Jurnal Mantik
Volume 4, Number 3, Nov 2020, pp. 1546-1550 E-ISSN 2685-4236
https://iocscience.org/ejournal/index.php/mantik

Fail Path Analysis on Openflow Network Using Floyd-Warshall Algorithm.

Ebim Fatur Rohman[1], Ritzkal[2], Yuggo Afrianto[3]

[1,2,3]Technical Information, Faculty Of Engineering And Sanis, Ibn Khaldun University Bogor,Indonesia
Jl. Sholeh Iskandar, Kedungbadak, Kec. Tanah Sereal, Kota Bogor, Jawa Barat16162

E-mail: syncbim@gmail.com, ritzkal@ft.uika-bogor.ac.id, yuggo@uika-bogor.ac.id.

ARTICLE INFO

Article history:
Received: 12/07/2020
Revised: 22/08/2020
Accepted: 30/09/2020

Keywords:
openflow, algoritma *floyd-warshall*, *fail-path*.

ABSTRACT

Software Defined Network (SDN) is a new concept used to solve network problems by separating the control plane and data plane in a different device. Communication between the control plane and the data plane uses the Openflow scenario. SDN has several capabilities in many network technology methods and has been implemented, among others, for routing mechanisms. In routing, there are several problems including fail path. When a fail path occurs, the system will look for another shortest path using the Floyd-Warshall algorithm. The floyd-warshall algorithm will be implemented using the POX controller. The implementation uses a mesh topology on the mininet emulator and pox as a controller. This research method includes analysis consisting of preparation of hardware and software requirements, and analysis of how the system works, design consisting of mesh topology, Floyd-warshall algorithm flowchart, propagation testing, throughput and latency testing, implementation consisting of operating system installation, Mininet , POX, Applying Mesh topology to the Mininet Emulator, and Applying the Floyd-warshall Algorithm to the POX Controller, manually calculating the Floyd-Warshall algorithm on the mesh topology and testing which consists of testing propagation, throughput and latency. Furthermore, testing with 3 tests is propagation , throughput and latency. In testing the throughput and latency there are 3 scenarios.

1. Introduction

The development of network technology that is currently running is still considered a static and conventional network, therefore the concept of dynamic network that is programmable has developed. Software Defined Network (SDN) is an example of the development of a network [2]. SDN has several technological capabilities including load balancing, multimedia multicast, intrusion detection, and routing. Routing is one that is used to support the data transfer process so that the communication process can run successfully. In routing there are problems such as link failure (fail path). When a fail path occurs it causes the system to look for another path, and if no alternative path is found, it will result in communication failure. In the process of finding another path in the SDN concept, a controller monitors the status of each port of each switch. In routing, an algorithm for finding the shortest path is usually used, one of which is the Floyd-Warshall algorithm [3]. This algorithm uses all pair shortest path search, which means that the shortest path search can be determined between all pairs of vertices [4].

Based on these problems, this study aims to analyze the fail path using the Floyd-Warshall algorithm on the SDN network using the Openflow protocol. In fail path testing is done by terminating the path between switches using the down link. When a fail path occurs, the path cannot be passed and the system automatically looks for an alternative path [3]. The process of finding the shortest path when a fail path occurs is carried out to produce another shortest path, namely by looking at the results of the comparison between the closest distance before and after the fail path occurs. In solving this problem, it is simulated with Mininet software. Mininet is a program to create a realistic virtual network in a single machine (virtual machine) with one command [1]. The formulation of the problem of this research are (i) How to apply the mesh topology to the mininet emulator and Floyd-warshall algorithm on the pox controller? (ii) How do I get the results of the tracking performance using parameters of propagation, throughput and latency? The objectives of this research are (i) Obtain the results of the implementation of the application of the mesh topology on the mininet emulator and the Floyd-warshall algorithm on the controller pox. (ii) Obtain the results of the tracking performance using parameters of propagation, throughput and latency.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

2. Research Methods

The research method or framework in this study consists of four stages, namely analysis, design, implementation and testing.

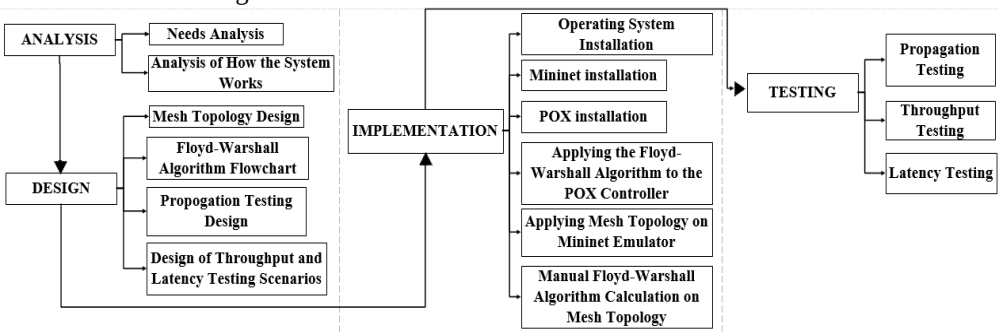

Fig 1. Research Methods

3. Result

2.1 Analysis

In the process of analyzing how it works, it will explain how the system works in this study. The following image will explain the analysis of how this system works:

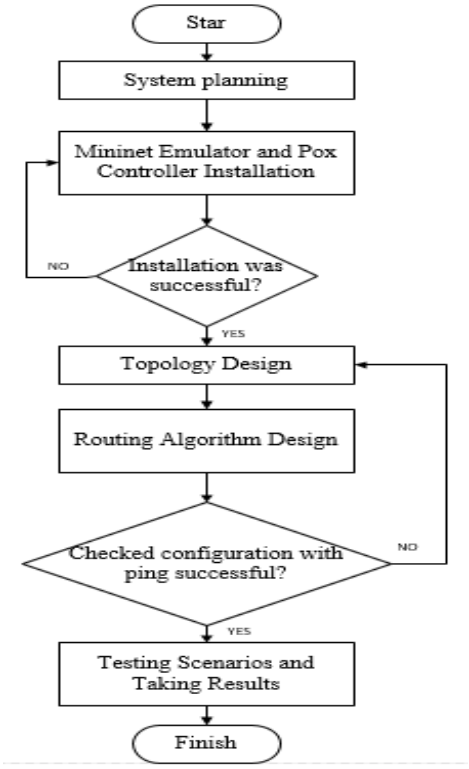

Fig 2. Flowchart System

2.2 Design

This research discusses the mesh topology and flowchart of the Floyd-warshall algorithm which is used to facilitate and understand the concept of the Openflow network to be analyzed.

a) Mesh Topology Design.

The result of the design carried out in this study is a Mesh network topology which is shown in Fig. 3

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

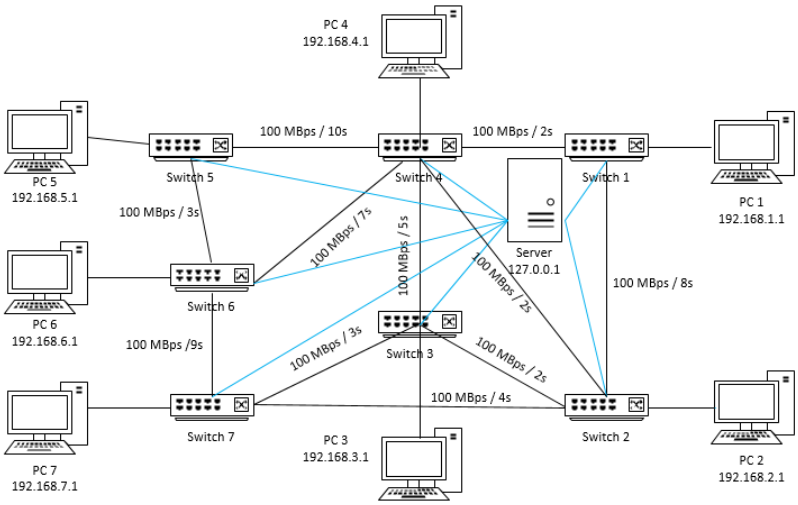

Fig 3. Mesh Topology

The topology results are made by applying the Floyd-warshall algorithm to the Openflow network. The topology used is a mesh topology using 7 PCs, 7 switches and 1 server. The black line connects between a switch and a PC or between a switch and the blue line connects between the server and the switch. The mesh topology will be applied to the mininet emulator. The mesh topology on the Openflow network describes the IP addressing of the computer network structure on the Openflow server, where PC 1 with IP address 192.168.1.1, PC 2 with IP address 192.168.2.1, PC 3 with IP address 192.168.3.1, PC 4 with IP address 192.168.4.1, PC 5 with the ip address 192.168.5.1, switch 6 with the ip address 192.168.6.1, and PC 7 with the ip address 192.168.7.1. Link Switch 1 goes to switch 2 with a delay of 8 and a bandwidth of 100, link switch 1 to switch 4 with delay 2 and bandwidth of 100, link switch 2 to switch 3 with delay 2 and bandwidth of 100, link switch 2 to switch 4 with delay 2 and bandwidth 100, link switch 2 to switch 7 with delay 4 and bandwidth of 100, link switch 3 to switch 4 with delay 5 and bandwidth of 100, link switch 3 to switch 7 with delay 3 and bandwidth of 100, link switch 4 to switch 5 with delay 10 and bandwidth 100, link switch 4 to switch 6 with delay 7 and bandwidth of 100, link switch 5 to switch 6 with delay 3 and bandwidth 100, link switch 6 to switch 7 with delay 9 and bandwidth 100.

b) Floyd-warshall Algorithm Flowchart Design.

The results of the Floyd-Warshall algorithm flowchart design shown in Figure 5 explain the shortest route search flow using the Floyd-Warshall algorithm for route searches. In the first process carried out is to input the origin and destination locations as initial data, in iteration 1, each matrix cell will be checked whether the distance between the two original points is greater than the sum between the origin and destination points to check whether the distance between the two origin point is greater than the sum of the origin (origin = 1st iteration) to the destination point. In other words, whether W [i, j]> W [i, k] + W [k, j], if so then the distance between two points is initially replaced by the sum of the distance from the origin to the destination point (destination point = iteration 1 with the distance between the origin (origin = 1st iteration) to the destination point W [i, k] + W [k, j]. If not, then the distance between two original points W [i is used. , j). Furthermore, the iteration process is carried out until the last iteration (number of iterations = total number of points).

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

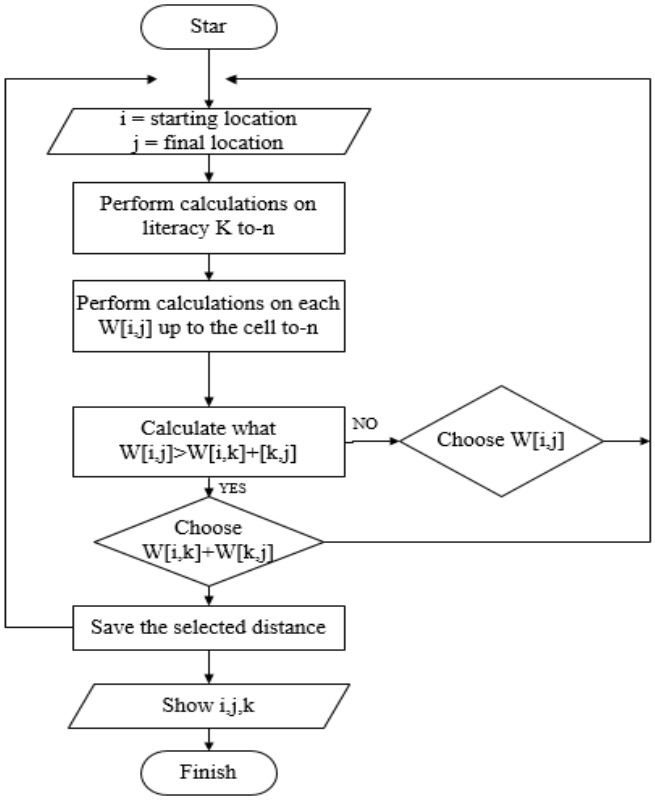

Fig 4. *Flowchart* Algoritma *Floyd-warshall*

c) Propagation Testing Design.

The results of this design are carried out for testing the propagation of the Floyd-warshall algorithm. This is to find out how long it takes when a fail path occurs in the middle of sending a packet. The black line connects between a switch and a PC or between switches, for a blue line that is connecting between a server and a switch, and if a fail path occurs, the path is marked in red as shown in Figure

2.3 Testing

a) Propagation Testing.

Based on the table of results from propagation testing on test 1 obtained on sending PC 1 packages to PC 7 with the occurrence of a fail path on the 7th packet delivery, which takes 9 seconds to get a new path, then test 2 is obtained on sending PC 2 packages to PC 7 with the fail path on the 7th packet sending, which takes 6 seconds to get a new path, then the 3rd test obtained on sending the PC 3 package to PC 7 with the occurrence of the fail path on the 7th packet delivery which takes 14 seconds to get the path new, then the test 4 obtained on sending PC 4 packages to PC 7 with the occurrence of a fail path on the 7th packet delivery, which takes 9 seconds to get a new path. Testing 5 is obtained on sending PC 5 packages to PC 7 with the fail path on the 6th packet sending, which takes 12 seconds to get a new path. Test 6 obtained on sending PC 6 packets to PC 7 with the occurrence of a fail path on the 5th packet delivery, which takes 13 seconds to get a new path. It can be seen that the floyd-warshall algorithm in finding a new path when a fail path occurs takes more than 5 seconds to find a new path when a fail path occurs.

Table 1
Propagation Testing.

Propagation Testing.	Testing					
	1	2	3	4	5	6
Result(sec)	9	6	14	9	12	13

b) Throughput Testing

Based on the table of results from throughput testing in scenario 1 with an average of 11.24 Mbytes / sec, scenario 2 with an average of 10.88 Mbytes / sec and scenario 3 with an average of 10.26 Mbytes / sec. It can be seen that there is a decrease in throughput when there is a fail path compared to when there is no fail path.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Table 2
Throughput Testing.

Throughput Testing	Testing (Mbytes/sec)						Avg (Mbytes/sec)
	1	2	3	4	5	6	
Scenario 1	11,62	10,74	11,64	10,62	11,28	11,52	11,24
Scenario 2	11,4	10,58	10,08	10,84	11,42	10,96	10,88
Scenario 3	7,18	10,22	11,52	9,68	11,48	11,48	10,26

c) Latency Testing

Based on the table of results from the latency test in scenario 1 with an average of 0.075 ms, scenario 2 with an average of 0.123 ms and scenario 3 with an average of 0.131 ms. It can be seen that it takes less time before the fail path occurs than it does after the fail path which takes a long time.

Table 3
Latency Testing.

Latency Testing	Testing (ms)						Avg (ms)
	1	2	3	4	5	6	
Skenario 1	0,098	0,070	0,061	0,075	0,075	0,070	0,075
Skenario 2	0,094	0,139	0,172	0,095	0,113	0,124	0,123
Skenario 3	0,135	0,143	0,116	0,112	0,129	0,150	0,131

4. Conclusion

Based on the formulation of the problem raised and the system has gone through the design, implementation and testing stages, the following conclusions can be drawn: (1) The results of the implementation of applying a mesh topology using the Mininet emulator are used as testing parameters of propagation, throughput and latency. Furthermore, applying the Floyd-Warshall algorithm to the controller pox is used to find the shortest path when a fail path occurs. (2) The results of the test proving on the application of the Floyd-Warshall algorithm to the propagation parameters in test 1 took 9 seconds to get a new path, in test 2 it took 6 seconds to get a new path, in test 3 it took 14 seconds to get a path new, in test 4 it took 9 seconds to get a new line, in test 5 it took 12 seconds to get a new line, in test 6 it took 12 seconds to get a new line. It can be seen that the Floyd-warshall algorithm in finding a new path when a fail path occurs takes more than 5 seconds to find a new path when a fail path occurs. Furthermore, throughput testing in scenario 1 is 11.24 Mbytes / sec, scenario 2 with Average 10.88 Mbytes / sec and scenario 3 with an average 10.26 Mbytes / sec. It can be seen that there is a decrease when there is a fail path compared to when there is no fail path. Furthermore, testing the latency in scenario 1 with an average of 0.075 ms, scenario 2 with an average of 0.123 ms and scenario 3 with an average of 0.131 ms. It can be seen that it takes less time before the fail path occurs than it does after the fail path which takes a long time.

5. References

[1] R. A. Maha, L. O. Sari, and E. Safrianti, "Simulasi dan Pemodelan Software Defined Network (SDN) untuk Manajemen Jaringan Data UR," Jom FTEKNIK, vol. 5, no. 2, pp. 1–6, 2018, doi: 10.1017/CBO9781107415324.004.

[2] I. A. Saputra, R. R. M., and S. N. Hertiana, "Uji Performansi Algoritma Floyd-Warshall Pada Jaringan Software Defined Network (SDN)," J. Elektron. dan Telekomun., vol. 16, no. 2, p. 52, 2016, doi: 10.14203/jet.v16.52 58.

[3] E. P. Aprilianingsih, R. Primananda, and A. Suharsono, "Analisis Fail Path Pada Arsitektur Software Defined Network Menggunakan Dijkstra Algorithm," J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 1, no. 3, pp. 174–183, 2017, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/59.

[4] 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 Komputer., vol. 1, no. 12, pp. 1842–1849, 2017.

[5] S. Dwinson, "Implementasi Load-Balancing Dengan Metode Round Robin Dalam Software Defined Networking (SDN) Menggunakan Controller POX," 2015.

[6] Ritzkal. 2020. "Tick Waste Application in Houses With Warning of Microcontroller Assistant Social Media.," Jurnal MANTIK Vol 3, hlm. 559-568.

[7] Ritzkal. 2018. "Manajemen jaringan untuk pemula.," Bogor: UIKA PRESS

[8] M Subchan, Dedi Setiadi. 2020. "Information System For Sale Of Muslim Clothes Based On E-Commerce Technology.," Jurnal MANTIK Vol 4, hlm. 311-318.