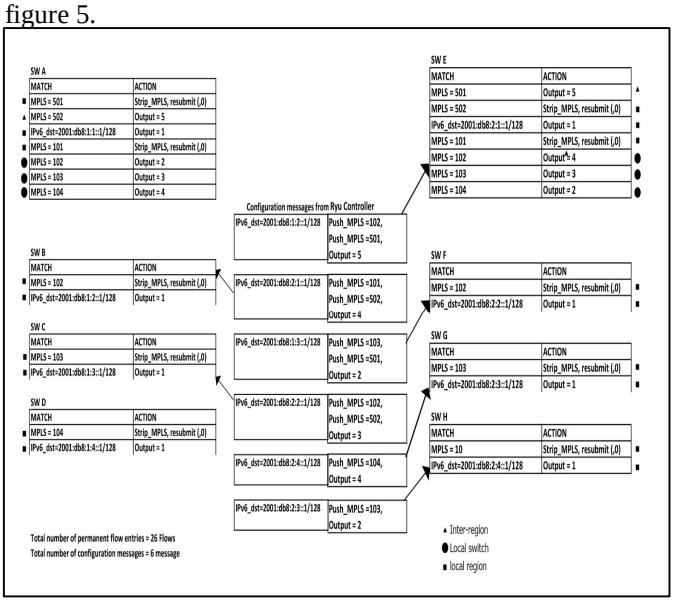


Figure 5:IPv6 Hybrid Permanent Flow Scheme table

The result from the experiment for the dependency of host per switch when Region = 4, Switches = 4 are kept constant, the

IPv6 HPF scheme value was 16 while NPF was 24 as shown in table 2.

Table 2: Analysis of IPv6 HPF– dependency of the number of Host per Switch

NUMBER OF HOST PER SWITCH	TOTAL NUMBER OF FLOWS ENTRIES (103)	PERMANENT FLOW ENTRIES	IPV6 HYBRID PERMANENT FLOW ENTRIES	NON-PERMANENT FLOW ENTRIES
2	0	0		
3	2	0	7	15
4	3	0	7	16
5	4	0	8	16
6	6	0	9	17
7	7	0	10	17
8	8	0	10	18
9	9	0	11	18
10	11	0	11	20
11	12	0	12	20
12	13	0	13	21
13	15	0	14	22
14	16	0	14	22
15	17	0	15	23
16	19	0	16	24

Table 3 considers and compare the HPF scheme using legacy IPv4 and as well as the IPv6 HPF and the figure recorded are percentiles of the total reduction recorded in the schemes. Number of Hosts, switch per region were varied during the experiments to extract the tables and the graphs below. Table 3 shows that there was a 33% reduction in the Non-Permanent flow entries (NPF) stored in the TCAM of the Openvswitch in the network.

Table 3: Comparism of HPF Reduction Scheme

Configuration	LEGACY HPF REDUCTION %	IPv6 HPF REDUCTION %
HOST PER SWITCH	48	33
SWITCH PER REGION	70	63

5. CONCLUSION

The analysis of the number of flow entries from the experiment buttresses the performance of the IPv6 HPF scheme. These reductions require a lower bandwidth in the control plane, a lower process of CPU in the controller, and a reduced TCAM memory size usage in each switch. The study demonstrated the scheme via an experimental IPv6 Mininet testbed. The results of this experimental testbed may be of benefit to next generation of Internet systems network, especially Internet Service Providers (ISPs) for real-time network flows that have unpredictable surge with responses to the transmission of big data across their network backbone.

ACKNOWLEDGEMENT

The authors wish to thank the Department of Computer Science, University of Ibadan for the support in this research work.

REFERENCES

[1]. L. Cui, F. R. Yu, and Q. Yan, "When Big Data Meets Software Defined Networking: SDN For Big Data and Big Data For SDN", IEEE Netw., Vol. 30, No. 1, PP. 58-65, 2016.

[2]. Rijmenam M Van Tesco and Big Data Analytics, a RecipeL61455-1390,U.S.A. 2017.

[3].Cui, L. ,Yu, F. R.,Yan, Q.When Big Data Meets Software Defined Networking: SDN For Big Data and Big Data For SDN", IEEE Netw., Vol. 30, No. 1, PP. 58-65.2016.

[4].Kurimoto,T. A Fully Meshed Backbone Network for Data-Intensive Sciences and SDN Services", in Proc. 8th Int. Conf. Ubiquitous Future Netw. (ICUFN), pp. 909-911.2016.

[5] Olabisi,A. A., Adeniji,, O. D. Abeng Enangha. A Comparative Analysis of Latency, Jitter and Bandwidth of IPv6 Packets Using Flow Labels in Open Flow Switch in Software Defined Network" Afr. J. MIS, Vol.1, Issue 3, pp. 30-36. 2019.

[6]. Adeniji, S.D , Khatun, S, Raja, RSA, Borhan MA 'Design and analysis of resource management support software for multihoming in vehicle of IPv6 Network. Proceedings of the Fifth IASTED International Conference.Vol 607,issue 089.pp 13.2008.

[7]. Logunleko K.B., Adeniji. O.D., Logunleko A.M. A Comparative Study of Symmetric Cryptography Mechanism on DES, AES and EB64 for Information Security. International Journal of Scientific Research in Computer Science and Engineering Vol.8, Issue.1, pp.45-51. 2020.

[8]. Adeniji O.d., Olatunji O.O . *Zero Day Attack Prediction with Parameter Setting Using Bi Direction Recurrent Neural Network in Cyber Security*.International Journal of Computer Science and Information Security (IJCSIS), Vol. 18, No. 3,pp 111-118 .2020.

[9].N. Kitsuwan, S. Ba, E. Oki, T. Kurimoto, and S. Urushidani, "Flows Reduction Scheme Using Two MPLS Tags in Software-Defined Network," IEEE Access, vol. 5, pp. 14 626–14 637, 2017.

[10].N. Kitsuwan, S. McGettrick, F. Slyne, D. B. Payne, and M. Rufni "Independent Transient Plane Design for Protection in OpenFlow Based Networks", IEEE/OSA J. Optic. Commun. Netw., Vol. 7, No. 4, PP. 264-275, 2015.

[11]. Adeniji O.D. & Ukam, J.J. Immune Inspired Concepts Using Neural Network for Intrusion Detection in Cybersecurity" Proceedings of the 20th iSTEAMS Multidisciplinary Trans-Atlantic GoingGlobal Conference KEAN University, New Jersey, USA Pp 119-126. .2019.

[12].Ojoawo A.O & Adeniji O.D. Energy Efficient Hierarchical Cluster Head Election Using Exponential Decay Function Prediction." International Journal of Wireless & Mobile Networks (IJWMN) Vol. 10, No. 5, 2018

[13].Braun,W,Menth,M.Software-defined networking using OpenFlow: Protocols, applications and architectural design choices," Future Internet, vol. 6, no. 2, pp. 302-336, 2014.

[14].Caraguay,, A. L. V. Lopez, L. I. B. Villalba L. J. G. , "Evolution and challenges of software defined networking," in Future Networks and Services (SDN4FNS), pp. 1-7. 2013 .

[15]. Kreutz, D. Ramos, F.Verissimo P.Towards secure and dependable software-defined networks," in Proceedings of the

second ACM SIGCOMM workshop on Hot topics in software defined networking, , pp. 55-60: ACM,, 2013.
[16].Vaughan-NicholsS. J. OpenFlow: The next generation of the network?," Computer, no8, pp. 13-15, 2011 .
[17] Eze Chika Victor, Adeniji Oluwashola David, “Character Proximity For RFID Smart Certificate System: A Revolutionary Security Measure To Curb Forgery Menace”international Journal of Scientific and Technology Research ,pp. 66-70. 2014.
[18].Adeniji O. D, Osofisan Adenike. *Route Optimization in MIPv6 Experimental Test bed for Network Mobility: Tradeoff Analysis and Evaluation.* International Journal of Computer cience and Information Security (IJCSIS), Vol. 18, No. 5,pp 19-28.2020.
[19] Porras, .P. A. S. Cheung, M. W. Fong, K. Skinner, and V. Yegneswaran,.Securing the Software Defined Network Control Layer," in NDSS, 2015.