

ANALYSIS OF USER MOBILITY PERFORMANCE ON SOFTWARE DEFINED WIRELESS NETWORK USING DIJKSTRA ALGORITHM

Icha Nurlaela Khoerotunisa¹, Sofia Naning Hertiana², Ridha Muldina Negara³

^{1,2,3}Telecommunication Engineering, School of Electrical Engineering, Telkom University, Indonesia
Email: nurlaelaknisa@telkomuniversity.ac.id, sofiananing@telkomuniversity.ac.id,
ridhanegara@telkomuniversity.ac.id

(Naskah masuk: 13 Februari 2021, diterima untuk diterbitkan: 26 Februari 2021)

Abstract

Over the last decade, wireless devices have developed rapidly until predictions will develop with high complexity and dynamic. So that new capabilities are needed for wireless problems in this problem. Software Defined Network (SDN) is generally a wire-based network, but to meet the needs of users in terms of its implementation, it has begun to introduce a Wireless-based SDN called Software Defined Wireless Network (SDWN) which provides good service quality and reach and higher tools, so as to be able to provide new capabilities to wireless in a high complexity and very dynamic. When SDN is implemented in a wireless network it will require a routing solution that chooses paths due to network complexity. In this paper, SDWN is tested by being applied to mesh topologies of 4,6 and 8 access points (AP) because this topology is very often used in wireless-based networks. To improve network performance, Dijkstra's algorithm is added with the user mobility scheme used is RandomDirection. The Dijkstra algorithm was chosen because it is very effective compared to other algorithms. The performance measured in this study is Quality of Service (QoS), which is a parameter that indicates the quality of data packets in a network. The measurement results obtained show that the QoS value in this study meets the parameters considered by the ITU-T G1010 with a delay value of 1.3 ms for data services and packet loss below 0.1%. When compared with the ITU-T standard, the delay and packet loss fall into the very good category.

Keywords: *dijkstra, quality of service, random direction, Software-Defined Wireless Network, user mobility.*

ANALISIS PERFORMANSI MOBILITAS USER PADA SOFTWARE DEFINED WIRELESS NETWORK MENGGUNAKAN ALGORITMA DIJKSTRA

Abstrak

Selama dekade terakhir, perangkat wireless berkembang sangat pesat hingga di prediksi akan berkembang dengan kompleksitas yang tinggi dan sangat dinamis. Sehingga dibutuhkan kemampuan baru untuk *wireless* dalam menangani masalah tersebut. *Software Defined Network* (SDN) umumnya diterapkan dalam jaringan berbasis *wire*, namun untuk memenuhi kebutuhan user dalam segi fleksibilitas, sehingga mulai memperkenalkan SDN berbasis *Wireless* yang dinamakan *Software Defined Wireless Network* (SDWN) yang memberikan *Quality of Service* yang baik dan mencapai efisiensi dan fleksibilitas yang lebih tinggi, Sehingga mampu memberikan kemampuan baru pada wireless dalam menangani kompleksitas yang tinggi dan sangat dinamis. Saat SDN di terapkan dalam jaringan wireless maka akan dibutuhkan solusi perutean yang dalam memilih jalur diakibatkan kompleksitas jaringan. Pada paper ini, SDWN diuji dengan diterapkan pada pada topologi mesh 4,6 dan 8 access point (AP) karena topologi ini sangat sering dipakai pada jaringan berbasis *wireless*. Untuk meningkatkan performansi jaringan, ditambahkan algoritma Dijkstra dengan skema mobilitas pengguna yang digunakan adalah RandomDirection. Algoritma dijkstra dipilih dikarenakan sangat efektif dibandingkan algoritma lain. Performansi yang diukur dalam penelitian ini adalah *Quality of Service* (QoS) yaitu suatu parameter yang menunjukkan kualitas paket data dalam sebuah jaringan. Hasil pengukuran yang didapat menunjukkan nilai QoS pada penelitian ini memenuhi parameter yang direkomendasikan oleh ITU-T G1010 dengan nilai *delay* terkecil adalah 1.3 ms untuk layanan data dan *packet loss* dibawah 0.1%. Jika dibandingkan dengan standar ITU-T, maka *delay* dan *packet loss* termasuk ke dalam kategori sangat baik.

Kata kunci: *dijkstra, mobilitas user, quality of service, Software-Defined Wireless Network, random Direction.*

1. INTRODUCTION

Over the last decade, wireless applications and mobile devices have developed rapidly to provide network connectivity to users without being connected to a wired network (Crow and Corporation 1997). The most popular is the Wireless Local Area Network (WLAN), that is standardized by IEEE 802.11 which has the advantages of interoperability, mobility, and flexibility [1][2] In the new standard, IEEE 802.11n is capable of produce maximum throughput [3] The next-generation WLAN will work in a dense and very dynamic network. Flexible network architectures and programmability such as the paradigm *Software Defined Network* (SDN) will give WLANs new capabilities to handle user requests while increasing the level efficiency and flexibility in such dense, dynamic and complex networks. [4]

To support networks in the future, wireless applications will develop increasingly complex and it is predicted that there will be a high migration from wire-based networks to wireless-based networks. so that wireless networks will experience high complexity in the future. so that new capabilities are needed to handle this complexity. One of the technologies that can handle network complexity is SDN. SDN is able to provide centralized programming so that it can handle network complexity in wireless, so implementing SDN will be a solution to the challenges of wireless in the future.

The SDN concept began to be applied to wireless network technology, namely Software Defined Wireless Network (SDWN). [5] SDWN is a new paradigm in networking, where SDWN has a concept to separate a control plane and data plane. Where in a traditional network the control and forwarding functions are in the same device. This separation provides advantages such as more centralized network control and provides flexibility in a network.

At SDWN during the delivery process, a package must choose the path to be passed. The complexity of the communication path includes bandwidth requirements, network physical configuration, processing time, packets from each device, packet delivery routing, and so on. Network flow optimization is one of the main problems related to selecting the shortest route, so a routing algorithm that can find the shortest route from a network is needed. One of the routing algorithms that works based on the selection of the shortest route is the Dijkstra algorithm [6] The Dijkstra algorithm has the advantage of being able to minimize delay and increase throughput and is easy to modify so that it can choose a path from one node to another with the shortest distance. This is very important if the graph represent large network where

the speed of data transfers between devices is an important value to pay attention to.

In previous research [7], it has proven that the Dijkstra algorithm can be applied to Local Access Network (LAN) and can minimize delay up to 5.17 ms and increase minimize packet loss up to 24.81%, but this research has shortcomings that have not been applied to wireless transmission media. Meanwhile, in previous SDWN research [8], research has been carried out on its effect on Quality of Service (QoS) based on the perspective of the user mobility model. Mininet-Wifi. However, this study has not discussed the network that uses the routing algorithm and the packet loss obtained is not optimal where the delay value obtained is 3.047 ms and the packet loss is 0.08%. However, there is no research that discusses wireless networks that use the dijkstra routing algorithm and are implemented in SDN networks. How is the network performance if Dijkstra's algorithm is applied to SDWN? How to compare the effect of user mobility on services in the form of data in SDWN if the yahoo jkstra is applied?

Therefore, it is necessary to do research using the Dijkstra algorithm to improve performance, where the author proposes to conduct research on the performance analysis of SDWN using the Dijkstra algorithm with ONOS as the controller in order to be able to improve performance when SDN is applied to wireless wireless based networks.

2. METHOD

The system design will be carried out by simulating a network emulator called mininet-wifi, which contains a Linux-based operating system and uses ONOS as the controller. Then it is necessary to design a topology which will be carried out several tests and compared to see the ability of the network to handle the movement of hosts (station) that move from one point to another in access point (AP) coverage. QoS is evaluated by adding various services such as data to see SDWN performance to ensure service availability. In this test, the 2.4 GHz frequency was selected based on the IEEE 802.11n standard which provides speeds of up to 100 Mbps. so that the access point and station parameters are based on the IEEE 802.11n standard. at this frequency the range of the access point is 70 km. [9]

Dijkstra algorithm is added because the dijkstra algorithm has the advantage of being able to minimize packet loss and increase throughput. In addition, the Dijkstra algorithm is more effective than other routing algorithms based on research[10][11].

The flowchart of the system design can be seen based on the flowchart in Figure 1.

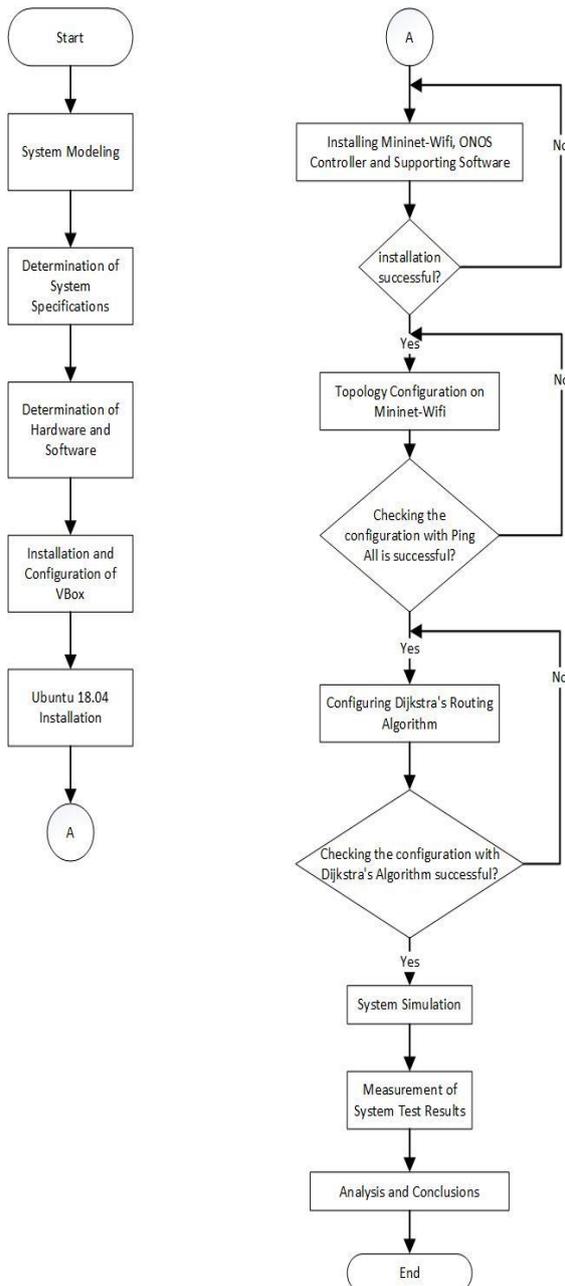


Figure 1. Flowchart of Research Stages

The Flowchart of the Dijkstra Algorithm work be seen based on the flowchart in Figure 2

This system simulates an SDN-based network using ONOS as the controller and Mininet-Wifi emulator as the data plane. Three topologies are tested to access the network's ability to handle movement of hosts (station) moving from one point to another in AP coverage. Sequentially denoted by Topologi 1 (TP1), Topology 2 (TP2), and Topology 3 (TP3) with the number of access points in the order of 4,6,8 AP. As seen in figure 3,4, and 5.

The test scenario is carried out by sending traffic in the form of data services from the host (station 1) which acts as a client to station 3 who is in charge of the server on TP1. For the radio link parameters of the end-device on the SDWN in Table 1.

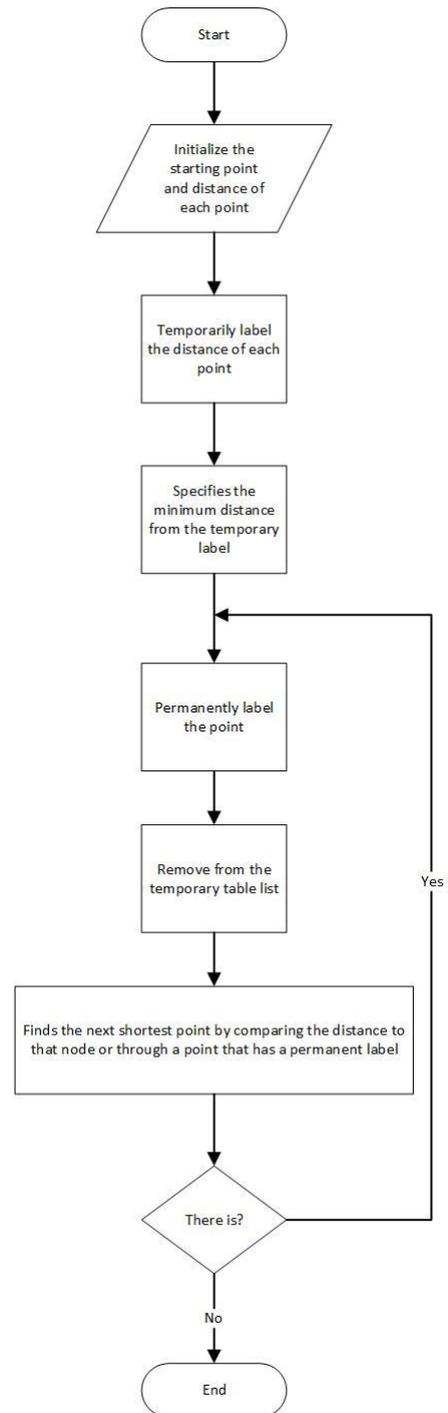


Figure 2. Flowchart of Dijkstra Algorithm

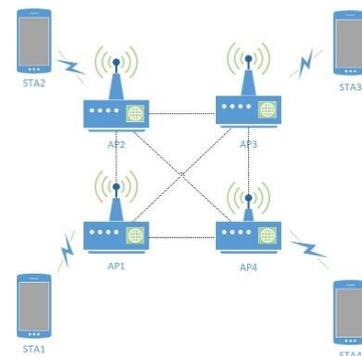


Figure 3. TP1

Table 1. Station TP1 Parameters

Station Parameters	Station 1	Station 3
Frequency	2412 Hz	2412 Hz
Signal	-36 dBm	-36 dBm
TX bitrate	6,5 Mbit/s	6,5 Mbit/s
BSS flags	Short-slot-time	Short-slot-time

Table 3. Station TP3 Parameters

Station Parameters	Station 3	Station 1
Frequency	2412 Hz	2412 Hz
Signal	-36 dBm	-36 dBm
TX bitrate	6,5 Mbit/s	6,5 Mbit/s
BSS flags	Short-slot-time	Short-slot-time

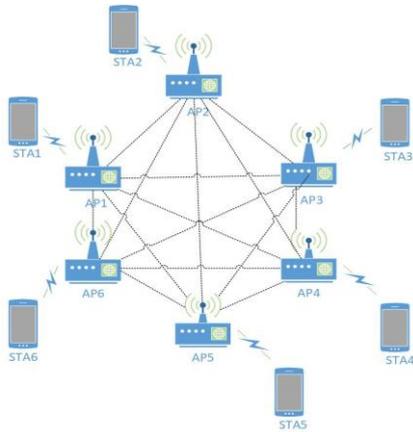


Figure 4. TP2

Meanwhile, the testing scenario for TP2 is done by sending traffic in the form of data services from the host (station 6) which acts as a client to station2 that is in charge of the server. For the radio link parameters of the end-device on this SDWN in Table 2.

Table 2. Station TP2 Parameters

Station Parameters	Station 6	Station 2
Frequency	2412 Hz	2412 Hz
Signal	-36 dBm	-36 dBm
TX bitrate	6,5 Mbit/s	6,5 Mbit/s
BSS flags	Short-slot-time	Short-slot-time

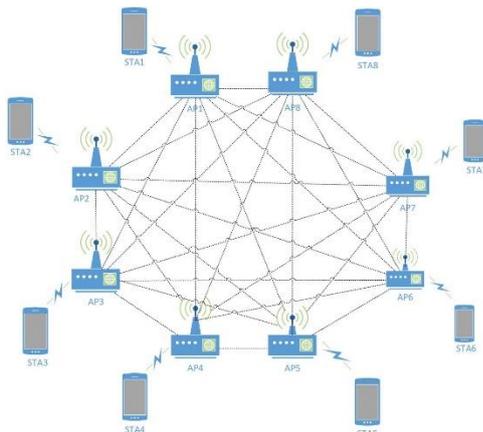


Figure 5. TP3

The test scenario is carried out by sending The test scenario is carried out by sending traffic in the form of data services from the host (station 3) which acts as a client to station1 who serves as a server on TP1 . For the radio link parameters of the end-device on this SDWN in Table 3.

In research measurement, This generated UDP traffic types, namely data traffic with an inter-departure time (IDT) of 100 pps and a packet size of 48 bytes using a Poisson distribution, so that the required bandwidth is 38.4 Kbps [12] The movement of nodes refers to the model provided by Mininet-Wifi. The movement model at nodes are based on the Random Direction model, this model is created to overcome the average density wave generated by the Random Way Point model. Density wave is a grouping of nodes in one part of the simulation area. This model uses a series of probabilities to determine the next position of the moving node. This model uses a probability matrix which defines the probability of a node moving forward, backward, or remaining stationary in the x and y-axis directions. After the direction of travel has been determined, the node will move at a constant speed for the specified time. [13]

In this study, the evaluation is based on Quality of Service (QoS). QoS is a parameter that determines the quality of data packets from a network. QoS is provided to ensure a user gets reliable performance. The parameters of QoS such as *delay*, *jitter*, *packet loss* and *throughput* [14]. In this study, QoS refers to the ITU-T G1010 standard.

Table 4. QoS ITU-T G.1010 Parameters [15]

Parameters	Data
One Way Delay	Preferred < 15 s, Acceptable < 60 s
Jitter	N.A.
Throughput	N.A
Packet Loss	0%

Packet Loss is defined as a failure to transmit data packets to their destination. Here is equation 1 to calculate packet loss

$$Packet\ Loss = \frac{(Packet\ sent - Packet\ Received)}{Packets\ sent} \times 100\ \% \quad (1)$$

Delay is the time it takes for a packet to process data transmission from sender to receiver. Throughput is the actual ability of a network to transmit data, it can be called bandwidth in actual conditions. Here is equation 2 for calculating throughput.

$$Throughput = \frac{Amount\ of\ data\ sent}{Data\ Delivery\ Time} \quad (2)$$

Jitter is the variation in delay from a packet to the receiver with the expected time. Here is equation 3 to calculate jitter.

$$J(i) = \frac{J(i-1) + (|D(i-1,i)| - J(i-1))}{16} \quad (3)$$

3. RESULT AND DISCUSSION

In the SDWN performance scenario, it evaluated with the mobility scenario in the user movement model based on Random Direction by generating traffic in the form of data on each test to see the relationship with the services generated. The results of the scenario testing are as follow.

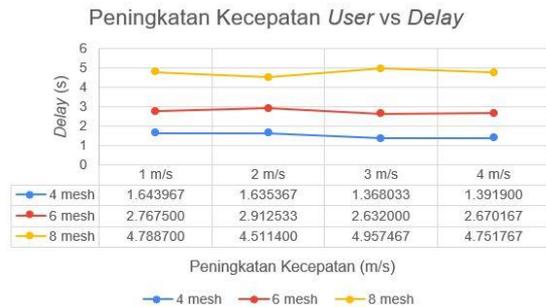


Figure 6. Delay using Dijkstra Algorithm

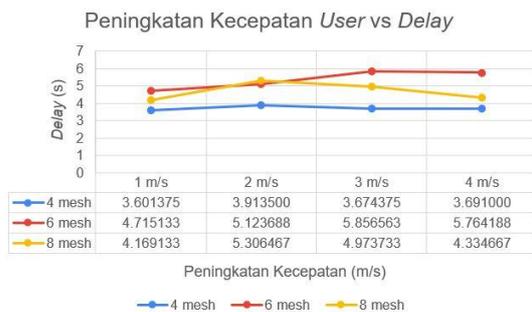


Figure 7. Delay does not use an Dijkstra Algorithm

In Figure 6. Parameter for the delay in data services, the smallest values is 1.36 ms on the mesh 4 access point topology with a speed of 3 m/s, the smallest values, while the highest values is 4.9 ms in 8 mesh topology with a speed of 3 m/s. This happens because when determining the data flow, mesh 8 has more access points (AP) compared to other topologies. So that the longer the process and time for sending packages from client to server.

The value of delay without the Dijkstra algorithm shows a greater value than using the Dijkstra algorithm. This can be seen in Figure 6 and Figure 7. In the mesh topology 4 AP, the delay value when using the smallest dijkstra algorithm is 1.36 ms at a speed of 3 m/s while when without using the dijkstra algorithm the smallest delay value is 3.60 ms at a speed of 1 m/s. In 6 AP mesh topology using the Dijkstra algorithm the smallest value is 2.63 ms at a speed of 3 m/s while at the time without using the dijkstra the smallest value is 4.71 ms at a speed of 1 m/s. In 8 AP mesh topology, the value of delay when using the dijkstra algorithm has the smallest value of 4.51 ms at a speed of 2 m/s, while when without using the dijkstra algorithm the smallest value is 4.16 ms at a speed of 1 m/s. So that when added the dijkstra algorithm is able to reduce the

delay on SDWN. Also, the delay values in all topologies still meet the categories recommended by the ITU-T G1010.

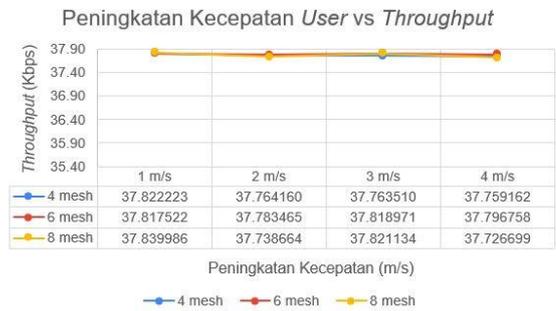


Figure 8. Throughput using Dijkstra Algorithm



Figure 9. Throughput does not use an Dijkstra Algorithm

In Figure 8. Throughput for services in form of data traffic has the lowest values of 37.726 Kbps, namely the 8 AP mesh topology at a user speed of 4 m/s. The highest throughput value is the 8 AP mesh topology with 37.839 Kbps. Overall, the highest average values according to the movement model is mesh 4 AP at 37.777 Kbps, mesh 6 AP at 37.804 Kbps and mesh 8 AP at 37.781 Kbps. The throughput values for data services is relatively stable at 37 Kbps due to the influence of movement from 1 m/s, 2 m/s, 3 m/s and 4 m/s so that it still guarantees the feasibility of QoS in the service generated.

The throughput value without the Dijkstra algorithm shows a smaller value than using the Dijkstra algorithm. This can be seen in Figure 8 and Figure 9. In the mesh topology 4 AP, the highest throughput value when using the Dijkstra algorithm is 37.82 Kbps at a speed of 1 m/s, whereas when without using the dijkstra algorithm, the largest throughput value is 37.14 Kbps at a speed of 1 m/s. In the 6 AP mesh topology using the Dijkstra algorithm the largest value is 37.818 Kbps at a speed of 3 m/s, while at the time without using dijkstra the largest value is 37.24 Kbps at a speed of 2 m/s. In the 8 AP mesh topology, the throughput value when using the dijkstra algorithm has the largest value of 37.83 Kbps at 1 m/s speed, while when without using the dijkstra algorithm the largest value is 36.91 Kbps at 4 m/s. so that when added the Dijkstra algorithm is able to increase the throughput on SDWN

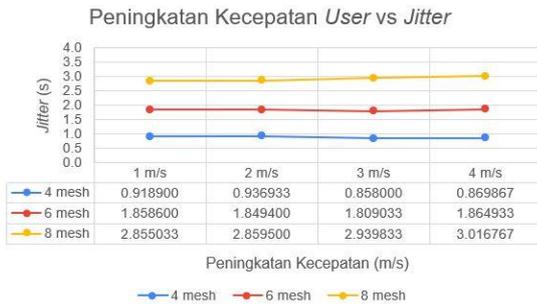


Figure 10. Jitter using Dijkstra Algorithm



Figure 12. Packet Loss using Dijkstra Algorithm

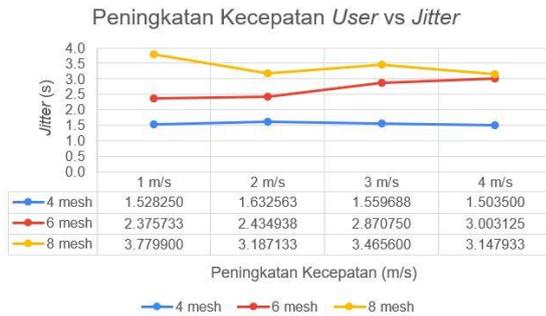


Figure 11. Jitter does not use an Dijkstra Algorithm

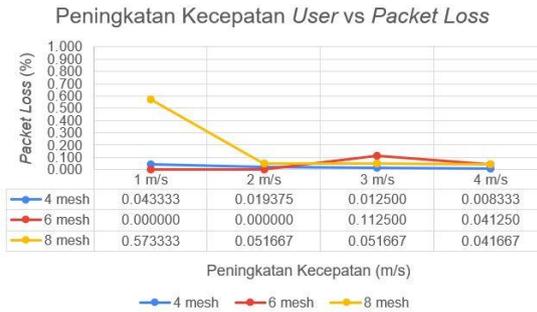


Figure 13. Packet Loss does not use an Dijkstra Algorithm

In Figure 10. Jitter or delay variation in data services obtained the highest values with a value of 3.016 ms at a speed of 4 m/s in 8 AP mesh topology. Whereas, the lowest value occurs in the 4 Access Point AP mesh topology with a value of 0.85 ms at a speed of 3 m/s at. Of the three topologies, the Random Direction movement model shows a stable value. So that it still guarantees the feasibility of QoS in services that are generated.

The value of jitter without the Dijkstra algorithm shows a smaller value than using the Dijkstra algorithm. This can be seen in Figure 10 and Figure 11. In the mesh topology 4 AP the jitter value when using the smallest dijkstra algorithm is 0.85 ms at a speed of 3 m / s while when without using the dijkstra algorithm the smallest jitter value is 1.50 ms at a speed of 4 m / s. In the 6 AP mesh topology using the Dijkstra algorithm the smallest value is 1.80 ms at a speed of 3 m / s while at the time without using the dijkstra the smallest value is 2.37 ms at a speed of 1 m / s. In the 8 AP mesh topology, the jitter value when using the dijkstra algorithm has the smallest value of 2.855 ms at a speed of 1 m / s while when without using the JKSTRA algorithm the smallest value is 3.14 ms at a speed of 4 m / s. So that when added the dijkstra algorithm is able to reduce jitter in SDWN.

In Figure 12. The packet loss on the service is in the form of data with a loss value of 0% in all topological models with a speed of 1 m / s to 4 m/s. Means that during data transmission no packet is lost. So that the value of losses that occur at each traffic speed shows a small value below 0.1%. So, it's in great shape based on the ITU-T G1010 standard.

The value of packet loss without the Dijkstra algorithm shows a smaller value than using the Dijkstra algorithm. This can be seen in Figure 12 and Figure 13. In the mesh topology 4 AP the value of packet loss when using the smallest Dijkstra algorithm is 0% at all speeds, while without using the Dijkstra algorithm the smallest packet loss value is 0.008% at a speed of 4 m / s. on 6 AP mesh topology using the Dijkstra algorithm the smallest value is 0% at all speeds. while at the time without using the Jkstra the smallest value is 0% at a speed of 1 m / s and 2 m / s. In the 8 AP mesh topology the packet loss value when using the dijkstra algorithm has the smallest value of 0% at all speeds, whereas when without using the Dijkstra algorithm the smallest value is 0.04% at a speed of 4 m / s. So that when added the dijkstra algorithm is able to reduce packet loss in SDWN.

From the results of the research conducted, the dijkstra Algorithm on SDWN is able to improve performance by reducing delay, increasing throughput, decreasing jitter and packet loss. So that the addition of the Dijkstra algorithm as a way to solve the problem of selecting the path when sending packets in the form of data can be a solution. In addition, the addition of the Dijkstra algorithm and the application of SDN to wireless can increase wireless capabilities when the complexity of the wireless network increases.

4. CONCLUSION

SDWN virtual network performance using the Dijkstra algorithm can improve performance by having stable Delay and Jitter values. The delay obtained in this study can be minimized to 1.3 ms and meets ITU-T recommendation standards, as well

as for relatively stable throughput values at 37 Kbps with the influence of movement from 1 m/s, 2 m/s, 3 m/s, and 4 m/s, so that it still guarantees the feasibility of QoS in the service raised and packet loss has a value below 0.1% which meets the ITU-T standard parameters and is in a good category. The IEEE 802.11n standard is capable of being applied to SDN-based wireless networks. Therefore, the mobility of users moving within the range of SDWN-based access points using the Dijkstra algorithm can guarantee the availability of types of services such as data from speeds of 1 m/s to 4 m/s. Future research is expected to be able to apply load balancing to SDWN using the round robin algorithm, and others.

DAFTAR PUSTAKA

- [1] K. Pahlavan and P. Krishnamurthy, "Evolution and Impact of Wi-Fi Technology and Applications: A Historical Perspective," *Int. J. Wirel. Inf. Networks*, no. 0123456789, 2020, doi: 10.1007/s10776-020-00501-8.
- [2] M. J. Abbas, "Interoperability Framework for Wireless Standards-Performance Analysis," *IEEE Int. Conf. 2018 Recent Adv. Eng. Technol. Comput. Sci. RAETCS 2018*, pp. 1–5, 2018, doi: 10.1109/RAETCS.2018.8443818.
- [3] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "Impact of IEEE 802.11n/ac PHY/MAC high throughput enhancements over transport/application layer protocols - A survey," *arXiv*, vol. 19, no. 4, pp. 2050–2091, 2017.
- [4] A. Lopez-Raventos, F. Wilhelmi, S. Barrachina-Munoz, and B. Bellalta, "Combining software defined networks and machine learning to enable self organizing w lans," *Int. Conf. Wirel. Mob. Comput. Netw. Commun.*, vol. 2019–Octob, no. 19, pp. 167–174, 2019, doi: 10.1109/WiMOB.2019.8923569.
- [5] I. Technology and S. Arabia, "On Software-De fi ned Wireless Network (SDWN) Network Virtualization: Challenges and Open Issues," no. January, pp. 1510–1519, 2018.
- [6] I. Attamimi, W. Yahya, and M. Hanfi, Hannats, "Analisis Perbandingan Algoritma Floyd-Warshall dan Dijkstra untuk Menentukan Jalur Terpendek Pada Jaringan Openflow," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1842–1849, 2017.
- [7] B. Anggita Linuwih, A. Virgono, and B. Irawan, "Design and Analysis Software Defined Networking for Lan Network: Application," *e-Proceeding Eng.*, vol. 3, no. 1, pp. 749–756, 2016.
- [8] A. Yusup, R. M. Negara, and D. D. Sanjoyo, "Analisis Implementasi Software-Defined Wireless Network (Sdwn) Menggunakan Mininet-Wifi Analysis of Software-Defined Wireless Network (Sdwn) Implementation Using Mininet-Wifi," Bandung, 2019.
- [9] M. S. Annas and D. Maulana, "Perancangan Audio Streaming Menggunakan Wifi Berbasis Mikrokontroler ATMega 328," vol. 1, no. 1, pp. 27–32, 2019.
- [10] S. W. G. Abusalim, R. Ibrahim, M. Zainuri Saringat, S. Jamel, and J. Abdul Wahab, "Comparative Analysis between Dijkstra and Bellman-Ford Algorithms in Shortest Path Optimization," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 917, no. 1, 2020, doi: 10.1088/1757-899X/917/1/012077.
- [11] S. Waleed, M. Faizan, M. Iqbal, and M. I. Anis, "Demonstration of single link failure recovery using Bellman Ford and Dijkstra algorithm in SDN," *ICIEECT 2017 - Int. Conf. Innov. Electr. Eng. Comput. Technol. 2017, Proc.*, pp. 0–3, 2017, doi: 10.1109/ICIEECT.2017.7916533.
- [12] S. Avallone, D. Emma, A. Pescapè, and G. Ventre, "A practical demonstration of network traffic generation," in *Proceedings of the Eighth IASTED International Conference on Internet and Multimedia Systems and Applications*, 2004, pp. 138–143.
- [13] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wirel. Commun. Mob. Comput.*, vol. 2, no. 5, pp. 483–502, 2002, doi: 10.1002/wcm.72.
- [14] D. Priadi, A. Muzakhim, N. Suharto, J. T. Digital, J. T. Elektro, and P. N. Malang, "Pengukuran Quality of Service (QoS) Pada Aplikasi File Sharing Dengan Metode Client-Server Berbasis Android," *J. JARTEL*, vol. 6, no. 1, pp. 39–49, 2018.
- [15] ITU-T, "End-user Multimedia QoS Categories," vol. 1010, 2001.