UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Examining the Performance of Software Defined Virtual Local Area Network

Adamu Abubakar Ibrahim* & Mohamud Abdulkadir Ahmed
Department of Computer Science
International Islamic University Malaysia
Kuala Lumpur, Malaysia
Email: adamu@iium.edu.my

Submitted: 16/6/2022. Revised edition: 11/9/2022. Accepted: 14/9/2022. Published online: 20/11/2022
DOI: https://doi.org/10.11113/ijic.v12n2.373

***Abstract*—For more than three decades, the Virtual Local Area Network (VLAN) has been one of the most popular systems virtualization groups of users on both local and corporate networks. Because of the advantages that VLAN provides, network managers and operators have continued to use it in the creation of their networks and have even extended its use to include cloud computing networking. Previous research studies have established a lot of issues associated with VLAN architecture. Typically, it was revealed that it is difficult to set up a complex VLAN, and even it is successful, it leads to computational time-consuming, and prone to error to process majority of complex VLAN. Considering this, the current research examines the performance of "Software Defined-VLAN". This is because Software-Defined Network (SDN) is a viable alternative network architecture that allows the separation of information and control functions on devices. It appears to be a promising optimized option VLAN administration. The SDN-enabled VLAN testbed was set out and implemented using OpenFlow. SDN-Enabled VLAN and a conventional VLAN. The finding revealed that SDN-enabled VLAN offers higher network performance, lower packet transfer delay, and a more efficient configuration.**

***Keywords*—Software-defined network, virtual local area network, open flow, latency**

I. INTRODUCTION

Managing virtual local area networks (VLANs) is often ranked as one of the most challenging tasks for network administrators. Configuring Virtual Local Area Networks (VLANs) is still a laborious, complex, and error-prone procedure. This is because network managers have to utilise the device's command line interface (CLI) to manually configure layer two switches. These tools need to be located in the same region in order for network managers to be able to grow the entire network (e.g., on the same floor). Even though Cisco's VLAN Trunking Protocol (VTP) is now supported, the protocol's capabilities are still very restricted in terms of both its scope and its performance [1-2]. The establishment of VTP domains is a precondition for the successful handoff of administrative responsibilities related to VTP. In addition, Cisco switches such as the Catalyst Family do not support the VTP protocol. Because it enables network managers to simply setup VLANs and gives a virtual view for monitoring and troubleshooting each VLAN on the network, a VLAN management tool or application is required, in our opinion, in order to address these difficulties. [3] SDN and OpenFlow are the greatest choices for fixing the issues with traditional VLAN management and performance, respectively. Both of these problems have been caused by traditional VLANs. This section gave a summary of the research as well as some background information on the topic.

The Open Networking Foundation (ONF), which was established in 2011 with the purpose of promoting a new networking paradigm known as Software-Defined Networking, is now striving to decouple the control plane from the data plane [4]. On the other hand, the idea of an SDN is still relatively new, and its acceptability is limited to the theoretical level at this point. SDN is described by the ONF as the separation of the control plane from the forwarding plane, with each control plane being responsible for many devices. The SDN Controller, which acts as the "brain" of the network, is tasked with the responsibility of determining how network traffic should be managed in order to achieve optimal performance. Typical depictions of the Software Defined Network architecture include Layer 1, Layer 2, and Layer 3.

The infrastructure layer, also known as the data plane, is where all of the components of the network that are responsible for forwarding data are situated. The control layer, which is often referred to as the control plane, is responsible for the programming and management of the forwarding plane. When utilising network applications and services, new network features can be implemented at the application layer. This layer is located in the network stack. Implementation of SDN might be viable or it might not be achievable depending on the network strategy that ventures want to use. SDN refers to a collection of protocols and technologies that cooperate to produce an integrated picture of networks and to supply centralised, intelligence-based network service provisioning and control [5]. These protocols and technologies can be broken down into several subcategories. In recent years, investigators have concentrated more on the theoretical features of SDN than they have on its practical applications. As a consequence of this, this research implement SDN by utilising the SDN paradigm within the VLAN idea.

The extensive implementation of networking across a variety of platforms has resulted in its transformation into an indispensable component of human existence. Businesses and individuals alike can benefit from exchanging knowledge and resources through the usage of networks. Cloud computing and large systems such as business networks need for a flexible technique of configuring and forwarding packages between network nodes. This requirement is in line with the changes that are now taking place. The need to do route lookups in the forwarding policy firmware causes these legacy networks to have a greater delay [6]. The efficiency of the network will suffer if the amount of time it takes for packets to go from a node that is sending out data to a node that is receiving data is lengthened. For the purpose of researching SDN-enabled VLANs, this dissertation makes use of OpenFlow. The next thing that will be looked at is an SDN-enabled VLAN together with an Access VLAN. Monitoring the passage of packets is something that this research do for switches that do and do not have SDN capabilities.

Along with the current section, this paper is divided into five sections. The remaining four sections of the study build upon the information presented in this section. In Section 2, the research present and discuss the relevant related work for the study. The research methodology is presented in Section 3, followed by the experimental analysis and presentation of the result in Section 4. Finally, section 5 present the conclusion of the work.

II. RELATED WORK

SDN strategies for VLAN is entirely an area where controller part of the framework plays an important role. Since its successful introduction, using software-defined networking in data centres and enterprise networks is becoming increasingly commonplace (SDN) is now being implemented more widely [7]. However, the SDN is dealing with these new security issues, particularly with regard to protecting the SDN architecture itself [8]. There are many previous research studies associated with SDN and VLAN. Crucial to this is the work of Lehocine and Batouche [3], which highlight the benefits and adaptability of deploying network applications to SDN centers, created for dealing with an issue or configuring habits, in contrast with managing network automatically, services on standard network infrastructure, created for the very same objective. To illuminate their research, they provide a research study of the" VLAN filtering system along with division" autonomic deployment in addition to both atmospheres; traditional network along with SDN network, proving that SDN basic provides the managers with a renovation in control in addition to administrative jobs, with an extensive translation of the business requires right into industrial networks.

It is essential to investigate software-defined virtual local area networks (SD-LAN), as they combine two distinct domains, each of which has been the subject of a significant number of earlier research studies that have highlighted the significance of the area. Mendiola *et al.* [9] examine a list of SDN protocols to find a solution for traffic engineering. They also evaluated the impact of SDN protocols on traffic engineering by comparing the protocols with the Path Computation Element (PCE) based architecture and providing traffic engineering solutions through Software-defined networking. In a separate piece of research, Huang *et al.* [10] identified potential research barriers for large-scale SDN testbeds. It demonstrates the many different SDN Testbeds. Mishra and AlShehri [11] explain about software-defined networking (SDN) and how to pick the best SDN controller. They also discuss about the challenges, opportunities, and research issues that are related to SDN.

Medlin [12] presented an extensive review of software-defined networking and the growth of SDN standardisation in the future of the general software-defined networking Concept. Keti and Askar [13] evaluated an SDN emulation tool called Mininet and studied Mininnet's limitations related to the simulation environment and resource capabilities. It has heightened how the Mininet emulator in different simulation environment works. In another transformation, Nunes and colleagues [5] investigated contemporary programmable networks with a particular focus on SDN. In addition to this, they offer a historical perspective on programmable networks. In addition, Huet *et al.* [14] provide an in-depth review of the most important implementation areas for SDN (OpenFlow). In addition to applications, language abstraction, and controller virtualization, as well as quality of service (QoS), and security, these important subjects contain the fundamental idea of software-defined networking (SDN). As a result, the research offers an implementation of SDN OpenFlow on both wireless and optical networks.

There are other numerous studies which they also investigated the issues associated with SND, among them is the work of Farkas *et al.* [15], presented a blueprint of how SDN concepts and SDN principles could potentially be implemented in Ethernet merchant silicon that already exists in light of the new network requirements. Futhermore, Kim and Feamster [16] identify issues with current network configuration and management mechanisms. The study identifies problems with existing network configuration and management mechanisms. The study also provides a way to improve multiple aspects of network managementMousa *et al*. [4] also discussed the concepts and applications of SDN, with a focus on the unresolved research challenges posed by the new technology. Lehocine and Batouche [3] highlighted the advantages and adaptability of deploying network applications on top of SDN facilities, which were created to solve a problem or configure a behaviour, as opposed to Autonomic Network Management (ANM) services on traditional network infrastructure VLAN filtering and segmentation. Lehocine and Batouche [3] also highlighted the benefits of deploying network applications on top of SDN facilities, which were designed for configuring a behaviour. Bojovic *et al*. [17] conduct research on implementation issues and their solutions, as well as the complexity of hybrid architecture from the perspective of network management.

III. RESEARCH METHODOLOGY

The research methodology within the context of this current study refers to the act of selecting, evaluating, and providing justification for the procedures that are utilized for investigating the SDN-VLAN [18]. In this section, the research examines the experimental methodology as well as the theoretical foundation for experiment to determine the impact of SDN-VLAN. In addition to that, it details the many instruments and tools that were utilized for the purpose of experiment as well as the approach that was utilised for constructing SDN-enabled VLANs in addition to traditional networks. There are numerous options of creating SDN-enabled VLAN lab, some simpler than others, depending on requirements or skill set. However, at the heart of an SDN testbed is virtualization where the research virtualizes almost everything: hosts, network links, switches, routers, controllers. The computational operation is crucial to the performance of this kind of operation [19]. On the other hand, evaluating traditional network the research use GNS3 simulation with CISCO IOSL2 switches. In this work, the research wants to emphasize that the lab is using is just one way of testing! As the research mention, there are many other ways to achieve the same things - for instance, by using LXC, QEMU, KVM, and many more.

A. Experimental Testbed Development

This section presents the detailed of how the experimental testbed are established involving the installation of a RYU controller and Mininet for testing purposes, as well as how to set up virtual machines inside of a virtual environment, configure virtual networks between the virtual machines, and set up virtual networks between the virtual machines themselves. The purpose of setting out the environment is to walk the reader through the process of installing and configuring the controller, as well as the typical steps involved in setting up a network, by making use of GNS3 along with CISCO IOSLv2. The necessary steps for creating the SDN environments, such as installing MININET, RYU Controller, Open-V-switch, and Wireshark, are crucial to this work. The experimental evaluation follows after setting up. This experiment was carried out in a virtual environment with essential software. The research utilises an operating system based on a Linux distribution in this testbed. All commands have been validated on Ubuntu server 16.4 LT; however, the results may vary when applied to other versions or distributions. The research was able to construct a full SDN development stack by combining the components that were discussed above. The workflow steps involved in setting up an SDN development testbed is presented in Fig. 1.

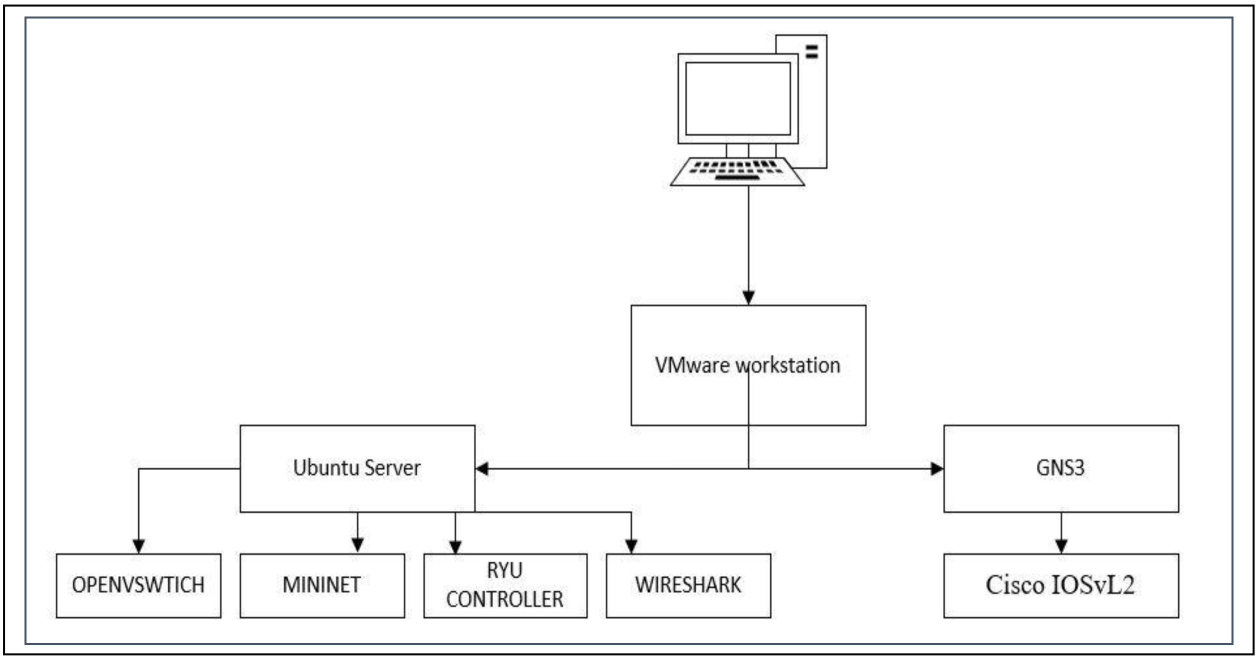

Fig. 1. The SDN development testbed

The SDN/VLAN testbed was established by VMware workstation pro 14, Ubuntu Server 16.4LTS, Mininet 2.2.2, Openvswitch 2.5.5, Wireshark 2.6, Graphical Network Simulator (GNS3) 2.1, and Cisco IOSvL2 15.2.

Software-defined networking (SDN) framework Ryu is a component-based framework that provides software mechanisms with an API. It makes it simple for programmers to create new network control and administration apps. Because Ryu can handle a wide variety of network device protocols, including OpenFlow, it was chosen for this investigation. The research will begin installing the RYU controller once the VMware is up and running. The OpenFlow protocol is then used to connect the Control Layer to the Infrastructure Layer. As a virtual network interface emulator, Mininet allows us to create and manipulate Software Defined Networking components on our laptop or PC. SDN networks can be implemented using Mininet, a visual simulation software. Multilayer software switch Open vSwitch (OVS) is released under the Apache 2 open source license. The developers' goal is to create a switch platform of production grade that supports standard administration interfaces and allows programmatic extension and control of forwarding capabilities.

Wireshark is a network traffic analyser. It captures every packet getting in or out of a network interface and shows them in a nicely formatted text. This research was able to run an IOS switching image on the VMware machine utilizing Cisco's Virtual IOS L2 software. The research is able to configure and test a wide variety of advanced layer two switching features in a Cisco VIRL IOSvL2 image. These features include spanning tree protocols, VLANs, and many others. Using the GNS3 add-ons, it is the quickest and most convenient method for importing the VIRL setup images into GNS3.

The research has carry out the experiment using a linear topology, as depicted in Fig. 2. This topology will consist of three switches, each of which will have four hosts connected to it. Despite the fact that the research has two VLAN IDs (100 and 200), hosts on the same VLAN ID should be able to communicate with one another and exchange network traffic with one another.

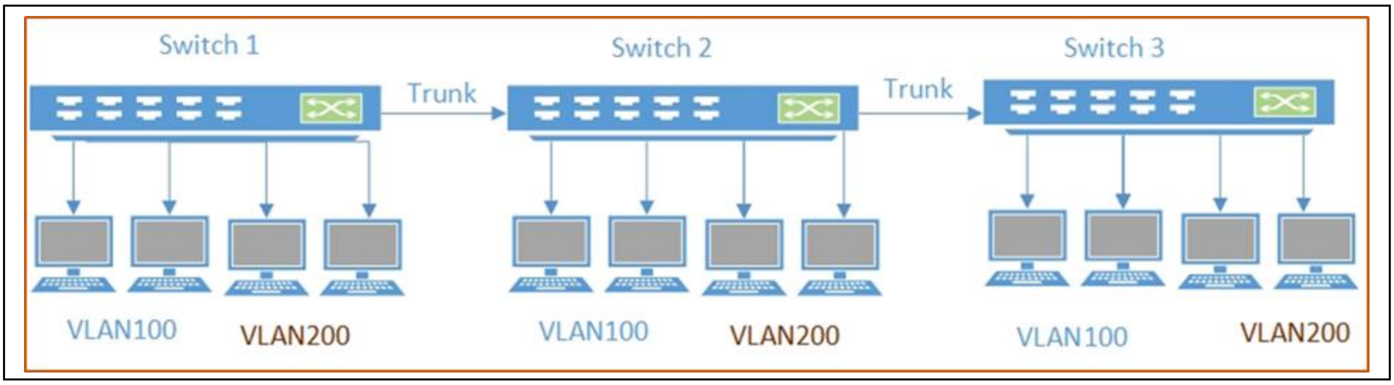

Fig. 2. The conceptual framework

B. Experimental Simulation and Presentation of the Results

After the testbed development, the Ryu controller is prepared to offer its services to the various network devices, and the research anticipate that the host in VLAN100 will be able to communicate with all of the other hosts that have the same VID, in addition to the host in VLAN200. In order to measure the performance across the network, the research carry out a number of tests, such as dump, net, and link, as well as a reachability test on the Mininet in order to determine the best scenarios. In order to evaluate how well the SDN and VLAN work together, several test scenarios were carried out.

IV. EXPERIMENTAL RESULTS

To this point, the research has successfully configured mininet, RYU controller, and openvswitch. In order to carry out the experiment, let's now check the TCP connection that exists between the switch and the controller. As shown in Fig. 3, the research begins by starting the RYU controller, which then opens the TCP listening port 6633. Next, the research run the mininet topology, at which point the switch begins a TCP session with the controller. In order to verify, the research make use of Wireshark to capture the packets (by display-filtering on TCP. port==6633), which enables us to view the TCP-handshake. The Fig. 3 below shows a Wireshark capture packet, which contains three messages that make up the handshake and SYN-SYN-ACK.

Once both the controller and the OVS have been connected, a series of messages will be communicated between the two devices. As an illustration, the OVS will communicate with the controller using an OFPT HELLO message. The first HELLO packet is displayed on the Wireshark display after it was successfully captured. The variation number is contained in the very first byte of the message. The OFPT TYPE is stored in the second byte. The type 0 message is the OFTP HELLO message. Following the transmission of the HELLO message by the switch, the controller will then send an OFPT FEATURES REQUEST (type=5) in order to recover the capabilities of the switch. These abilities may include the sustained open flow version, switch arrangement, port equipment address, and so on. The switch communicates the function details to the source via an OFPT FEATURES REPLY (type=6) transmission. Both the controller console and the switch console display the message when it is brought to the player's attention.

There are a number of commercial and open source SDN controllers that can be downloaded from the internet. The comparison of these two products is not within the purview of this dissertation's investigation. The characteristic of the SDN and the traditional network is compared and contrasted in this section. The test known as latency is the one that enables one to make this comparison. The implementation of the software-defined network (SDN) concept using the OpenFlow protocol is the primary objective of this thesis. In order to achieve this objective, it is necessary to carry out a number of experiments, each of which must include Wireshark packet captures. After the research have gathered the most important performance parameters from both the conventional network and the SDN, the research models the data so that it can be represented graphically. The measure of latency is one of the most important parameters that the compared (see Fig. 3). Therefore, throughout the entire transmission session in which 16 packets were captured, it was discovered that SDN-based latency is significantly reduced. This indicates that back and forward communications in the SDN-based VLAN are quite a bit faster than they are in the regular non-SDN VLAN.

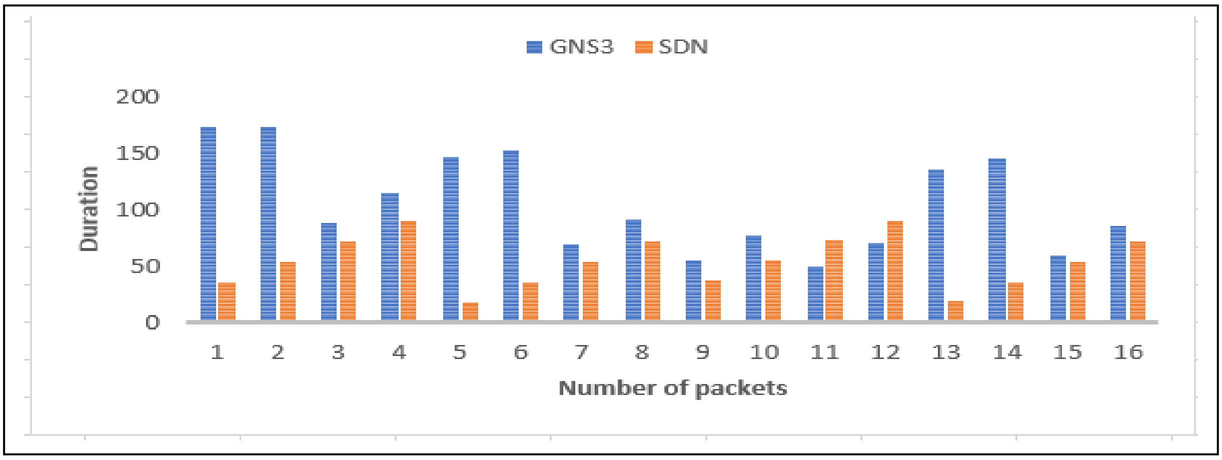

Fig 3. The dynamic component of transmission latency comparison

In order to accomplish this result, the research first conduct a ping test, and then the research use Wireshark to capture and analyse packets while measuring latency. The conventional network model, which makes use of Cisco switches and is represented by the blue bar on the graph, contrasts with the software-defined networking (SDN) network, which is represented by the orange bar. The research use the ping command in both the software-defined network (SDN) and the traditional network. Fig. 3 demonstrates that the latency of the packets in software-defined networks (SDN) is lower than the latency of the packets in conventional networks.

The SDN framework used in this analysis revealed that the networks in question being programmable both at the time of deployment and at a later stage, depending on the changes that have been made to the requirements, it tends to affect every part. The SDN framework contribute to the success of newly established businesses by being flexible, agile, and virtual control. The SDN framework configuration is based on the use of open software; OpenFlow, which is the industry accepted implementation and standard. This has a control plane which was utilized by this study to logically centralized the VLANs, in order to manage traffic in a centralized fashion. As compared to the traditional networks, SDN performs better, while the VLANs on non-SDN are rigid and unable to adapt to changing conditions. They are not helpful to new business endeavours in any way. They lack both agility and flexibility in their movement. Their hardware appliances are specifically dedicated.

The implication of this study lies with the fact that this paper, integrate the SDN and OpenFlow concepts into the Virtual Local Area Network concept. The most important thing that the research contributed was the design and implementation of a system that allows a VLAN network to be configured in either a software-defined network or a traditional network. According to the findings, the research conducted the evaluation by configuring an SDN-enabled VLAN by employing the RYU controller as a control plane and mininet as a data plane within a virtualized environment with some advanced features installed. The paper is structured in three distinct sections. First, the research implemented a software-defined network-enabled virtual local area network (SDN-enabled VLAN) in addition to a conventional network. This allowed us to introduce the software-defined network paradigm. In the second step of our research, the research analyses the performance of SDN-enabled VLANs in addition to traditional networks. Finally, the research investigates the behaviour of packets while they are being transmitted through Cisco switches that are enabled with SDN.

The future work of this research dwells on scope and results. While this study focuses on VLANs and SDNs in separate environments, future research may merge the SDN-based VLANs to have a controller to manage them all, which could be within SDN Multi-Domain Supervisory Controller

[19] another VLAN that is managed by the same controller. This study focuses on VLANs and SDNs in separate environments. In addition, considering that the finding of this research is associated to the transmission session latency within which SDN-based latency is lower, the result can also be tested further in an identical scenario but with different parameters. This is because the finding is related to the fact that the SDN-based latency is lower.

V. CONCLUSION

In conclusion, according to our findings, SDN are a valuable asset to the new networking paradigm. In addition to being easily configured, a significant reduction in configuration time was observed during the implementation, as well as a reduction in the number of miss configurations. In addition, miss configurations were reduced (error IP address and Creation of VLAN). In contrast to traditional networks, which require users to manually configure their devices, wireless networks do not require this step. In addition to these benefits, the findings were surprising in that they demonstrated that, in terms of performance, SDNs are superior to traditional networks. This was demonstrated by the fact that, despite the fact that the initial packet always has high latency values due to the fact that it must be processed by the controller, the remaining packets have lower latency when compared to traditional networks.

ACKNOWLEGEMENT

This research is made possible and supported by UMP-IIUM Sustainable Research Collaboration 2022 Research Grant

REFERENCES

[1] G. Zhai, Z. Long, J. Zhong, & Y. Cui. (2012). Design and Research of VLAN Communication Experiment Based on the Web Environment. *Procedia Engineering*, 29, 2485-2490.

[2] O. Obono., & B. P. Tawo. (2022). Improving Security in A Virtual Local Area Network. *Journal of Theoretical and Applied Information Technology*, 100(10).

[3] M. B. Lehocine, & M. Batouche. (2017). Flexibility of Managing VLAN Filtering and Segmentation in SDN Networks. *2017 International Symposium on Networks, Computers and Communications, ISNCC 201*.

[4] M. Mousa, A. M. Bahaa-ElDin, & M. Sobh. Software Defined Networking Concepts and Challenges. *Proceedings of 2016 11th International Conference on Computer Engineering and Systems, ICCES 2016*, 79-90.

[5] B. A. A. Nunes, M. Mendonca, M., Nguyen, X. N., Obraczka, K., & T. Turletti. (2014). A Survey of Software-defined Networking: Past, Present, and Future of Programmable Networks. *IEEE Communications Surveys and Tutorials*, 16(3).

[6] R. Kumar, M. Hasan, S. Padhy, K. Evchenko, L. Piramanayagam, S. Mohan, & R. B. Bobba, (2018). End-to-End Network Delay Guarantees for Real-Time Systems Using SDN. *Proceedings - Real-Time Systems Symposium, 2018*.

[7] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, & S. Mounir. (2022). A Comprehensive Survey on SDN Security: Threats, Mitigations, and Future Directions. *Journal of Reliable Intelligent Environments,* 1-39.

[8] H. S. Alotaibi, M. A. Gregory, & S. Li. (2022). Multidomain SDN-Based Gateways and Border Gateway *Protocol. Journal of Computer Networks and Communications*.

[9] A. Mendiola, J. Astorga, J. Ortiz, J. Vuleta-Radoičić, J., Juszczyk, A., Stamos, K., ... & M. Higuero. (2017). Towards an SDN-based Bandwidth on Demand Service for the European Research Community. *2017 International Conference on Networked Systems (NetSys)*, 1-6.

[10] Huang, A., Nikaein, N., Stenbock, T., Ksentini, A., & Bonnet, C. (2017, May). Low latency MEC framework for SDN-based LTE/LTE-A networks. *2017 IEEE International Conference on Communications (ICC), IEEE*, 1-6.

[11] S. Mishra, & M. A. R. AlShehri. (2017). Software Defined Networking: Research Issues, Challenges and Opportunities. *Indian Journal of Science and Technology,* 10(29), 1-9.

[12] Y. R. Medlin. (2016). An Extensive Review on Software Defined Networking (SDN) Technologies. *DJ Journal of Advances in Electronics and Communication Engineering*, 2(1), 1-9.

[13] F. Keti, & S. Askar. (2015). Emulation of Software Defined Networks Using Mininet in Different Simulation Environments. *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, 205-210.

[14] F. Hu, Q Hao, & K. Bao. (2014). A Survey on Software-defined Network and Openflow: From Concept to Implementation. *IEEE Communications Surveys & Tutorials,* 16(4), 2181-2206.

[15] J. Farkas, S. Haddock, & P. Saltsidis. (2014). Software Defined Networking Supported by IEEE 802.1 q. arXiv preprint arXiv:1405.6953.

[16] H. Kim & N. Feamster. (2013). Improving Network Management with Software Defined Networking. *IEEE Communications Magazine*, 51(2), 114-119.

[17] Z. Bojović, P. Bojović, & J. Šuh. (2018). Implementing Software Defined Networking. *Journal of the Institute of Telecommunications Professionals.*

[18] J. Robertson. (2000). The Three Rs of Action Research Methodology: Reciprocity, Reflexivity and Reflection-on-reality. *Educational Action Research,* 8(2), 307-326.

[19] M. N. Ali, M. H. Rahman, A. A. Ibrahim, M. Al-Naeem, and E. Hossain. (2022). A High Static Performance Hierarchical Three-Dimensional Shifted Completely Connected Network. *IEEE Access,* 10, 43812-43836.

[20] A. A. Ibrahim, A. R. Atayee, and I. A. Lawal. (2022). SDN Multi-Domain Supervisory Controller with Enhanced Computational Security Count. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* 14(2), 23-29.