# Network Slicing On Software Defined Network Using Flowvisor and POX Controller To Flowspace Isolation Enforcement

1st M.T. Kurniawan
Industrial Engineering Faculty
Telkom University
Bandung, Indonesia
teguhkurniawan@telkomuniversity.ac.id

2nd Ibram Moszardo
Industrial Engineering Faculty
Telkom University
Bandung, Indonesia
imoszardo@telkomuniversity.ac.id

3rd Ahmad almaarif
Industrial Engineering Faculty
Telkom University
Bandung, Indonesia
ahmadalmaarif@telkomuniversity.ac.id

**Abstract— Software defined network (SDN) is becoming more and more popular due to its feature such as programming control, centralize monitoring, fine-grained control, flexibility, multitenant support and scalability. Problems with the previous architecture called traditional network like network device configuration process is done one by one, decentralize control, and on multitenant support there are still problem for enforcement tenant, the tenants cannot manage their own network without being disturbed by other tenants. This study aims to perform network slicing on SDN to enforce isolation between tenants by using flowvisor and POX Controller. For isolation in this study is flowspace is part of the flowvisor that is able to enforce network isolation. Network slicing method used to multitenant support on SDN. To achieve the research objectives, two types of tenants were used and two testing processes were carried out namely connectivity testing and functionality testing. In both tests, network quality tests were also carried out by calculating the Quality of Service (QoS). The results of the study show that the flow visor can be implemented for isolation enforcement, and results of Quality of Services is good for both testing process. The conclusion from the study is that the use of Flowvisor to enforce Flowspace can slice network to support multi-tenants because each tenant can use their own slice and is not disturbed by other slices. For future study, can be increased the number of slices and is expected to be tested in the real environment.**

***Keyword: Flowspace Isolation, Software Defined Network, Flowvisor, Network Slicing***

## I. INTRODUCTION

In the millennial era like now, there are many kinds of technology that make it easier as a human being in this country, namely Indonesia. Of the many existing technologies, the internet is one of them. The internet has many roles in various fields of technology that are developing because the internet has become the basis or as a bridge for the creation of various technologies that are felt today. As the early architecture of the Internet, the traditional network architecture also has a few drawbacks. On a traditional network, it takes a lot of device configuration and a lot of effort to run properly. Because of these things, there is a technology in the network that can be expected to solve the problem called Software Defined Network or SDN. The existing condition of traditional networks according to [1] is that traditional networks do not have the concept of a control plane so that when the network has become large and more complex, it will be very difficult to manage and develop the network and can hinder existing innovation. Then according to [2] the traditional network requires a very large effort to manage and is not dynamic. In addition, traditional networks are difficult to implement policies or policies and on SDN networks save more operational costs. Why it is necessary to do this research is because it is based on the existing conditions that exist in traditional networks and to support multi-tenants in network architecture. The urgency of conducting this research is due to the benefits obtained from the SDN network and multi-tenant. The benefits of using an SDN network in research according to [3] are that the SDN network costs less during maintenance than traditional networks and network management on the SDN network is easier thanks to the controller compared to traditional networks that have to make changes to every device used. Then according to [4], multi-tenant offers advantages such as cost savings, more optimal efficiency, easier maintenance or maintenance as well as scalability and greater computing capacity. Software Defined Network (SDN) is a new concept of network architecture by separating the control plane from the hardware. The basic concept of SDN architecture makes network configuration easier and more flexible. The most important components in SDN generally consist of two parts, namely the control plane and the data plane [5]. In SDN network, multi-tenancy is one of the important factors to support SDN network. Multi-tenancy is an architectural pattern on a network that aims to allow tenants to be run by the same infrastructure service provider [6]. One method of multi-tenancy is network slicing. Network slicing according to [7] is an approach to networking that is built on the concept of network virtualization which is expected to provide capabilities such as flexibility and modularity. Furthermore, one of the controllers on SDN that can be used is the POX controller. POX controller is an open-source controller that aims to develop SDN networks. POX controllers provide an efficient way to implement the OpenFlow protocol between controllers and switches [8]. Between physical devices (hardware) and software (software) on the internet network there is a Flowvisor approach. Flowvisor is a virtual layer that exists on a computer and works like an operating system that uses a set of commands to run hardware. Flowvisor uses the OpenFlow protocol to control network traffic [9]. The

problem comes when the traditional architecture has many problems and the tenants cannot manage their own network without being disturbed by other tenants on the SDN network topology. Therefore, in this final project, focusing on how to isolate the network topology in the form of enforcement of Flowspace isolation using the network slicing method using Flowvisor. The purpose of Flowspace isolation is to separate the SDN network topology into spaces in the form of slices so that each tenant can control their own space without being disturbed by other tenants to support multi-tenants. Therefore, researchers will conduct research on network slicing using Flowvisor for enforcement of Flowspace isolation on Software Defined Network networks.

## II. Theoretical Basis

### A. Software Defined Network

Software Defined Network (SDN) is a new approach to designing, building and managing computer networks by separating the control plane and data plane. The main concept in SDN is network centralization, where all settings are in the control plane [10]. In an SDN architecture, control operations are centralized in a controller that determines network policies. Many controller platforms are open source such as Floodlight, Open Daylight and Beacon [11]. Service providers can allocate resources to customers through the application layer, configure and modify network policies and logical entities in the control plane, and manage physical network elements in the data plane [11]. The control plane is the brain or controller of the Software Defined Network (SDN), the control plane can be run separately from the data plane [10]. Control Plane is one of the important components on the network that functions to control the network, such as forwarding tables, system configuration, determining routing table information and network management [12]. The Data Plane is another important component that functions to forward packets, decipher packet headers, manage Quality of Service and packet encapsulation [12], and is a network hardware that is specially programmed and fully controlled by the Control Plane [10].

### B. Network Slicing

Network slicing is an approach to networking that is built on the concept of network virtualization which is expected to provide capabilities such as flexibility and modularity. Network slicing uses techniques such as Software Defined Networking (SDN) which aims to create multiple virtual networks, each designed and customized for a set of services that share the same set of requirements, on top of a common network [7]. The isolation system in network slicing is able to create a strong layer that is able to separate each slice with the provisions that have been made previously. Slices in network slicing have their own controller to monitor their slice performance independently. The controller contained in each slice cannot interfere in carrying out actions, be it modification or monitoring of other slices [9].

### C. Flowvisor

Flowvisor exists between physical hardware and software, works like a virtual layer on a computer and works like an operating system that uses a set of commands to run hardware. Flowvisor uses the OpenFlow protocol to control network traffic. Flowvisor controls multiple OpenFlow networks where each network controlled is called a slice. Slices controlled by Flowvisor have one controller. Each controller on one slice cannot control another slice. Flowvisor as a device that is able to enforce isolation on each network separation [9]. Flowvisor creates slices with rich network resources and authorizes controllers on each slice to different controllers and also promotes isolation between slices [13].

### D. Flowspace

A flowspace is a set of slices defined by a set of regions (which may not be contiguous) within the Flowvisor. In general, Flowvisor intercepts the flow of network flows using Flowspace. If given a packet header (one dot), Flowvisor can determine which Flowspace can have it. FlowVisor can separate two slices by ensuring their Flowspaces don't overlap anywhere in the topology; or can decide which switch can be used to communicate from one part to another [9]. Flowspace isolation is one of the isolation that can be done with Flowvisor to ensure that the flow in one slice does not interfere with the flow in another slice [13].

### E. Quality of Services

Quality of Service (QoS) is a method of measuring how good a network quality is and is an attempt to define the characteristics and properties of a service. Quality of Services is used to measure a set of performance attributes that have been specified and associated with a service [14]

## III. Methodology

Systematic problem solving is used to complete research in an orderly and structured manner to achieve the ultimate goal of this research. The following is a research systematic which is described by a flowchart.

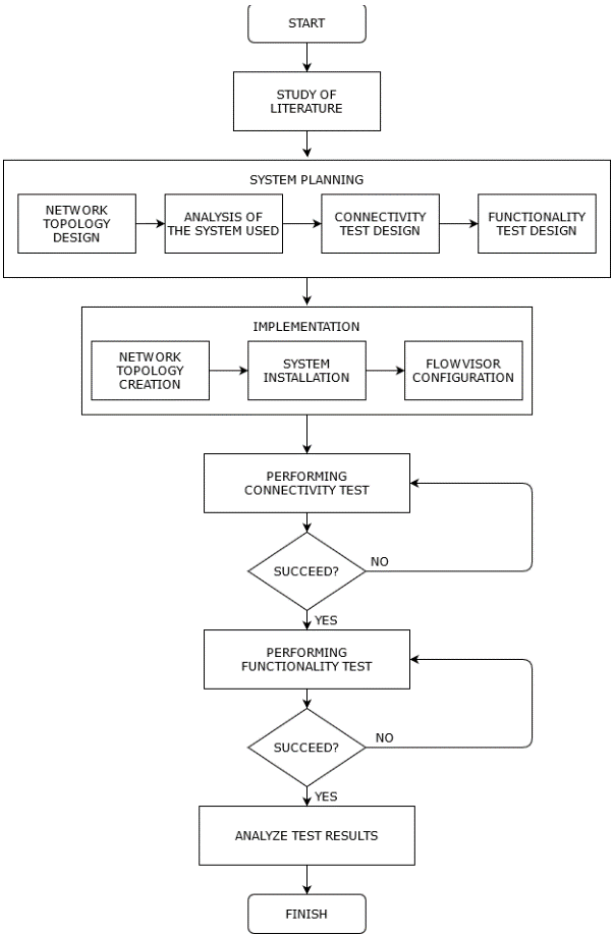


*Figure 1 Systematic Problem Solving*

### A. Study of Literature

At the literature study stage, it begins by identifying and reviewing research-related sciences such as network slicing, SDN, Flowvisor, Flowspace, and quality of services. Literature sources are obtained from previous research journals, trusted websites to YouTube.

### B. System Design

The system design stage is done by designing the SDN network topology that is made, analyzing the system needed at the time of research and designing tests that will be used in research, namely connectivity testing and functionality testing

### C. Implementation

The implementation phase is done by making a network topology on the mininet, installing the system, and configuring the Flowvisor.

### D. Testing and Analysis

In the testing and analysis phase, testing the connectivity and functionality on both slices in the SDN network topology before and after performing Flowspace isolation. Then test and analyze the Quality of Services on the topology before and after Flowspace isolation and test and analyze resource utilization on the topology before using Flowvisor and after using Flowvisor.

### E. Conclusion

The final stage of systematic problem solving is to draw conclusions on the results obtained at the time of research and provide suggestions for further research

## IV. Implementation and Analysis

Making a network topology on a mininet is the basis of a research that is being carried out. The SDN network topology is based on the design as shown in the mininet application. In this network topology Switch 1 (s1), Switch 2 (s2), and Switch 3 (s3) are connected to each other. PC 1 (h1) and PC 2 (h2) are connected to Switch 1 (s1), PC 3 (h3), PC 4 (h4), PC 5 (h5) and PC 6 (h6) are connected to Switch 2 (s2), PC 7 (h7) is connected to Switch 3 (s3), finally PC 8 (h8) and PC 9 (h9) is connected to Switch 4 (s4). The following is an SDN network topology created on Mininet.

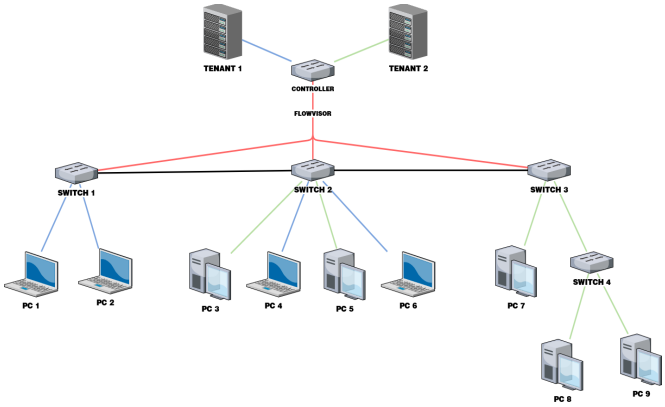


*Figure 2 SDN Network Topology*

### A. Connectivity Testing

Connectivity testing performs connection tests between each host on a network topology without using Flowvisor and performs Quality of Services measurements on an SDN network topology that has not yet isolated. This connectivity test tests whether each host is connected to each other before isolation is carried out at the next stage. Here's one of the pings against fellow tenants and different tenants on connectivity testing.

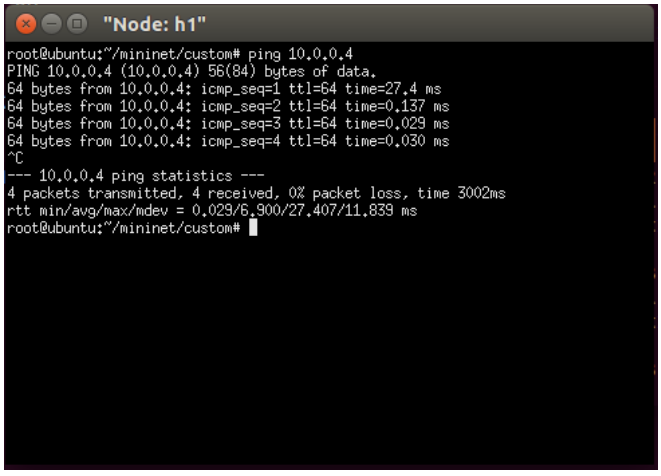


*Figure 3 Ping Fellow Tenants Before Isolation, H1 ping H4*

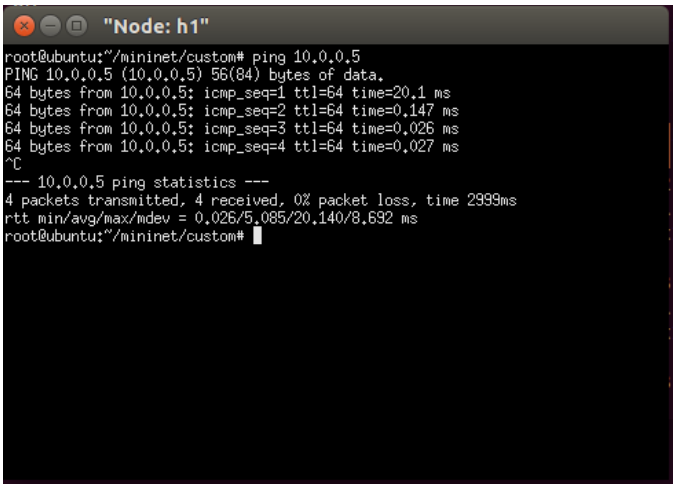


*Figure 4 Ping Difference Tenant Before Isolation, H1 to H5*

## B. *Functionality Testing*

Functionality testing is the core purpose of the research conducted, namely to isolate Flowspace. The purpose of Flowspace isolation is so that each tenant slice can control its own share without being disturbed by other tenant slices. To isolate Flowspace, it is necessary to configure Flowvisor on the network topology so that each existing host has its own space. Here's one of the pings against fellow tenants and different tenants on functionality testing.

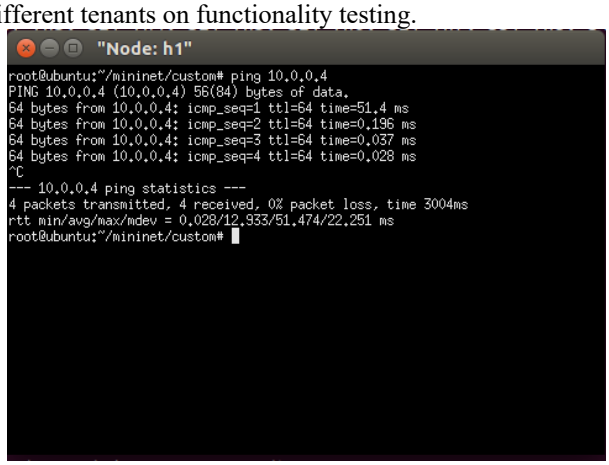


*Figure 5 Ping Same Tenant After Isolation, H1 to H4*

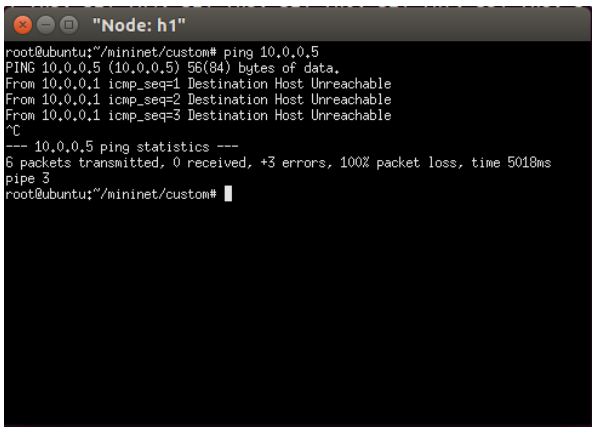


*Figure 6 Ping Difference Tenant After Isolation, H1 to H5*

Based on the picture, each host is no longer able to connect to other hosts but only connects to hosts in the same tenant. Why is this happening because the SDN network topology already enforces Flowspace isolation and tenant 1 and tenant 2 can control their own slices without interfering with each other.

## C. *Quality of Services on Connectivity and Functionality Testing*

After testing the connectivity and functionality, the researchers did a Quality-of-Service comparison between the network topology before isolation and after isolation. Slice 1 and Slice 2 are the results before doing the isolation and Slice 1 – FV and Slice 2 – FV are the results after doing the isolation. The following is the result of throughput comparison on connectivity and functionality testing in tabular form.

*Table IV1 Throughput Comparison on Connectivity and Functionality Testing*

| Bandwidth | Slice 1 | Slice 2 | Slice 1 - FV | Slice 2 - FV |
|---|---|---|---|---|
| 2 Mb | 1,02416 Mb | 1,53339 Mb | 1,02621 Mb | 1,49923 Mb |
| 4 Mb | 2,04650 Mb | 3,07880 Mb | 2,04496 Mb | 3,07608 Mb |
| 6 Mb | 3,06871 Mb | 4,60401 Mb | 3,05318 Mb | 4,61440 Mb |
| 8 Mb | 4,09547 Mb | 6,14161 Mb | 4,10402 Mb | 5,73068 Mb |

Based on the table, in slice tenant 2 in connectivity testing and slice tenant 2 in functionality testing, the throughput value at 6 Mb bandwidth generated in connectivity testing is greater because the SDN network topology has not implemented Flowspace isolation so that there is no prior check when communicating between hosts. In table V.4, the throughput value at bandwidth 2, 4, 6 and 8 has increased along with the increase in bandwidth in both tests. The reason why slice tenant 1 and slice tenant 2 are different is because of the difference in distance between host 6 to host 1 and host 3 to host 9. Slice tenant 2 tends to produce a higher throughput value than slice tenant 1 because host 3 goes to host 9 passes through a 2x switch device, namely switch 3 and switch 4 which results in the actual bandwidth used for data transfer being slightly larger than the slice tenant 1. Next, the researcher provides a comparison of the delay table on connectivity and functionality testing.

*Table IV.2 Comparison of Delay in Testing Connectivity and Functionality*

| Bandwidth | Slice 1 | Slice 2 | Slice 1 - FV | Slice 2 - FV |
|---|---|---|---|---|
| 2 Mb | 1,47619 ms | 0,98634 ms | 1,47298 ms | 0,98142 ms |
| 4 Mb | 0,73558 ms | 0,49145 ms | 0,73930 ms | 0,49131 ms |
| 6 Mb | 0,48004 ms | 0,32831 ms | 0,49528 ms | 0,32767 ms |
| 8 Mb | 0,36783 ms | 0,24598 ms | 0,36845 ms | 0,24552 ms |

Based on the table, slice tenant 1 produces a higher delay than slice tenant 2 because in slice tenant 1 the actual bandwidth used is smaller than slice tenant 2 which results in higher delays. The delay time of slice tenant 2 has decreased because when testing Quality of Services host 3 to host 9 uses maximum bandwidth which results in a decrease in delay time. The reason why the throughput and delay values when testing connectivity and functionality only produces a slight difference is because the communication between hosts used in both tests is the same and in Flowspace isolation according to [9] Flowspace isolation only aims to make the network into a separate slice space according to its tenant. to support multi tenants so that each slice does not interfere with one another, it is different from bandwidth isolation which aims to produce better Quality of Services. Next, the researcher provides a comparison of tables and jitter graphs on connectivity and functionality testing.

*Table IV.3 Jitter Comparison on Connectivity and Functionality Testing*

| Bandwidth | Slice 1 | Slice 2 | Slice 1 - FV | Slice 2 - FV |
|---|---|---|---|---|
| 2 Mb | 0,00719 ms | 0,00590 ms | 0,00235 ms | 0,00039 ms |
| 4 Mb | 0,00051 ms | 0,00121 ms | 0,00206 ms | 0,00115 ms |
| 6 Mb | 0,00170 ms | 0,00161 ms | 0,00156 ms | 0,00081 ms |
| 8 Mb | 0,00022 ms | 0,00013 ms | 0,00093 ms | 0,00021 ms |

Based on the table, the jitter values of slice tenant 1 and slice tenant 2 in the two tests resulted in a significant difference. At the time of connectivity testing, the jitter value tends to be unstable due to collisions between packets because they have not performed Flowspace isolation. At the jitter value, the functionality test experienced a steady decline, namely 0.00235 ms, 0.00206 ms, 0.00156 ms and 0.00093 ms because they had isolated Flowspace on the SDN network and only experienced minor instability on slice tenant 2, namely 0.00039 ms, 0.00115 ms, 0.00081 ms and 0.00021 ms. This reason is based on the statement [5] that SDN networks usually experience many collisions and the traffic load variations are very high. The ordinary SDN network has not been able to separate the TCP and UDP packet nets into separate parts so that the packets that will be sent will experience many collisions which result in ordinary SDN network traffic having a high jitter average value. Collision is data packets colliding with each other during the delivery process.

### *D. Resource Utilization Test Results*

Resource utilization testing is carried out to see the total CPU and memory usage on the SDN network topology without using Flowvisor (connectivity testing stage) and when using Flowvisor (functionality testing stage). The following is a table of results from testing resource utilization.

*Table IV.4 Resource Utilization - CPU*

| No | Time (s) | CPU (%) | |
|---|---|---|---|
| | | Without Flowvisor | With Flowvisor |
| 1 | 10 sec | 17,43% | 13,68% |
| 2 | 20 sec | 29,87% | 24,76% |
| 3 | 30 sec | 37,68% | 35,93% |
| 4 | 40 sec | 44,12% | 50,5% |
| 5 | 50 sec | 50,73% | 70,81% |
| 6 | 60 sec | 55,99% | 82,62% |

Based on the table, CPU usage when using Flowvisor increased higher than CPU usage before using Flowvisor. This is caused by the number of terminals that are run and the burden caused by running Flowvisor. Based on the tests that have been carried out, resource utilization without Flowvisor pingalls and only uses 1 controller with a network topology that has not isolated Flowspace, then on resource utilization using Flowvisor, the CPU usage load increases due to pingall on a network topology that has isolated Flowspace and using 2 controllers, namely ports 10001 and 10002 for 2 different tenant slices and running flowvisor on the new terminal. It can be concluded that the SDN network topology using Flowvisor uses more CPU resources than the SDN network topology without Flowvisor. Next is the average result of memory usage on SDN network topology without Flowvisor and with Flowvisor.

*Table IV.5 Resource Utilization - Memory*

| No | Time (s) | Memory (Mb) | |
|---|---|---|---|
| | | Without Flowvisor | With Flowvisor |
| 1 | 10 sec | 846,57 Mb | 1.031,02 Mb |
| 2 | 20 sec | 826,20 Mb | 1.030,99 Mb |
| 3 | 30 sec | 817,18 Mb | 1.030,96 Mb |
| 4 | 40 sec | 816,96 Mb | 1.023,77 Mb |
| 5 | 50 sec | 816,94 Mb | 1.038,24 Mb |
| 6 | 60 sec | 816,94 Mb | 1.039,36 Mb |

Based on the table, the memory resource usage of the SDN network topology without Flowvisor is lower than the SDN network topology with Flowvisor. The reason why this happens is because the SDN network topology without using Flowvisor (connectivity testing stage) only requires 1 controller and pingalls the network topology that has not been isolated. On the SDN network topology when using Flowvisor (functionality testing stage) requires 2 controllers for 2 slice tenants with ports 10001 and 10002, running Flowvisor on a new terminal and pingall the network topology that has isolated which results in greater memory usage than the network topology SDN without Flowvisor. It can be concluded that the SDN network topology using Flowvisor uses a larger memory resource than the SDN network topology without Flowvisor.

## V. CONCLUSION

In software defined network, multitenant support is mayor concern for scalability. And to archive multitenancy network slicing was implemented. Network slicing on SDN can support multi-tenants was successful because every tenants in network topologies have isolated one and other by Flowspace without interfering with one and other. And Quality of

Services testing carried out on the network topology before isolating and after isolating resulted in throughput, delay and jitter values that were not too much different. This is because the scope of the research carried out is only Flowspace isolation. There are several suggestions that can be used as a reference for further research in the future. Adding the number of hosts, the number of switches and the number of tenants in the network topology created. And the study will implement at the real world.

## REFERENCES


[1] Widianto, M. H. (2019). Sofware Defined Network (SDN) , Konsep Jaringan yang digunakan untuk Startup Terkini | BINUS UNIVERSITY BANDUNG - Kampus Teknologi Kreatif.

[2] IBM. (2021). SDN Versus Traditional Networking Explained | IBM.

[3] Singh, A. K., & Srivastava, S. (n.d.). (2018) A survey and classification of controller placement problem in SDN

[4] Roundy, R. (2020). Benefits of Single vs. Multi Tenant SaaS - DZone Cloud..

[5] Muttaqin, A. R., Yahya, W., & Siregar, R. A. (2018). Implementasi Network Slicing dengan menggunakan Flowvisor untuk Mengontrol Traffic Data Packet pada Jaringan Software Defined Network. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya, 2(2), 793–801.

[6] Bezemer, C. P., Zaidman, A., Platzbeecker, B., Hurkmans, T., & Hart, A. (2010). Enabling multi-tenancy: An industrial experience report. IEEE International Conference on Software Maintenance, ICSM, April 2017.

[7] King, D., & Lee, Y. (2017). Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing.

[8] Kaur, S., Singh, J., & Ghumman, N. S. (2014b). Network Programmability Using POX Controller.

[9] Sherwood, R., Gibb, G., Yap, K., Appenzeller, G., Casado, M., Mckeown, N., & Parulkar, G. (2009). FlowVisor: A Network Virtualization Layer. In Network (p. 15).

[10] Ummah, I., & Abdillah, D. (2016). Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking. In Indonesian Journal on Computing (Indo-JC) (Vol. 1, Issue 1).

[11] Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): a survey. Security and Communication Networks, 9(18), 5803–5833.

[12] Hidayat, I., & Perdana, B. A. (2020). Arsitektur Software Defined Network : Implementasi Pada Small Network (Vol. 01, Issue 01, pp. 1–13).

[13] Oswald, C., & Azodolmolky, S. (2017). Software-Defined Networking with OpenFlow - Second Edition (p. 2). Packt Publishing.

[14] Fahmi, H. (2018). Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik. Jurnal Teknologi Informasi Dan Komunikasi, 7(2), 98–105.