ONOS Intent Path Forwarding using Dijkstra Algorithm

Taufik Irfan[1], Rifqy Hakimi [2], Aris C. Risdianto [3], Eueung Mulyana[4]
School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia[1,2,4]
School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Korea[3]
Email: taufik.irfan@students.itb.ac.id[1]*, rifqyhakimi@stei.itb.ac.id*[2]*, aris@nm.gist.ac.kr*[3]*, eueung@stei.itb.ac.id*[4]

***Abstract* – The use of intent on computer networking is a new concept that can be used in Software-defined Networking (SDN) technology, where the purpose is to translate high-level policies into the network configuration. ONOS Intent Path Forwarding (ONOS-IPF) is the application designed to implements the concept of the use of the intent on Software-defined Networking (SDN) with an Open Network Operating System (ONOS) as its controller.**

With a policy to have high availability and less latency than ONOS default application packet forwarding, this ONOS-IPF application will create a packet forwarding based on bandwidth capacity and network traffic utilization rather than use hop count only. In order to achieve its goal of applying policy into a form of network configuration, the utilization of the Dijkstra shortest path algorithm as the main method is needed in this application. As a result, the end-to-end path latency created by for this application is smallest than default ONOS forwarding application when it used to deliver packet size is more than 3200 bytes. This ONOS-IPF application also a high availability with time convergence around 2 seconds and have a secondary path as a backup for the primary path.

Keywords—Intent, ONOS, Dijkstra

I. INTRODUCTION

Nowadays the technology of networking has very rapid development. Many concepts of new technology are offered as a form of business solution. The use of intent in network programming is some of the innovation in network technology where the main purpose is the transformation from policy into network configuration in the form of intent. Intents are used as an abstraction of business purpose in network service provider companies. These companies will make own policies according to their needs in order to meet the goals of the business itself.

In this paper, the author will conduct research about the use of intent with the implementation of an Off-platform Application (OPA) called ONOS Intent Path Forwarding (ONOS-IPF) in Software-defined Networking (SDN) framework. The main purpose of this application is to implement the concept of using the intent that can transform the policy of user to getting a best path packet forwarding from source to destination into network device configuration. Although in ONOS application there are already have an application for packet forwarding, that application is using shortest hop-count as best-path calculation. So the purpose of this research is not only to create the best path from source to destination, but also the best path must use the smallest bandwidth value on the link rather than using the shortest hop. In order for this purpose to be carried out, the Dijkstra algorithm is needed as the algorithm used to perform the routing and forwarding process.

This paper contains a literature review, application design, application simulation, and result, and the last is a conclusion and future work. In a literature review, it tells about software-defined networking, ONOS application, and some paper related to this research. While in application design, it tell about the algorithm used in this application. And the last is in application simulation, in this part, there is 2 section which is the simulation using Abilene network topology and the measurement.

II. LITERATURE REVIEW

A. Software-defined Networking

Software-defined Networking (SDN) is a computer networking technology that simplifies the network controlling process with separation of control-planes from each device so every device only has a data-plane while for network control and policies are centrally controlled by software-based controllers [12]. Even though the control plane is separate from the device but the control process can still run well, this is because of the OpenFlow protocol

OpenFlow protocol is a communication protocol used by the SDN controller to network devices. OpenFlow works by interacting with the flow table of each device. It makes a flow for each action or policy on the device in handling each data or packet such as for example forwarding or dropping.

B. Open Network Operating System (ONOS)

The Open Network Operating System (ONOS) is a software-based controller in SDN that offers capabilities in intent and API technology. Intent is basically a model object that cannot be changed which is also a form of abstraction from application requests to ONOS to change network behavior [17]. One of the advantages of the ONOS feature is the convenience to users in the installation of intent. On ONOS there are various features of installing intent using the REST API, as follows [4]:

1. Host to Host: installation of an intent that is used for direct communication between two hosts.
2. Point to Point: installation of intent that is used for communication between two interfaces of the device.
3. Single to Multi Point: installation of an intent used for communication between one to many interfaces of the devices.
4. Multi to Single Point: installation of intent that is used for communication between many to one interface on the device.

C. Dijkstra Shortest Path Algorithm

Dijkstra algorithm is an algorithm that can be used in finding the best path between two nodes. By using a topology graph theory where G = (V, E), V is a representation of vertices/nodes and E is the edge or link connecting each node. For each edge, it can be written as E (u, v), where u is the node source, while v is the destination node.

```
for all v ϵ V–{s}
      do dist [v] <--- ∞
S <--- {}
Q <-- V
while Q not empty
do u <--- minimum dist[Q]
  S <-- S U {u}
    for all v ϵ neighbors [u]
      do if dist[v] > dist [v] + w(u,v)
          then d[v] <-- d[u] + w (u,v)
```

Fig. 1. Pseudocode Dijkstra algorithm

In the Dijkstra algorithm like in Figure 1, Dijkstra will visit each of the other nodes adjacent with the lowest cost of the edge first to find the destination node. During the visiting process for each node that has been visited, it will be called the visited node, while for the edge (u, v) that has been passed, it will be called edge relaxation. When an edge (u, v) has become edge relaxation, there will be a path formed between the source nodes (node u) to the destination node (node v), and a value that is an accumulation of the cost of that path. The process of this search algorithm will be completed when all nodes have been visited [14].

D. Related Works

There is plenty of research about ONOS, intent, and Dijkstra shortest path algorithm, but some of them only use ONOS and intent without Dijkstra shortest path algorithm like in this paper "Intent-Based Path Selection for VM Migration Application with Open Network Operating System" [1]. This paper research's regarding the use of intent in ONOS application for creating two different paths which are the first path is path using reactive forwarding and the second path is proactive forwarding based on ONOS API. Both of path is a purpose for Virtual Machine (VM) migration.

Besides that paper, there is also a paper about ONOS and Dijkstra shortest path algorithm with the title "Implementasi Routing Berbasis Algoritme Dijkstra Pada Software Defined Networking Menggunakan Kontroler Open Network Operating System". In this paper, the research is about creating the shortest path from source to the destination node using Dijkstra shortest path algorithm in ONOS application. In this paper instead of using link bandwidth as link cost of a weighted graph, it using Geo Distance location as a cost of a weighted graph. Also the different from ONOS-IPF application is, in this paper as Northbound (NBI) abstraction in ONOS is using CLI instead of REST API like ONOS-IPF application. Due to its use a Geo Distance location, the result from this paper application is based on node location mapping.

III. Application Design

ONOS-IPF application is an off-platform application created with the purpose of using intent as a form of abstraction from the policy and control of a network forwarding. In order this ONOS-IPF application capable to work as a translation for policy abstraction into a network configuration there is four main steps, as follows:

1. Use of the REST-API interface in building communication for retrieving information data from the ONOS Core, which is used as the main variables of this ONOS-IPF application.
2. The formation of a Weighted Graph Topology based on variables that have been obtained from method 1 as a method for translating the physical form of a network topology into a logical form that will be used in calculating the best path.
3. Calculating and mapping of the primary and backup best paths using Dijkstra's algorithm on network topology
4. Translating from the results of the previous best path calculation in method 3 into an intent that will then be installed in the intent manager on the ONOS software-based controller via the REST-API interface.

IV. Application Simulation

For the topology that will be used in the simulation will use the Abilene topology 2003 like in Figure 2.

Fig. 2. Abilene Network Topology 2003 [16]

TABLE 1. ABILENE TOPOLOGY VERTICES / NODES

City Name	Switch Name
New York	s1
Chicago	s2
Washington DC	s3
Seattle	s4
Sunnyvale	s5
Los Angeles	s6
Denver	s7
Kansas City	s8
Houston	s9
Atlanta	s10
Indianapolis	s11

In order to create a network testbed environment by Mininet, node and edge information is need. By the network topology in Figure 3, nodes information can be gathered as follows in Table 1. it is also necessary to adjust the bandwidth value for each link and design the weight value for each link as the link cost value for Dijkstra's algorithm calculation.

To design the link cost value for each link, the ONOS-IPF application uses a link cost formula of the Open Shortest Path

First (OSPF) routing protocol with a value for the reference bandwidth of 10,000,000,000 bits as follows as Equation 1.

OSPF Cost Formula:

$$Cost = \frac{\boldsymbol{Reference\ BW}}{\boldsymbol{Interface\ Bw}}\ (\boldsymbol{bps}) \qquad (1)$$

default Reference BW= 10^{10}

In addition to node information, from Figure 3 can obtain information about the edge and bandwidth for each edge as follows in Table 1. Because from topology there is no information about Host, so bandwidth for every host to switch will use the default value (10.000 Mbps) with link cost will be (1). Also bandwidth limitation in Mininet, therefore every bandwidth in each every link will reduce by the power of 1000. So the edge information can be gathered as follow as in Table 2.

TABLE 2. ABILENE TOPOLOGY EDGE

Node (u)	Node (v)	Topology BW (Gbps)	MININET BW (Mbps)	Link Cost (metric)
s1	s2	9.952	10	1000
s1	s3	9.952	10	1000
s2	s11	9.952	10	1000
s3	s10	2.488	2	5000
s4	s5	9.952	10	1000
s4	s7	9.952	10	1000
s5	s6	9.952	10	1000
s5	s7	2.488	2	5000
s6	s9	9.952	10	1000
s7	s8	2.488	2	5000
s8	s9	9.952	10	1000
s8	s11	9.952	10	1000
s9	s10	2.488	2	5000
s10	s11	2.488	2	5000

Then the following weighted graph for this topology is in Figure 3.

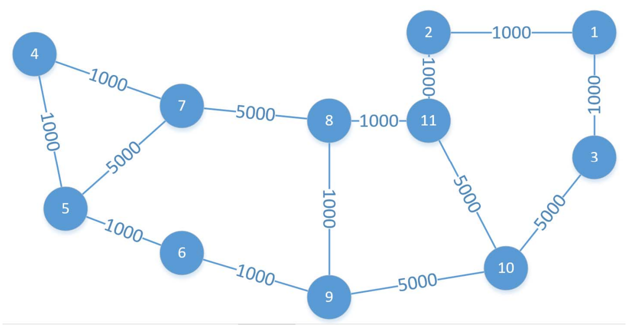
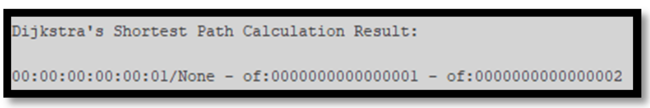


Fig. 3. Weighted Graph (G= (V, E)) Topology.

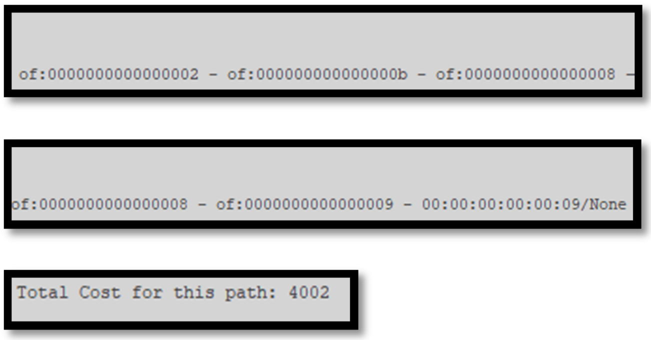
The following are best path calculation for source hosts 1 (New York) to destination hosts 9 (Houston) using manual calculation based on figure 3:

- Best Path : h1 -> s1 -> s2 -> s11 -> s8 -> s9 -> h9
- Total Cost : 1 + (1000 x 4) + 1 = 4002
- Primary path intent calculation from ONOS-IPF:



```
Dijkstra's Shortest Path Calculation Result:

00:00:00:00:00:01/None - of:0000000000000001 - of:0000000000000002
```



```
of:0000000000000002 - of:000000000000000b - of:0000000000000008 -
```

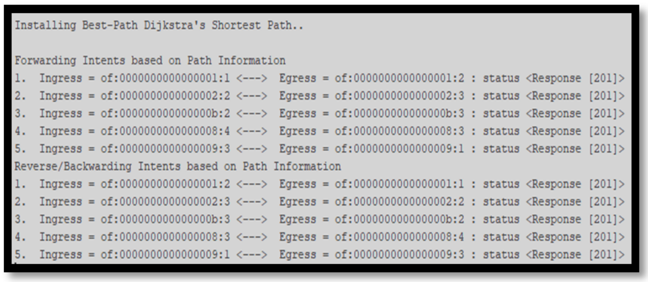
```
of:0000000000000008 - of:0000000000000009 - 00:00:00:00:00:09/None
```

```
Total Cost for this path: 4002
```

Fig. 4. ONOS-IPF primary path calculation result

The result of the best path calculation done by ONOS-IPF application is the same as the expected result from our manual calculations. From the result calculation, this ONOS-IPF application will create intents for every the primary path in both directions.

- Primary path intent installation:

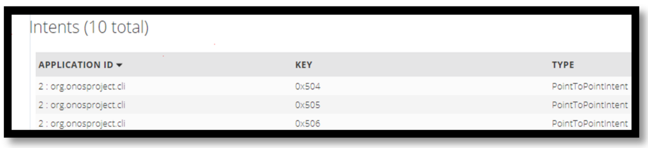


```
Installing Best-Path Dijkstra's Shortest Path..

Forwarding Intents based on Path Information
1.  Ingress = of:0000000000000001:1 <--->  Egress = of:0000000000000001:2 : status <Response [201]>
2.  Ingress = of:0000000000000002:2 <--->  Egress = of:0000000000000002:3 : status <Response [201]>
3.  Ingress = of:000000000000000b:2 <--->  Egress = of:000000000000000b:3 : status <Response [201]>
4.  Ingress = of:0000000000000008:4 <--->  Egress = of:0000000000000008:3 : status <Response [201]>
5.  Ingress = of:0000000000000009:3 <--->  Egress = of:0000000000000009:1 : status <Response [201]>
Reverse/Backwarding Intents based on Path Information
1.  Ingress = of:0000000000000001:2 <--->  Egress = of:0000000000000001:1 : status <Response [201]>
2.  Ingress = of:0000000000000002:3 <--->  Egress = of:0000000000000002:2 : status <Response [201]>
3.  Ingress = of:000000000000000b:3 <--->  Egress = of:000000000000000b:2 : status <Response [201]>
4.  Ingress = of:0000000000000008:3 <--->  Egress = of:0000000000000008:4 : status <Response [201]>
5.  Ingress = of:0000000000000009:1 <--->  Egress = of:0000000000000009:3 : status <Response [201]>
```

Fig. 5. ONOS-IPF intent primary path installation

- Intent verification in ONOS:



Intents (10 total)

APPLICATION ID	KEY	TYPE
2 : org.onosproject.cli	0x504	PointToPointIntent
2 : org.onosproject.cli	0x505	PointToPointIntent
2 : org.onosproject.cli	0x506	PointToPointIntent

Fig. 6. Primary path intent verification in ONOS

After the primary path is already installed, then the calculation of backup link as shown in Figure 7 which is the second form of weighted graph topology without considering the installed primary path.

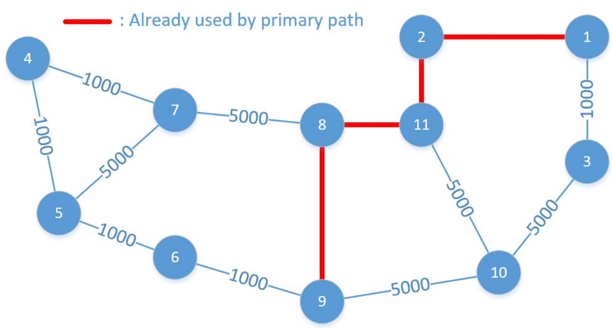


Fig. 7. Weighted Graph (G= (V, E)) Topology for backup path.

- Best Path : h1 -> s1 -> s3 -> s10 -> s9 -> h9
- Total Cost : 1 + 1000 + (5000 x 2) + 1 = 11002
- Backup path intent calculation from ONOS-IPF:

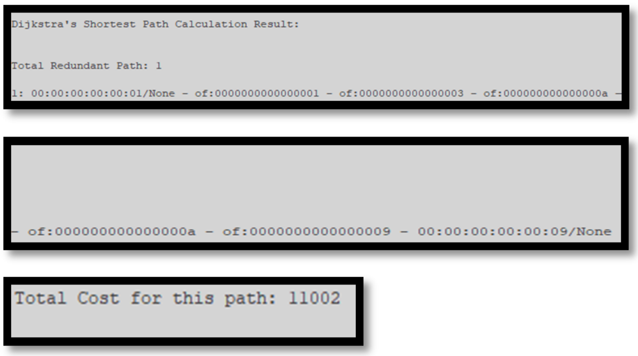


Fig. 8. ONOS-IPF backup path calculation result

Same as the primary path, the result of backup path calculation will be translated in the form of an intent by ONOS-IPF application by using ONOS controller intent framework.

• Backup path intent installation:

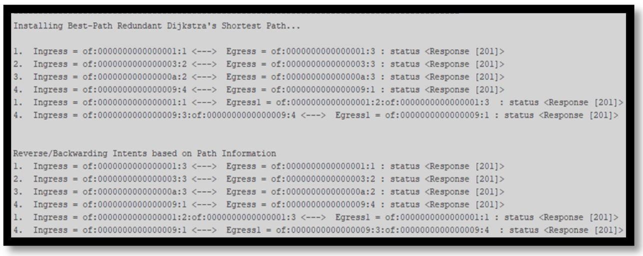


Fig. 9. ONOS-IPF intent backup path installation

• Intent verification in ONOS:

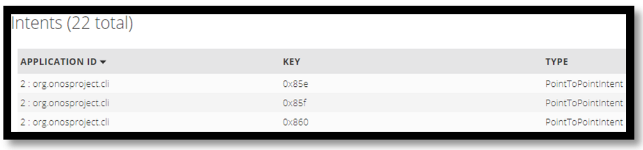


Fig. 10. Primary and backup path intent verification in ONOS

As the result from both installed primary path and backup path, now from New York City as represented by Host 1 can communicate with Houston City as represented by Host 9 with multipath like in Figure 11.

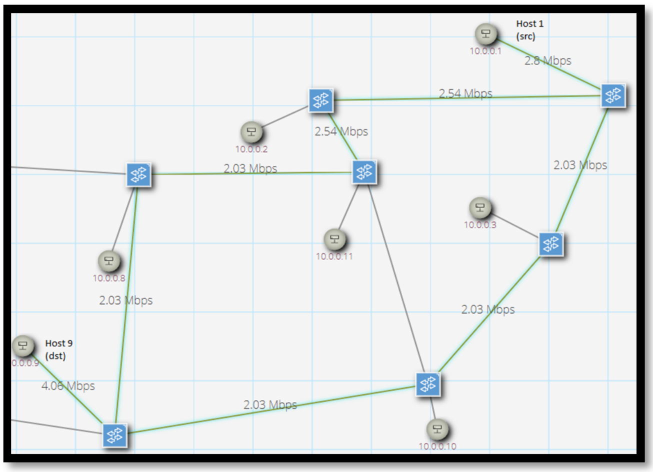


Fig. 11. Packet forwarding from Host 1 to Host 9

By using the source node and destination node from Table 3, performance testing for latency will be carried out in 2 stages. For the first stage of testing, packet sending will be carried out with the minimum packet size. The purpose of using this minimum packet size is to be able to see the different paths used by the ONOS default forwarding application (onos-app-fwd) with ONOS-IPF Application.

TABLE 3. ABILENE TOPOLOGY VERTICES / NODES

Node		ONOS Default App		ONOS IPS	
src	dst	Best Path	Cost	Best Path	Cost
s1	s4	1-2-11-8-7-4	9002	1-2-11-8-9-6-5-4	6002
s2	s5	2-11-8-7-5	12002	2-11-8-9-6-5	5002
s3	s9	3-10-9	10002	3-1-2-11-8-9	5002
s4	s11	4-7-8-11	7002	4-5-6-9-8-11	5002
s5	s7	5-7	5002	5-6-7	2002
s6	s3	6-9-10-3	11002	6-9-8-11-2-1-3	6002
s7	s6	7-5-6	6002	7-4-5-6	3002
s8	s3	8-11-10-3	11002	8-11-2-1-3	4002
s9	s1	9-10-3-1	11002	9-8-11-2-1	4002
s11	s3	11-10-3	10002	11-2-1-3	3002

While for the second stage, it will use the maximum packet size. The goal is to see the difference in latency on the ONOS default forwarding application (onos-app-fwd) and off-platform application ONOS-IPF when getting a load in their path. Both tests use the ICMP protocol. For the results of testing of the two stages as follows:

• Latency with ICMP packet minimum size (64 bytes).

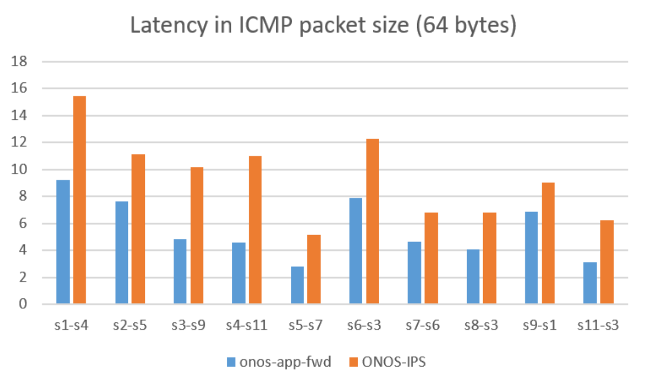


Fig. 12. Latency comparison using packet size 64 bytes

• Latency with ICMP packet maximum size (65.535 bytes)

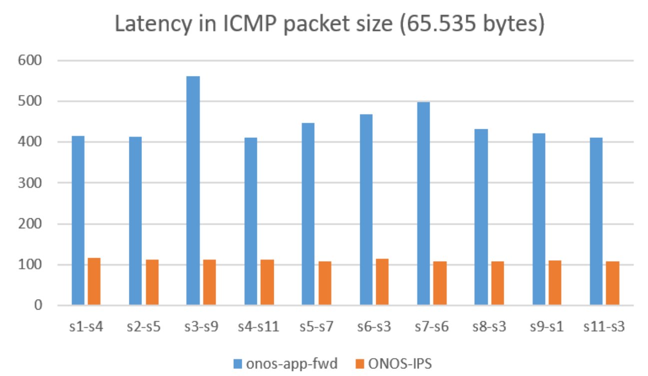


Fig. 13. Latency comparison using packet size 65.535 bytes

As we can see from the two graphs of comparisons in Figure 12 and Figure 13, there is varying latency when using the default ONOS forwarding application with this ONOS-

IPF application. From Figure 12, let us take the example from source on switch 1 to the destination host on switch 9 when using small packet size of 64 bytes, the latency generated by the ONOS-IPF application is much higher by around 15 ms which is almost double that of the default ONOS forwarding application that is around 9 ms.

But when a large packet of 65,535 bytes is used, then the latency generated by the ONOS-IPF application is much lower by 100 ms compared to the default ONOS forwarding application which can reach 400 ms as shown in Figure 13. The latency generated from the ONOS IPS application can be less into a quarter of the time from the default application.

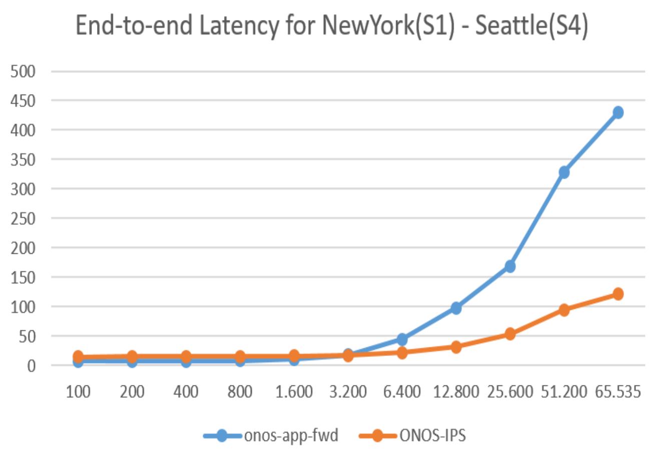


Fig. 14. Latency comparison from New York (S1) to Seattle (S4)

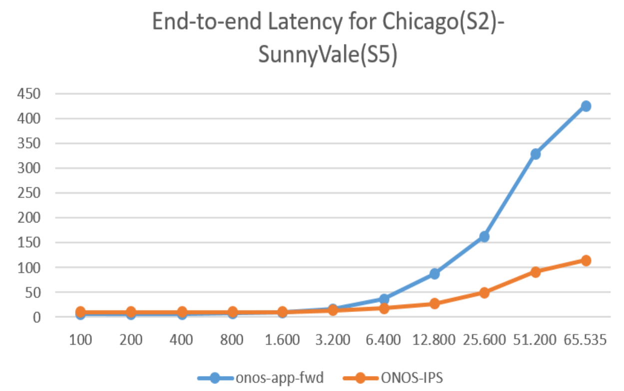


Fig. 15. Latency comparison from Chicago (S2) to Sunnyvale (S5)

As for more detail end-to-end measurement from table 3, the author uses more specific packet size which is from packet size 100 bytes to 65.535 bytes like in figure 14 and figure 15. In figure 14, there is an end-to-end latency measurement from New York City (S1) to Seattle City (S4). As we can see from that figure, that from packet size 100 bytes until bellow 3200 bytes the ONOS default forwarding application has the smallest latency than ONOS-IPF application, but when packet size is more than 3200 bytes ONOS-IPF application has smallest end-to-end latency. It also happened in figure 15, which is the measurement from Chicago City (S2) to Sunnyvale City (S5).

Besides the latency ratio, there is another comparison of the used path. Where default application has only one path, but the ONOS-IPF application has two paths where the first path is the primary path while the second path is the backup path. As shown in the Table 4, the primary path generated by the ONOS-IPF application is the path of the results of the Dijkstra algorithm calculation with the lowest cost value where the bandwidth of the link is the cost value, while the second path is the path from the default application which is obtained from the number smallest hop count like in Table 3.

TABLE 4. PATH COMPARISON DEFAULT VS PRIMARY VS BACKUP

Node		Path Availability		
		Default App	ONOS-IPF	
src	dst		Primary	Backup
s1	s4	1-2-11-8-7-4	1-2-11-8-9-6-5-4	1-2-11-8-7-4
s2	s5	2-11-8-7-5	2-11-8-9-6-5	2-11-8-7-4-5
s3	s9	3-10-9	3-1-2-11-8-9	3-10-9
s4	s11	4-7-8-11	4-5-6-9-8-11	4-7-8-11
s5	s7	5-7	5-4-7	5-7
s6	s3	6-9-10-3	6-9-8-11-2-1-3	6-9-10-3
s7	s6	7-5-6	7-4-5-6	7-5-6
s8	s3	8-11-10-3	8-11-2-1-3	8-11-10-3
s9	s1	9-10-3-1	9-8-11-2-1	9-10-3-1
s11	s3	11-10-3	11-2-1-3	11-10-3

When a network experiences interference with topologies such as broken links, this application will immediately perform re-calculations and re-install for a new primary path. Some testing is performed to calculate the length of time needed from the process of interrupting the path until a new best path is re-installed in the new topology. The results of this test can be seen in Table 5 and Table 6. In Table 5 the testing is carried out 100 times at 100 ms intervals for sending a packet and the result for each calculation is around 2190 ms. While Table 6 is for 10 times testing with a packet sent intervals of 1000 ms or 1 second, and the result is around 2000 ms or 2 seconds for each time calculation for every topology changes.

TABLE 5. PATH RECOVERY TIMES WITH INTERVAL 100 MS FOR 100 TIMES

Node		Fail over Time Recovery			
		Link	ONOS-IPF		
src	dst	Down	New Path	Time	xmt/rcv/loss
s1	s4	8-9	1-2-11-8-7-4	2200 ms	100/78/22%
s2	s5	8-9	2-11-8-7-4-5	2100 ms	100/79/21%
s3	s9	2-11	3-10-9	2100 ms	100/79/21%
s4	s11	6-9	4-7-8-11	2300 ms	100/77/23%
s5	s7	4-7	5-7	2200 ms	100/78/22%
s6	s3	8-11	6-9-10-3	2200 ms	100/78/22%
s7	s6	4-5	7-5-6	2300 ms	100/77/23%
s8	s3	11-2	8-11-10-3	2200 ms	100/78/22%
s9	s1	8-11	9-10-3-1	2200 ms	100/78/22%
s11	s3	2-1	11-10-3	2100 ms	100/79/21%

TABLE 6. PATH RECOVERY TIMES WITH INTERVAL 1000 MS FOR 10 TIMES

Node		Fail over Time Recovery			
		Link	ONOS-IPF		
src	dst	Down	New Path	Time	xmt/rcv/loss
s1	s4	8-9	1-2-11-8-7-4	2000 ms	10/8/20%
s2	s5	8-9	2-11-8-7-4-5	2000 ms	10/8/20%
s3	s9	2-11	3-10-9	2000 ms	10/8/20%
s4	s11	6-9	4-7-8-11	2000 ms	10/8/20%
s5	s7	4-7	5-7	2000 ms	10/8/20%
s6	s3	8-11	6-9-10-3	2000 ms	10/8/20%
s7	s6	4-5	7-5-6	2000 ms	10/8/20%
s8	s3	11-2	8-11-10-3	2000 ms	10/8/20%
s9	s1	8-11	9-10-3-1	2000 ms	10/8/20%
s11	s3	2-1	11-10-3	2000 ms	10/8/20%

V. CONCLUSION

From the simulation test results, it can be concluded that this ONOS-IPF application capable generate two paths which are primary and backup path using Dijkstra shortest path algorithm from intent into device configuration. Even though the path that generated by this ONOS-IPF application is further than ONOS default forwarding application, but it has the smallest end-to-end latency for packet size is more than 3200 bytes or 3 KB. And also time convergence for this application to recalculate that path is around 2000 ms or 2 seconds.

VI. FUTURE WORKS

There are some limitations for this application which is it only can do calculate the shortest path for 2 hosts only as 1 source and 1 destination. This limitation is because this application only uses intents installation feature in ONOS point-to-point for 1 to 1 host only, single-to-multi point for 1 to 2 host only and multi-to-single point for 2 to 1 host only. To be able to do path forwarding for multi-host, it needs a single-to-multi point for 1 host to a multi-host algorithm and multi-to-single point for multi to 1 host algorithm.

Also although this application already capable to the generated primary path and backup path this application still not implement a multipath algorithm, There are still many things that can be developed in this ONOS IPS application, such as development on the use of primary path and backup paths such as the addition of multicast algorithms. and the addition of new algorithms to the application so that it functions for multimedia data networks, where the main line with greater bandwidth can be used for the use of certain applications that require large bandwidth capacities such as voice data, video data, and other multimedia data. While the reserve path is the second route where it becomes a path for sending data such as text data, image data, and other data based on priority.

VII. ACKNOWLEDGMENT

This research is partially supported by Asi@Connect grant Asi@Connect-17-094 (IF050-2017) - OF@TEIN+: Open/ Federated Playground for Future Networks.

REFERENCES

[1] Marenda. D. A., Suranegara. G. M., Risdianto. A. C., Hakimi. R., Mulyana., "Intent-Based Path Selection for VM Migration Application with Open Network Operating System",2018 4th International Conference on Wireless and Telematics (ICWT).2018.pp:1-6.

[2] Ramadhan. F., Primananda. R., Yahya. W., "Implementasi Routing Berbasis Algoritme Dijkstra Pada Software Defined Networking Menggunakan Kontroler Open Network Operating System",Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.2018.vol:2.pp:2531-2541.

[3] Sanvito. D., Moro. D., Gulli. M., Filippini. I., Capone. A., Campanella. A.,"Enabling external routing logic in ONOS with Intent Monitor and Reroute service", 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft).2018.pp:332-334.

[4] Campanella. A.,"Intent Based Network Operations", 2019 Optical Fiber Communications Conference and Exhibition (OFC). 2019.pp:1-3

[5] Comer. D., Rastegatnia. A., "OSDF: An Intent-based Software Defined Network Programming Framework", 2018 IEEE 43rd Conference on Local Computer Networks (LCN).2018.pp:527-535

[6] Han. Y., Li. J., Hoang. D., Yoo.J., Hong. J. W.,"An Intent-based Network Virtualization Platform for SDN", 2016 12th International Conference on Network and Service Management (CNSM).2016.pp:353358

[7] Bianco. A., Giaccone. P., Mashayekhi. R., Ullio. M., Vercelleno. V.,"Scalability of ONOS reactive forwarding applications in ISP networks",Computer Communications 102.2016

[8] Fernandez. M. P.,"Comparing OpenFlow Controller Paradigms Scalability: Reactive and Proactive", 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA).2013.pp:1009-1016

[9] Jiang. J., Wuang. H., Liao. J., Chen. S.,"Extending Dijkstra's shortestpath algorithm for software defined networking", The 16th Asia-Pacific Network Operations and Management Symposium.2016.pp.1-4

[10] Ichiro. M. "Trends on Network Testbeds in the World",Journal of the National Institute of Information and Communications Technology.2005.Vol.52 Nos.3/4

[11] Goransson. P. Black. C, "Software Defined Network a Comprehensive Approach", MORGAN KAUFMAN. 2014

[12] Nadeau. T, Gray. K, "SDN: Software Defined Networks". O'Reilly Media. August 2013.

[13] Zodomolmolky. S, "Software Defined Networking with Openflow". PACKT Pulishing. Oktober 2013.

[14] Ortega-Arranz. H., Lianos. D. R., Gonzalez-Escribano. A., "The Shortest-Path Problem: Analysis and Comparison of Methods". Morgan&Claypool Publisher. 2014

[15] Buttler. B.,"What is intent-based networking?" NetworkWorld From IDG, 10 Mei.2019, <https://www.networkworld.com/article/3202699/what-is-intent-based-networking.html>

[16] Abilene Backbone Network.Internet2, 10 Mei.2019, https://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/internet-2/architecture.html

[17] Koshibe. A., "Intent Framework", 10 Mei 2019 <https://wiki.onosproject.org/display/ONOS/Intent+Framework