DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

Design an Algorithm for dynamic Slice of SD-WAN Using VLAN

Talah Oday Alani[1], Ameer Mosa Al-Sadi[2]
[1,2]Computer Engineering Department, University of Technology, Baghdad, Iraq
[1]ce.19.03@grad.uotechnology.edu.iq, [2]Ameer.M.AlSadi@uotechnology.edu.iq

Abstract— *Software-Defined Network (SDN) is one of the most predominant technologies for networking in the existing and next-generation networks. SDN can conFig. , control, protect, and optimize network resources through software. The fundamental benefit of SDN is enabling the application of dynamic management. In addition, the literature shows that partitioning a Software-Defined Wide Area Network (SD-WAN) into several logical networks efficiently will optimize its performance. The main aim of paper is to design an algorithm to slice SD-WAN dynamically into several virtual networks according to the server-clients' correlation using the Virtual Local Area Network (VLAN). The several virtual networks improve QoS of SD-WAN and reduce its broadcasting domain. The proposed framework consists of two parts. The first part is the management algorithm that finds the best server for each client; then it groups this server with their client in a dedicated logical network. The second part includes creating a VLAN for each logical network in an SD-WAN. The application of the POX controller calculates and maintains the dynamic VLAN, which will be applied by the control plan to slice the topology in the data plan. SD-WAN topology is tested before and after applying VLANs. The results show enhancement in latency by 42.85%, throughput by 4.61%, loss packet by 72% and jitter by 47.86% after applying VLAN. Finally, the greatest gain is reducing the broadcasting ratio by 77.77%.*

Index Terms— *Virtual Local Area Network (VLAN), Software-Defined Network (SDN), Software-Defined Wide Area Network (SD-WAN), Quality of Service (QoS), OpenFlow (OF).*

I. INTRODUCTION

Today's society relies largely on technology to carry out daily tasks. Within a firm, critical information is always shared fast. Fast and secure transferring for information was critical issue for traditional network due to missing an end-to-end path reservation mechanism. Fortunately, technology development has improved a way for granted transmission for information. Traffic engineering utilizes some tunneling techniques called Virtual Local Area Network (VLAN) to act as mechanism to reserve an end-to-end path [1], [2]. VLAN was a good solution for the path reservation for previous era.

Today networks recall the need to be developed due to the introducing of the new services like the Internet of Things (IoT), cloud integration and cloud services, Big Data, and Information Technology (IT) consumerization and mobility. This escalation restricts the use of VLAN approaches to address all of these fresh problems. It refers to the requirement for managing a new solution that supports the programming language represented by a new model called Software-Defined Networking (SDN) and can offer more management freedom for VLAN[3]–[5].

Received 22/September/2022; Accepted 29/October/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

VLAN is a tunneling technology separated from the physical core network. It could be conFig. d as logical networks on layer 2 (data link layer). This technique allows making a reservation for some network resources for specific traffic. VLAN has already achieved a particular level of optimization and contributed to solving several challenges, such as slicing the shared network into several smaller networks and saving the privacy of specific services. VLAN applications can also increase security, control the network, and simplify network management [6], [7]. VLANs work by adding tags to network frames and managing these tags in networking systems to give the appearance and functionality of network traffic that is physically on a single network but behaves as though it is split between several networks. Despite being connected to the same physical network, VLANs can maintain network application separation without deploying additional cabling and networking hardware [8], [9].

SDN is a type of networking architecture, which allows software applications to design the behavior of the whole network and its devices from a central location utilizing open APIs. In other meaning, SDN is an emerging networking architecture that is dynamic, controllable, cost-effective, and adaptive, making it suitable for high-bandwidth dynamic applications [10]–[12]. The network control and forwarding operations are decoupled in this design, allowing network control to be directly programmable and the underlying infrastructure to be abstracted for applications and network services [4], [13]. The northbound and southbound SDN architectural APIs define the connection between applications, controllers, and networking systems [14]. A northbound interface connects the controller to applications, whereas a southbound interface connects the controller to actual networking gear. These parts do not have to be physically situated in the same area because SDN is a virtual network overlay [15], [16].

The new controller rules and OF switches capabilities increase the desire to employ VLAN with SDN, to explore and examine the new abilities of dynamic management for VLAN, which SDN provides. That explains the desire to utilize VLAN with SDN and view it as a topic of this study. Besides, the lack of research contributes to optimize the QoS and broadcast domain in SDN using reserved bandwidth and optimal domains. This motivates the researcher to investigate this niche and work to design a model to optimize the features mentioned above.

Our paper contributes by optimize QoS and minimize the broadcasting domain of SD-WAN using an algorithm to maintain the dynamic VLAN. The contributions of this paper illustrated below:

- Design algorithm to compute and determine network slices (logical networks) for every server and its clients in SD-WAN.
- Create multiple logical networks using VLAN to optimize QoS by reserving bandwidth and optimal domain.
- Improving the average delay of every VLAN by maintaining a dynamic VLAN segment for a group of hosts with its required server in a dynamic manner.
- Optimize the broadcast domain by slicing the network into several logical networks (VLAN).

SD-WAN and its VLAN slices will be applied using the MININET SDN emulator, POX controller and MobaXterminal remote server. As mentioned in the previous step, the POX controller has been used, which is managed by a software-defined networking application that will be designed and programmed in MATLAB.

The rest of this paper is arranged as the following: section II will briefly revise the related work; section III present the methodology of the proposed work; section IV will

Received 22/September/2022; Accepted 29/October/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

discuss the implementation and results of proposed work. Finally, this paper is concluded in section V.

II. RELATED WORK

Several studies have used SDN architecture to enhance virtualization management at various levels. In 2016 [17], the researchers used VLANs to develop SDN by incorporating SDN and OpenFlow into the campus and enterprise network environments. This architecture built and implemented an application for quickly controlling and flexibly troubleshooting the VLANs. This program provides both static VLAN and dynamic VLAN settings. This study answers the hybrid mode service, which requires packet processing from both the OpenFlow control and traditional control planes.

In 2017 [18], M. B. Lehocine and M. Batouche aim to offer efficient VLAN segmentation for big networks, with minimum input data from the network administrator and low overhead processing for the switches. In the same year, O. M. Poncea et al. [19] achieved multi-VLAN management by introducing a new source routing approach to SDN that uses stacked VLAN tags, and it proved to have many benefits over other forwarding approaches. In Mininet, this approach was tested using the Ryu controller. This solution can be allowed by default across the data center, simplifying flow tables in core switches, or only when the number of flows exceeds the total capacity if the packet size is a problem. When to do it is up to the controller. This technique was used to improve multipath routing because packets can be led on different paths from the source using fine-grained distribution algorithms, and switches in the center would be unaware of it.

In 2018 [20], default, native and management VLANs were viewed as three distinct VLAN types. The effects of the packet tracer software confirm that VLAN-based network isolation isolates the transmitted traffic of the devices regardless of their existence in the exact location with separate network classifications, even if all of the systems in the regions will share the same cable and switch. Create VLANs for data and traffic management as the LAN expands, and more network devices are introduced to keep the broadcast frequency valid. That same year, J. Chen et al. [21], propose a pro-VLAN solution that determines an exact backup path and assigns a single VLAN id to each network connection that the security system supports. It takes advantage of the most critical features of SDN and can recover one connection loss in SDN, all while offering high performance, scalability, and applicability. According to the simulation results, both pro VLAN and pro-PATH achieve a fast failure recovery.

In 2019 [22], A. Hameed and M. Wasim achieve four VLAN functions by using SDN and VLAN networks. These functions include broadcast inclusion, QoS, security policy enforcement, and traffic grasp. In this study, experimental networks are used for both testing SDN-capable switches and VLANs.

In 2021 [23], M. B. Lehocine and M. Batouche aim to deliver efficient VLAN segmentation for big networks with the least amount of switch processing and input data required from the network administrator. The RYU SDN controller was selected to validate the suggested technique for deploying VLAN filtering and segmentation on SDN networks. The SDN network is emulated using Mininet to build an entire OpenFlow network.

III. THE PROPOSED METHOD/ALGORITHM

This section will explain the method to achieve the main goal of this study as steps-in detail, then it will describe and demonstrate the designed model to apply study target. The

Received 22/September/2022; Accepted 29/October/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

main goal of this study is to design an algorithm to slice the SD-WAN into multiple logical networks using dynamic VLAN. Applying this algorithm in SD-WAN will improve its QoS and reduce its broadcasting domain. The improvements of QoS include minimizing the latency, jitter, and packet loss and optimizing the network's throughput.

The suggested system consists of two parts. The first part of the framework is designing the management algorithm that finds the optimal server for every client. The optimal server is defined as the server with the path of optimal weight to the client than other servers. After that, the algorithm will collect every server with its clients in the separated logical network (VLAN). The second part of the framework is creating a VLAN for every logical network in SD-WAN, according to the decision of the first part.

Using VLAN in SD-WAN will improve the controller performance by protecting the controller from overhead when getting a high number of new rules for many flows and also reduce the computational time after applying VLAN. This work will be achieved using the POX controller, which will be controlled by an SDN application built and written in the MATLAB and python programming languages.

A. First part of Framework: Design Management Algorithm

The first part uses the algorithm to make the dynamic clustering for SD-WAN. The algorithm starts by finding the optimal path between every client and every server can provide the required serves. Then, select the server with the optimal weight path as the optimal server for this client. The optimal path can be calculated according to minimum distance or maximum throughput.

The Dijkstra algorithm is employed in part one to find the optimal path. The work of this algorithm depends on finding the path of the optimal weight between two nodes in a graph. The algorithm procedure will be explained in detail below to show the working steps from the beginning to the final decision. The following algorithm demonstration has two options that will be clarified during the explanation.

Step 1: The algorithm starts by inputting the number of nodes and link value (distance and bandwidth).

Step 2: Input the number of servers and locations (nodes of higher centrality).

Step 3: Input the number of clients and their locations (randomly) in topology.

Step 4: Apply the optimal path from every client to every server.

Dijkstra algorithm is applied to calculate the best path with the optimal weight of a specific option. Note that one of two different options must be selected to identify the weight criteria of the optimal path:

a) Weight = minimum distance.

b) Weight = maximum throughput.

Each client will search and choose the server of the optimal path and request its services.

Step 5: Draw the graph with highlighting the optimal path between client and server.

Step 6: Group every server with its clients and send the optimal paths between every server and its client to the controller to program them as a single VLAN (create abstraction to be sent to the controller).

Decide the optimal server for every client (select server has the best path to the client). These steps are illustrated in *Fig. (1-A)*. At the end of the first part, it is noteworthy to mention that the two weight options can lead to different slicing types. Select every option to serve different implementations as illustrated in the following two paragraphs:

Received 22/September/2022; Accepted 29/October/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

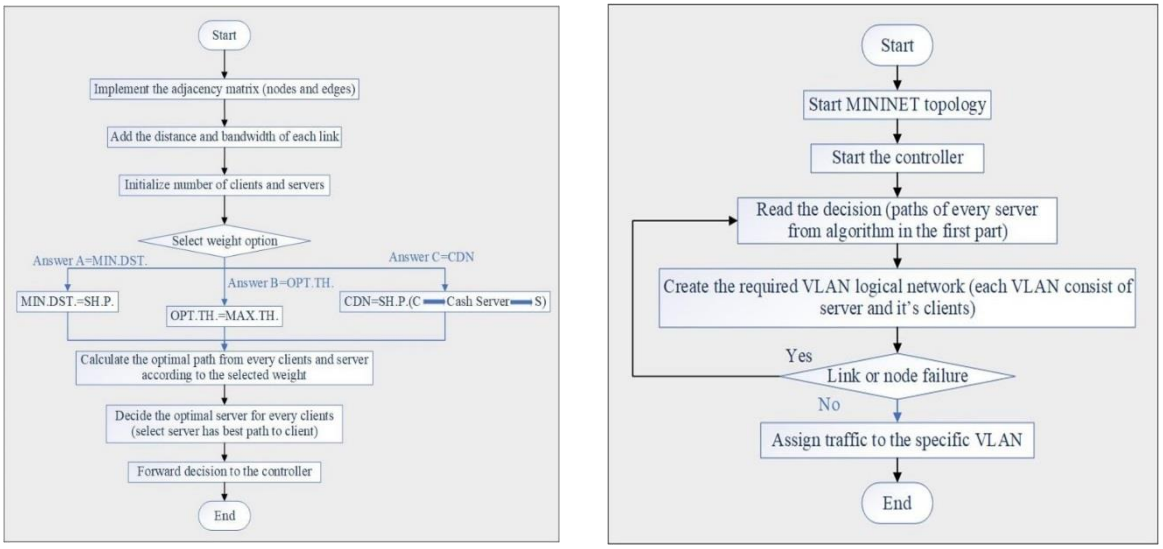
DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

a. In the first option of the algorithm, the clients request the servers that choose the path with the lowest distance to achieve the lowest latency. That means optimizing the path for the packet of a normal volume of data with minimal latency.
b. In the second option of the algorithm, the clients request the servers that choose the path with the best throughput, which means the high volume of data sent via a network in a specific period.

B. Second Part of Framework: Slicing Network using VLAN

The second part of the proposed method that is slicing the network, completes the first part which is the management algorithm in order to optimize the QoS in SD- WAN. In this part, which is applied in the Mininet emulator, the controller receives the decisions obtained from the first part (the VLAN slices) and groups the paths of each server in a single VLAN. Each VLAN consist of the paths between a single server and its clients to provide a better QoS and less broadcast ratio. In case of path changing between the client and server for a specific VLAN due to link or switch failure, the VLAN will request a new path from the controller's application (the proposed algorithm). The algorithm will recalculate a new path, and the VLAN will be updated dynamically according to the received decision from the algorithm. This is accomplished by keeping the controller tied with the decision-making of the algorithm mentioned in the first part. After creating VLANs, the broadcast ratio, bandwidth, jitter, packet loss, and latency will be tested. Then compare the results with and without VLAN. The mentioned procedure is detailed in the following steps, as shown in *Fig. (1-B)*:

1. Start MININET topology.
2. Start the POX controller, and read the VLANs of every server.
3. Create VLAN according to the received decisions of the first part and cluster the paths of each server and its clients in a single VLAN by the controller.
4. The controller checks any change in the topology, if any; will request a new path from the controller's application (algorithm).
5. The controller will update the VLAN dynamically according to the new decision.



(A) FIRST PART: DESIGN MANAGEMENT ALGORITHM (B) SECOND PART: SLICING SD-WAN USING VLAN

FIG. 1. PROPOSED SYSTEM FLOWCHART.

Received 22/September/2022; Accepted 29/October/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

IV. IMPLEMENTATION AND RESULTS

A. Testbed Components

In the first part of framework, create an algorithm utilizing the MATLAB programming language [24] and a built graph theory algorithm that can determine the best network segment to connect a group of clients to the necessary server. To achieve the second part of framework using the MININET emulator [25], the virtual machine hypervisor called VM virtual box will be used. Using the remote toolbox called MobaXterm is the best way to log in to the inner file of MININET and easily access controllers, clients, servers and hosts. This remote server can access the VM and its hostess MININET. The simple topology is the program generated and implemented in MININET. This topology is executed by classes to generate switches, hosts and links. Also, specifying the properties of each link, especially the bandwidth; if the bandwidth of each link is not specified, the link takes random bandwidth. After implementing the topology, the application of the POX controller [26] has received the decision from the algorithm (first part) in the form of a text file with VLANs and the roles of each switch.

B. Test Methods

There is a need to test several criteria to evaluate this work and prove that it achieves positive results. These criteria are latency, throughput, jitter, packet loss, and broadcast ratio. So, a collection of tests will be applied to examine the mentioned criteria. These steps are mentioned in the following:

1. Custom topology has been created with many servers and clients to test the network's performance between each client and server in two stages.
2. In the first stage, test the topology before slicing. To perform this test, connect one client and its selected server according to the results of the part one algorithm, with zero background traffic in the network. Then, repeat the same steps to create background traffic. Many clients connect to the server to produce the background traffic in three different percent's as the following per (0%, 50% and 75%) to test the network's performance. The background traffic will cross the same connection as the foreground traffic.
3. In the second stage, test the topology after creating slices (VLANs) of SD-WAN and make a similar connection between the same client and servers in stage one. Starting by testing the network with foreground traffic only. Then, repeat the same tests after generating the background traffic in VLAN. The background traffic is produced by connecting multiple clients to the targeted servers with different percent of traffic like (0%, 50% and 75%) similar to the tests in stage one.

The main goal for testing the network before and after generating VLANs is to show the algorithm's performance and the difference in the result. In addition, the aim is to test the network before and after generating the background traffic to show the difference in result and notice how SD-WAN will behave when generating load. All these procedures mentioned above apply to each scenario will be illustrated:

Scenario 1: The latency will test using different tools to ensure the range effectiveness of the developed algorithm.

A. Test the latency using the ping tool. In this test, calculate the average of 20 timestamps between each client and its server.
B. Test the latency using ICMP & Wireshark. This test captures the send and receives packet between client and server to calculate the average of 20 timestamps in each type of traffic.
C. Test the latency using TCP protocol with each type of traffic. That calculates the average latency of sending and receiving the packet.

Received 22/September/2022; Accepted 29/October/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

Scenario 2: UDP protocol tests the throughput, jitter, and packet loss. The Iperf tool is used in the client and server to calculate these parameters.

Scenario 3: Testing the broadcast ratio using ARP protocol. This ratio, illustrated by the receiving frame over transmit frame, is shown in equation (1).

$$\textbf{Broadcast Ratio} = \frac{Receive\ frame}{Trinsmit\ frame} \quad (1)$$

The number of the received frame is the sum of the successful reception frames.

C. **Result of the First Part of Framework "Design Management Algorithm."**

This section discusses the results of the dynamic method and tests the proposed system algorithm and it's created logical networks. The result of the first and second parts of the framework will be discussed in detail and show the difference in results with and without slicing.

Simple topology: this topology contains 14 switches and 14 links, as shown in *Fig. 2*. The topology graph presented in *Fig. 2* is created by MATLAB graph plotting.

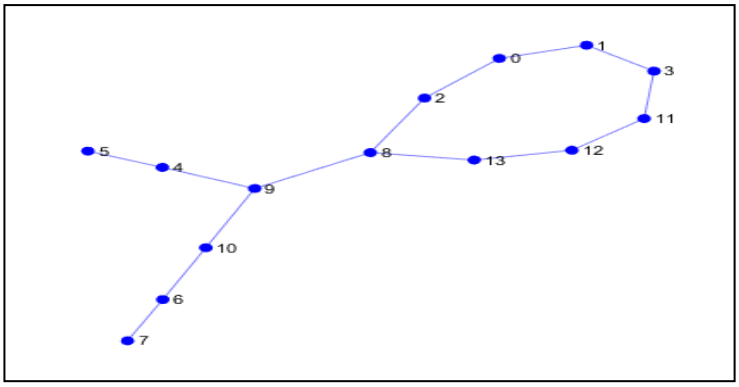


FIG. 2. SIMPLE TOPOLOGY IN MATLAB.

The results of the algorithm created six different paths between the server and their clients in Table I. Table I shows the results when the optimization weight is the minimum distance or the maximum throughput of each path in topology.

TABLE I. SUMMARY RESULTS OF TOPOLOGY IN MATLAB

Path no.	Client	Path of minimum distance	Optimal server of minimum distance	Minimum distance (km)	Path of highest throughput	Optimal server of highest throughput	Highest throughput (Mbps)
Path1	Node 0	0,2	Node 2	200	0,2,8,9,10	Node 10	4000
Path 2	Node 1	1,0,2	Node 2	300	1,0,2,8,9,10	Node 10	5000
Path 3	Node 4	4,9	Node 9	200	4,9,8,2	Node 2	3000
Path 4	Node 5	5,4,9	Node 9	300	5,4,9,8,2	Node 2	4000
Path 5	Node 6	6,10	Node 10	100	6,10,9,8,2	Node 2	4000
Path 6	Node 7	7,6,10	Node 10	200	7,6,10,9,8,2	Node 2	5000

1-(Part One Result of Topo1- "Path 1 of 6"):

Fig. (3-A) shows the client at "node 0". When applying the algorithm with the option of the minimum distance, the client at "node 0" has selected the server at "node 2" as the optimal server with the path which is presented in the figure below. Hence, the selected path will achieve lower latency between them. While, the client at node zero will select the server at "node 9" as the optimal server when the algorithm is applied with the route of the

Received 22/September/2022; Accepted 29/October/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

highest throughput, as shown in *Fig. (3-B)*. The path shown in this group of figures are presented in detail previously in the Table I.

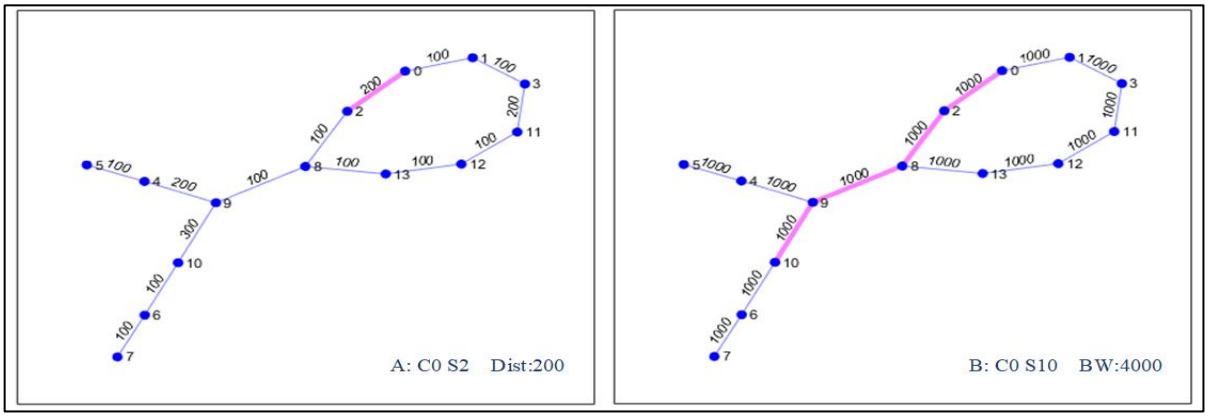


FIG. 3. (A): GRAPH PRESENTS THE NEAREST SERVER "SERVER 2" TO CLIENT (0). (B): GRAPH PRESENTS THE SERVER OF THE HIGHEST CAPACITY PATH "SERVER 10" TO CLIENT (0).

2-(Part One Result of Topo1- "Path 2 of 6"):

Fig. (4-A) shows the client at "node 1". The optimal server for "node 1" is the server at "node 2", in case the algorithm is applied to select the minimum distance to enhance the latency. Otherwise, when the client at "node 1" requests services with the highest throughput, the optimal server is "node 2", as highlighted in *Fig. (4-B)*.

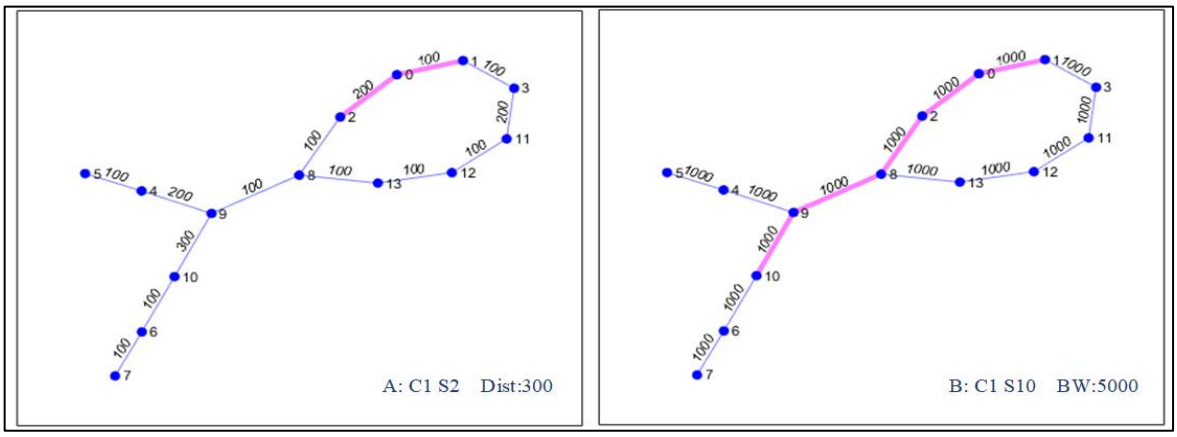


FIG. 4. (A): GRAPH PRESENTS THE NEAREST SERVER "SERVER 2" TO CLIENT (1). (B): GRAPH PRESENTS THE SERVER OF THE HIGHEST CAPACITY PATH "SERVER 10" TO CLIENT (1).

3-(Part One Result of Topo1- "Path 3 of 6"):

Fig. (5-A) shows the client at "node 4". When the client at "node 4" requests the services with a minimum distance, select the server at "node 9". This server is the optimal server for the client at "node 4" to achieve the lowest latency. Otherwise, the optimal server for the same client is "node 2" when clients request serves with the highest throughput, as highlighted in *Fig. (5-B)*.

Received 22/September/2022; Accepted 29/October/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

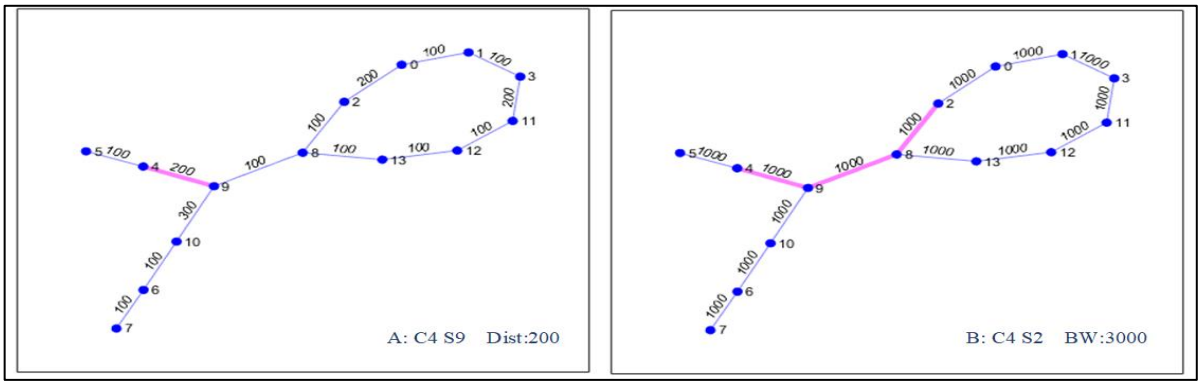


FIG. 5. (A): GRAPH PRESENTS THE NEAREST SERVER "SERVER 9" TO CLIENT (4). (B): GRAPH PRESENTS THE SERVER OF THE HIGHEST CAPACITY PATH "SERVER 2" TO CLIENT (4).

4-(Part One Result of Topo1- "Path 4 of 6"):

Fig. (6-A) shows the client in "node 5". The algorithm selects the optimal server with a minimum distance at "node 9" for the client at "node 5". The optimal server for the same client is the server at "node 2" when the client request serves with the highest throughput, as highlighted in *Fig. (6-B).*

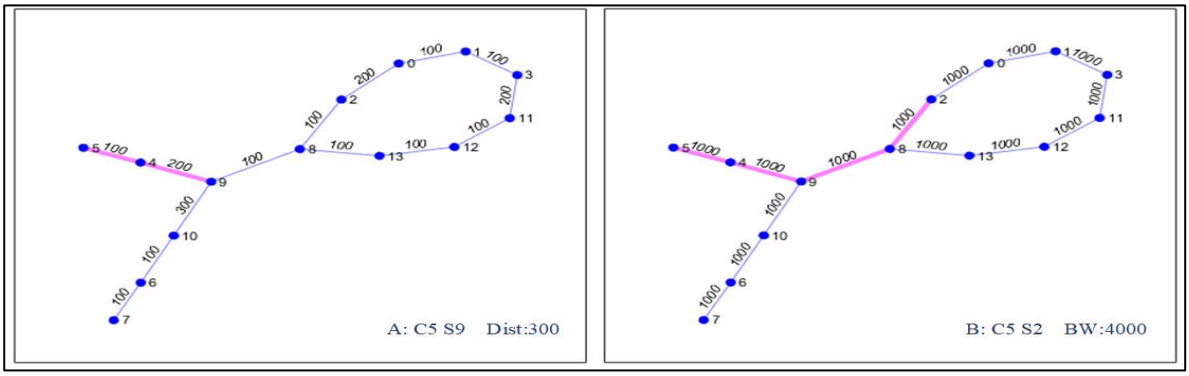


FIG. 6. (A): GRAPH PRESENTS THE NEAREST SERVER "SERVER 9" TO CLIENT (5). (B): GRAPH PRESENTS THE SERVER OF THE HIGHEST CAPACITY PATH "SERVER 2" TO CLIENT (5).

5-(Part One Result of Topo1- "Path 5 of 6"):

Fig. (7-A) presents the client at "node 6". The optimal server for the client at "node 6" requests the services with a minimum distance from the server at "node 10". Otherwise, the optimal server for the same client is the server at "node 2" when clients request services with the highest throughput, as highlighted in Fig. *(7-B).*

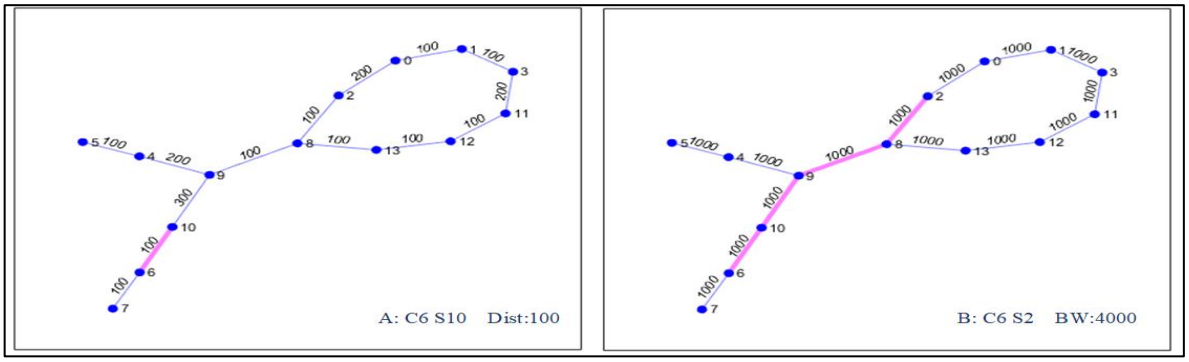


FIG. 7. (A): GRAPH PRESENTED THE NEAREST SERVER "SERVER 10" TO CLIENT (6). (B): GRAPH PRESENT THE SERVER OF THE HIGHEST CAPACITY PATH "SERVER 2" TO CLIENT (6).

Received 22/September/2022; Accepted 29/October/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

6-(Part One Result of Topo1- "Path 6 of 6"):

Fig. (8-A) shows that the optimal server for the client at "node 7" is the server at "node 10" when the client requests the services with a minimum distance. On the other hand, the optimal server for the same client is "node 2" when the client requests services with the highest throughput, as highlighted in *Fig. (8- B)*.

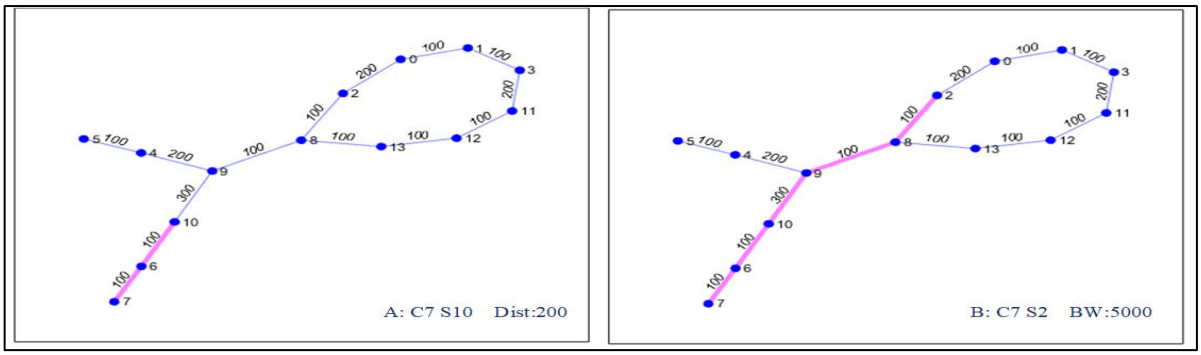


FIG. 8. (A): GRAPH PRESENTS THE NEAREST SERVER "SERVER 10" TO CLIENT (7). (B): GRAPH PRESENTS THE SERVER OF THE HIGHEST CAPACITY PATH "SERVER 2" TO CLIENT (7).

D. Result of the second part "Slicing SD-WAN using VLAN."

This section will present the topology and testing results of topo1 in the MININET emulator, as shown below in *Fig. 9*. These tests include five scenarios which are testing latency (ping, ICMP & Wireshark and TCP), testing (throughput jitter and packet loss) and testing broadcast ratio. It's not worth mentioning that these scenarios are explained above in section (IV.B).

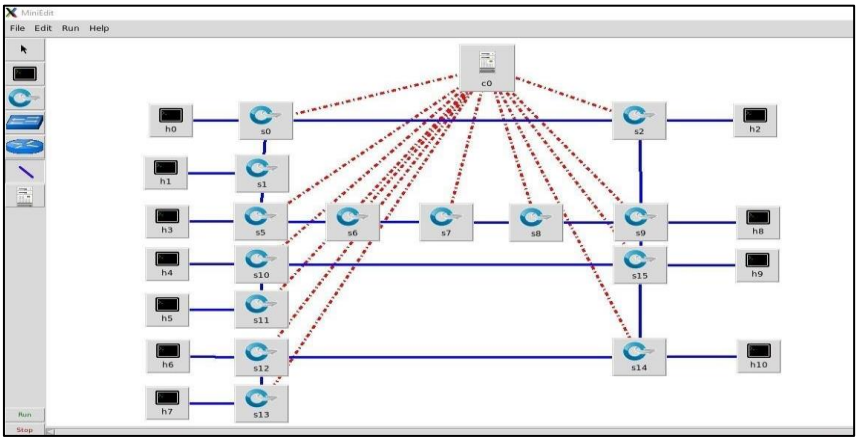


FIG. 9. SIMPLE TOPOLOGY IN MINIEDIT.

1. Testing Latency

The average latency has been tested in three ways (Ping, ICMP& Wireshark, and TCP) discussed in the following:

1.1 PING

Fig. 10 shows the results of latency testing between selected clients and their servers using the ping tool. This result tests the latency before and after applying VLAN. In *Fig. 10*, the y-axis shows the latency in ms, and the x-axis introduces the three states of background traffic (0%, 50% and 75% of link bandwidth in Mbps). The orange colour (diagonal line pattern) shows the result of latency without VLAN, and the blue colour (cross line pattern) shows the result of latency with VLAN. The number (1,2,3) in order refers to the result of average latency between (client 0 → server 2, client 4 → server 9 and client 6 → server 10). The first group of bars (from the left) represents the average latency of twenty

Received 22/September/2022; Accepted 29/October/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

pings when the background traffic is zero. The second and the third groups of bars represented similar latency when the background traffic increased to 500Mbps and 750Mbps sequentially. Analyzing the test results demonstrate that in all cases, the latency after slicing the SD-WAN is better than the latency before creating the slices by an average of 33.48%. Also, it is noticeable that the maximum latency enhancement appears in the group of order 1 when the load is 750Mbps. In group one, the value of latency before applying VLAN is equal to 0.132ms, while after applying VLAN enhanced to 0.073ms.

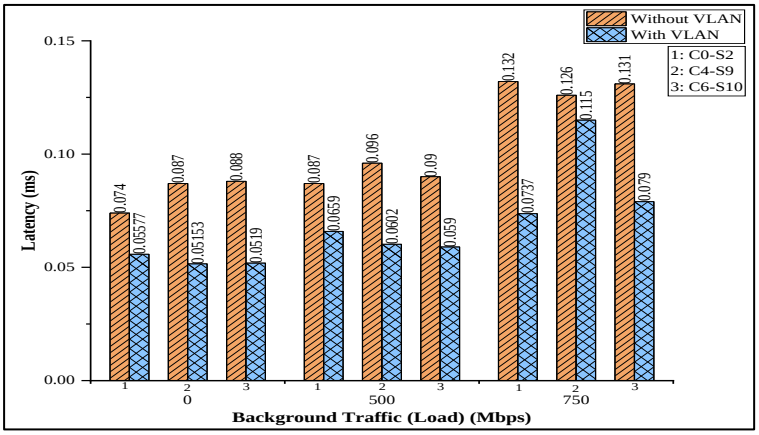


FIG. 10. RESULT OF LATENCY MEASURED BY PING TOOL.

1.2 ICMP & Wireshark

The latency results between each client and their servers using the ICMP and Wireshark, as illustrated in *Fig. 11*. This result tests the latency before and after applying VLAN. As mentioned in paragraph (A), the y-axis shows the latency, and the x-axis shows the background traffic.

The latency result in ICMP& Wireshark introduces similar behavior of latency tests with the ping tool. However, they obtained a little less value than the result of latency in the ping tool. The latency in Wireshark is lower than in ping tests because ping latency has interface delay. For example, in the ping test (group one with load zero), the value of average latency without VLAN=0.074ms and after applying VLAN=0.0557ms, but in ICMP & Wireshark test, the value of average latency to the same group without VLAN=0.037ms and after VLAN the value became 0.033ms.

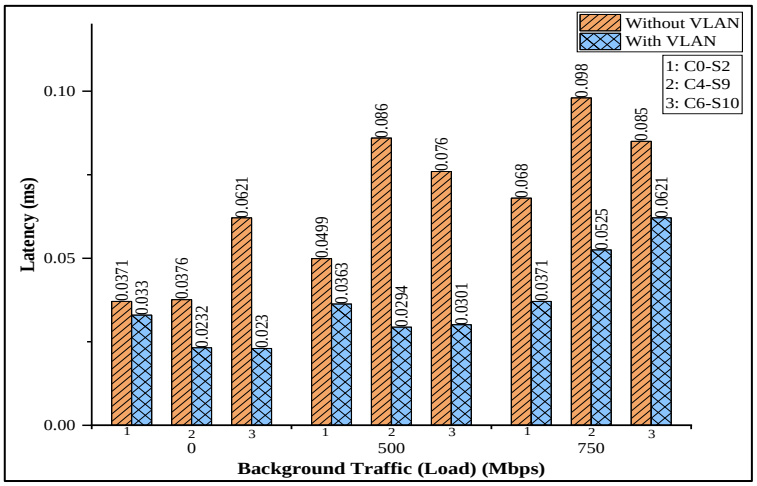


FIG. 11. RESULT OF LATENCY MEASURED BY ICMP & WIRESHARK.

Received 22/September/2022; Accepted 29/October/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

1.3 TCP

TCP is another protocol used to test the latency between client and server. This protocol analyzes the packet flows to calculate latency. *Fig. 12* has the same description as the ping and Wireshark tools Fig. . The results in *Fig. 12* announced that the average enhancement after applying VLAN is 42.85%.

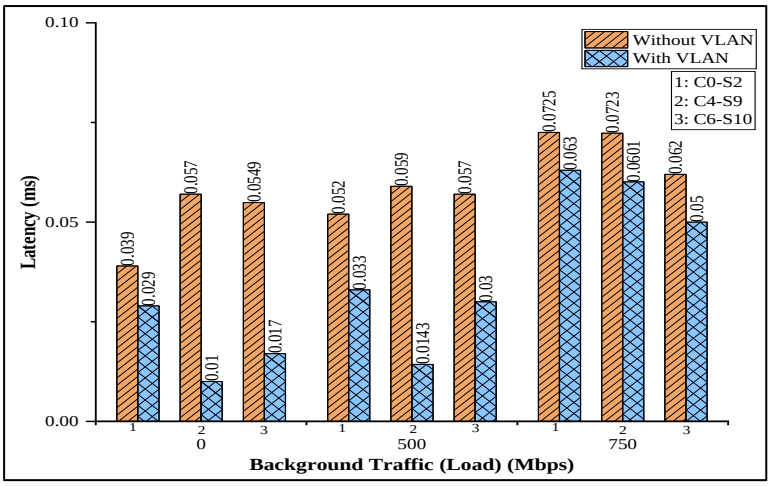


FIG. 12. RESULTS OF LATENCY MEASURED BY TCP.

Such as, in group 3, when the load value is 750Mbps, the average latency before applying VLAN= 0.062ms. This value was enhanced to 0.05ms after applying the VLAN.

2. Testing (Throughput, Jitter, Packet loss)

2.1 Throughput

Fig. 13 shows the throughput results between selected clients and their servers using the UDP protocol. The y-axis presents the throughput in Mbps, and the x-axis presents the three states of background traffic (0%, 50% and 75% of link bandwidth). The dark purple colour (diagonal line pattern) shows the result of throughput without VLAN, and the violate colour (cross line pattern) shows the result of throughput with VLAN.

The throughput was tested using the Iperf tool between each client and its server. To analyze the results presented in *Fig. 13* that in all cases, the throughput value increased after applying VLAN by an average of 1.53%. To be specified, it enables the maximum bits per second (bps) that a VLAN may receive. In the case of load zero (group one), the value of throughput before and after applying VLAN equals 100Mbps; when a load= 500Mbps, the throughput value without VLAN became 97.7Mbps.

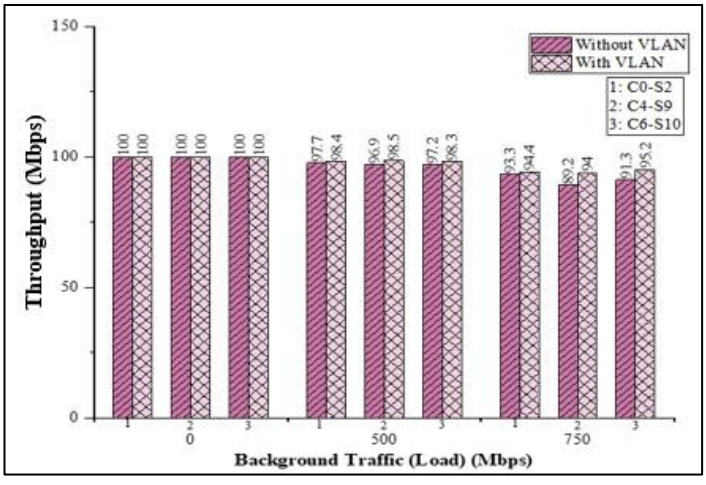


FIG. 13. RESULTS OF THROUGHPUT MEASURED BY UDP.

Received 22/September/2022; Accepted 29/October/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

While after VLAN was applied, the throughput value increased to 98.4Mbps. In addition, when load =750 Mbps, the value of throughput before applying VLAN decreased to 93.3Mbp but improved to 94.4 Mbps after applying VLAN.

2.2 Jitter

Jitter tested using UDP protocol by Iperf tool. In *Fig. 14*, the y-axis indicates the jitter values, and the x-axis indicates the background traffic. The dark purple colour (diagonal line pattern) shows the result of jitter without VLAN, and the violate colour (cross line pattern) shows the result of jitter with VLAN.

The results show the jitter was enhanced by 47.22% after applying VLAN between each client and server. The jitter in the network results from direct mass broadcast data transmission to every device in the network. With VLAN, it is set up the appropriate communication equipment in each VLAN, cutting down on broadcast traffic and improving network performance. The jitter value in group one was enhanced after applying VLAN to each type of load. Such as, in group 1, when the load is zero, the value of jitter before applying=0.025ms and after applying VLAN optimized to =0.02ms. When load increased to 750Mbps, the value of jitter also increased before applying VLAN but optimized to 0.019ms after applying VLAN.

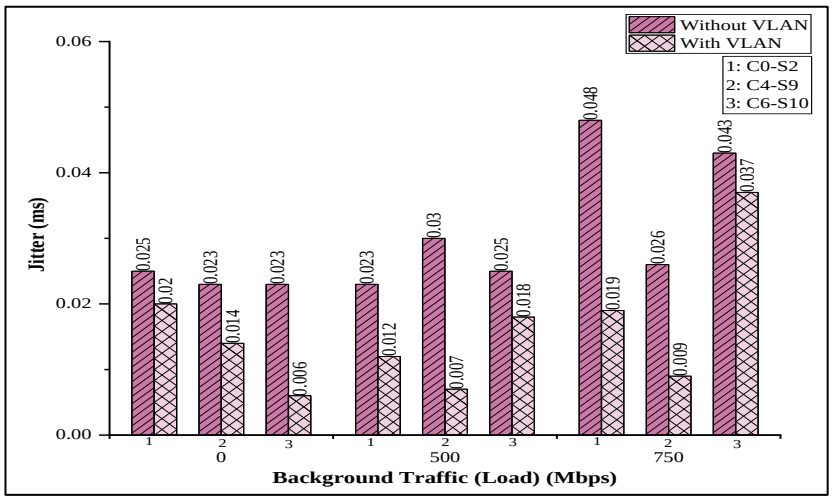


FIG. 14. RESULT OF JITTER MEASURED BY UDP.

2.3 Packet loss

Packet loss is also tested using the UDP protocol between each client and server. *Fig. 15* display the results before and after applying VLAN. The y-axis views the packet loss percentage, and the x-axis views the background traffic. The dark purple colour (diagonal line pattern) views the result of packet loss without VLAN, and the violate colour (cross line pattern) views the result of packet loss with VLAN. As a result, the packet loss percentage is decreased after applying VLAN by 58.18%. To show that, in load 750Mbps, the value of packet loss before VLAN is equal to 1.8%; otherwise, when applying VLAN, the value of packet loss is enhanced to 0.76%. It is decreased after applying VLAN.

Received 22/September/2022; Accepted 29/October/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

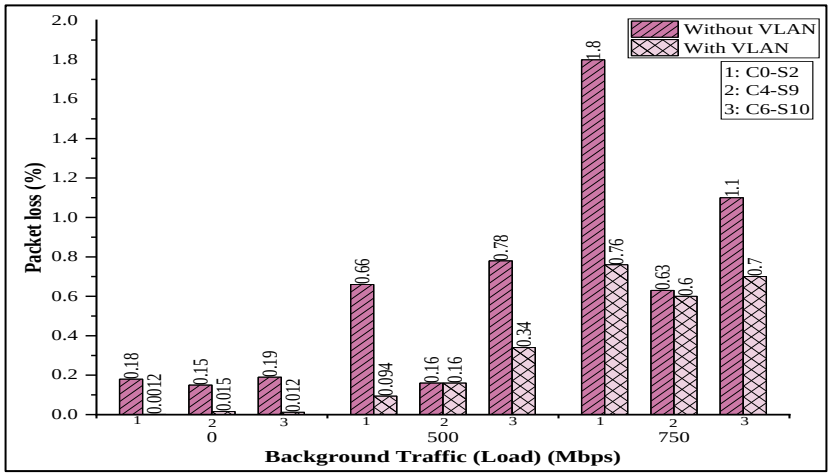


FIG. 15. RESULT OF PACKET LOSS MEASURED BY UDP.

3. Broadcast Ratio

Test the broadcast using ARP protocol. *Fig. 16* shows the result of the broadcast ratio after and before applying VLAN. in this Fig. , the y-axis shows the broadcast ratio, and the x-axis shows the nodes of each group of VLANs (clients and server). The star shape (blue colour) shows the result of the broadcast ratio with VLAN, and the square shape (black colour) shows the result without VLAN.

This ratio is calculated by dividing the received frame by the transmitted frame. The results show the ratio before applying VLAN is (9:1), then optimized to (2:1) after applying

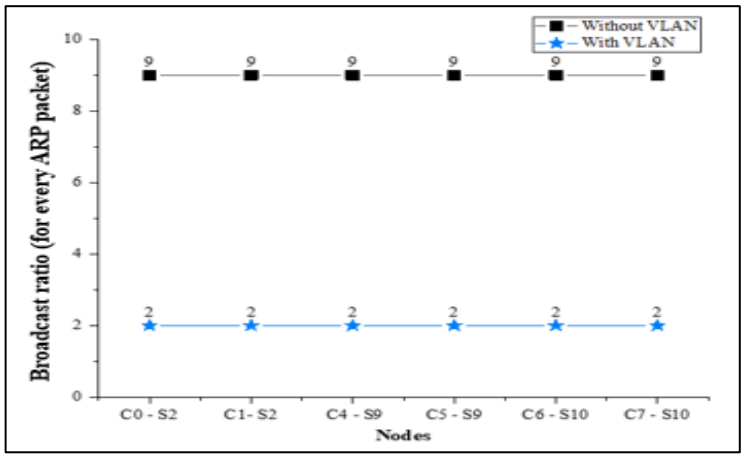


FIG. 16. RESULT OF BROADCAST RATIO.

VLAN. The ratio of broadcast is (2:1) in VLAN 1 (c0- s2) because this VLAN contains only two hosts, but without VLAN the ratio (9:1) refers to all number of hosts in topology, this ratio is increased when the number of hosts increased in topology. The percentage of enhancement in broadcast ratio after applying slicing by 77.77%. That means VLAN divides large broadcast domains into smaller ones, enhancing network performance. If a device in one VLAN sends a broadcast frame, all devices in the VLAN receive the frame, but devices in other VLANs do not.

Received 22/September/2022; Accepted 29/October/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

V. CONCLUSION

The huge increment in the number of internet users, devices and services led to an insatiable performance in the current network in the near future. Therefore, the network developer works to introduce a new programmable network which is SDN. The primary advantage of SDN is that it enables dynamic management, such as creating dynamic clustering. This work noticed this privilege to design the framework able to separate every server with its clients in a dedicated virtual network. The project started by conducting a wide range literature survey about using VLAN with SDN network. The performed survey proves that there is a missing in the research that developed a suitable algorithm to slice SD-WAN into server-client's logical networks. This increases the need for an algorithm to divide SD-WAN into several virtual networks based on server-client correlation utilizing VLAN. The optimization included an algorithm to maintain the dynamic VLAN. The proposed framework consists of two parts; the first part was the management algorithm designed to select the optimal server for each client. Next, the algorithm clustered each server and its clients into an individual logical network. The second part of the framework includes creating a VLAN for each logical network in SD-WAN to improve its QoS and reduce the broadcasting domain. Finally, the achievement of this study is summarized by the following:

1. An SDN-based QoS management approach to improve QoS by slicing the network into an individual logical network. The management algorithm places the optimal server for each client. The evaluation results show that this approach can significantly improve the latency and throughput.

2. A mechanism for providing end-to-end QoS to each client. Slicing the network into groups of VLANs and dedicating each VLAN to the type of each server and its clients. This mechanism reduces latency, jitter, packet loss, and broadcast ratio. In addition, enhancing the throughput of each VLAN.

3. The proposed algorithm is implemented over simple topology with three cases of load (0%, 50%, 75%). The results are evaluated as a comparison between SD-WAN performance before and after applying the VLAN. The simulations showed that the algorithm has the optimization of about 42.85% of latency, and 4.61% of throughput, and 47.86% of jitter, and 72% of loss packet. The biggest benefit comes from lowering the broadcasting ratio by 77.77%.

ACKNOWLEDGMENT

This work was supported in part by the Iraqi Ministry of Higher Education and Scientific Research – the University of Technology- Iraq, which represents the institution of the first and second authors.

REFERENCES

[1] H. S. Yaseen and A. Al-saadi, “Load Balancing and Detection of Distributed Denial of Service Attacks Using Entropy Detection,” vol. 21, no. 4, pp. 60–73, 2021.

[2] R. Masoudi and A. Ghaffari, “Software defined networks: A survey,” *J. Netw. Comput. Appl.*, vol. 67, pp. 1–25, 2016.

[3] A. M. Al-Sadi, A. Al-Sherbaz, J. Xue, and S. Turner, “Developing an asynchronous technique to evaluate the performance of SDN HP Aruba switch and OVS,” *Adv. Intell. Syst. Comput.*, vol. 857, no. July, pp. 569–580, 2019.

[4] J. Suhail and K. S. Rijab, “Enhance the Performance of Wireless Sensor Network based on Software Define Network and Kalman Filter,” *Eng. Technol.*, vol. 18, no. Special Issue, pp. 1436–1448, 2021.

[5] S. R. B. P. A. Inigo Matthew, “A Comprehensive Analysis of Virtaul Local Area Network (VLAN) & Inter-VLAN Routing Strategies,” *Tech. Reasearch Orgainisation India*, vol. 11, no. 1 (61), pp. 14–16, 2016.

[6] R. Vadivelu and S. Malathy, “Design and Performance Analysis of Complex Switching Networks through VLAN , HSRP and Link Aggregation,” *Int. J. Printing, Packag. Allied Sci.*, vol. 5, no. 1, pp. 139–148, 2017.

[7] A. I. Mathew and S. R. B. Prabhu, “A Study on Virtual Local Area Network (Vlan) and Inter-Vlan Routing,” *Int. J.*

Received 22/September/2022; Accepted 29/October/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.13

Curr. Eng. Sci. Res., vol. 4, no. 10, pp. 2393–2395, 2017.

[8] N. L. Sowjanya and R. Anitha, "An efficient VLAN implementation to decrease traffic load in a network," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 2147–2153, 2020.

[9] J. Xiaowei, L. Zhimin, and Wenlong, "Application of Routing Communication between VLANs in A Layer 3 Switch," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 569, no. 3, 2019.

[10] A. S. Dawood and M. N. Abdullah, "A Survey and Comparative Study on Software-Defined Networking," *Int. Res. J. Comput. Sci.*, vol. 3, no. no.8, pp. 1–10, 2016.

[11] I. Danda B. Rawat, Senior Member, IEEE, and Swetha R. Reddy, Member, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 1, p. 22, 2017.

[12] V. Thirupathi, C. Sandeep, S. Naresh Kumar, and P. Pramod Kumar, "A comprehensive review on sdn architecture, applications and major benifits of SDN," *Int. J. Adv. Sci. Technol.*, vol. 28, no. 20, pp. 607–614, 2019.

[13] M. Waheed, A. Saeed, and T. Abd, "Signaling Load Reduction in 5G Network and Beyond," *Eng. Technol. J.*, vol. 39, no. 10, pp. 1481–1491, 2021.

[14] A. Mousa and M. Abdullah, "A Survey on Load Balancing, Routing, and Congestion in SDN," *Eng. Technol. J.*, vol. 40, no. 10, pp. 1–11, 2022.

[15] T. A. Assegie and P. S. Nair, "A review on software defined network security risks and challenges," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 6, pp. 3168–3174, 2019.

[16] M. K. Faraj, A. Al-saadi, and R. J. Albahadili, "An Investigation of Using Traffic Load in SDN Based Load Balancing," *Iraqi J. Comput. Commun. Control Syst. Eng.*, no. April, pp. 65–74, 2020.

[17] V.-G. N. and Y.-H. Kim, "sdn based enterprised and compus network a case of vlan management.pdf," *J. Inf. Process. Syst.*, vol. 12, p. 14, 2016.

[18] M. B. Lehocine and M. Batouche, "Flexibility of managing VLAN filtering and segmentation in SDN networks," *2017 Int. Symp. Networks, Comput. Commun. ISNCC 2017*, 2017.

[19] O. M. Poncea, F. Moldoveanu, and V. Asavei, "VLAN-PSSR: Port-switching based source routing using Vlan tags in SDN data centers," *UPB Sci. Bull. Ser. C Electr. Eng. Comput. Sci.*, vol. 79, no. 2, pp. 15–24, 2017.

[20] D. A. Aziz, "The Importance of VLANs and Trunk Links in Network Communication Areas," *Int. J. Sci. Eng. Res.*, vol. 9, no. 9, pp. 10–15, 2018.

[21] J. Chen, J. Chen, J. Ling, J. Zhou, and W. Zhang, "Link Failure Recovery in SDN: High Efficiency, Strong Scalability and Wide Applicability," *J. Circuits, Syst. Comput.*, vol. 27, no. 6, pp. 1–30, 2018.

[22] A. Hameed and M. Wasim, "On the study of SDN for emulating virtual lans," *2019 8th Int. Conf. Inf. Commun. Technol. ICICT 2019*, pp. 162–167, 2019.

[23] B. Rauf, H. Abbas, A. M. Sheri, W. Iqbal, and A. W. Khan, "Enterprise Integration Patterns in SDN: A Reliable, Fault-Tolerant Communication Framework," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6359–6371, 2021.

[24] Y. K. Yeo and Y. K. Yeo, "Introduction to MATLAB®," *Chem. Eng. Comput. with MATLAB®*, pp. 1–38, 2020.

[25] Arahunashi, A. K., Neethu, S., & Ravish Aradhya, H. V. (2019). "Performance Analysis of Various SDN Controllers in Mininet Emulator". 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT). doi:10.1109/rteict46194.2019.9016.

[26] H. M. Noman and M. N. Jasim, "POX Controller and Open Flow Performance Evaluation in Software Defined Networks (SDN) Using Mininet Emulator," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 881, no. 1, 2020.

Received 22/September/2022; Accepted 29/October/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*