



High Availability Performance on OpenDayLight SDN Controller Platform (OSCP) Clustering and OpenDayLight with Heartbeat-Distributed Replicated Block Device (DRBD)

Andre Rizki Dewo Nugraha^{1*}, Ridha Muldina Negara², Danu Dwi Sanjoyo³
^{1,2,3} Faculty of Electrical Engineering, Telkom University
^{1,2,3}Jl. Telecommunications Canal Buah Batu Bandung 40257 Indonesia
Corresponding email: andrerdn17@gmail.com

Received 30 July 2018, Revised 10 August 2018, Accepted 21 August 2018

Abstract - Nowadays people are asking for a reliable network when technology at its limit. Software-Defined Network (SDN) is an answer to that problem of network development where all the control over the network becomes centralized. However, all services controlled by a centralized controller have a big disadvantage if the controller down. The High Availability (HA) is the solution. HA controller is divided into master and slave, when the master controller is down then the slave controller will respond to replace the function of the master controller. In this research, the system will be made by using two methods namely OpenDayLight SDN Controller Platform (OSCP) clustering and Heartbeat-DRBD (Distributed Replicated Block Device). OSCP clustering is a feature on the OpenDayLight controller that is ready to be used and only need to be configured, with OSCP the main and backup controller clustering will be on the connected cluster. Heartbeat-DRBD is an application commonly used to create High Availability systems on a server, but in this study will be used for the controller, Heartbeat will monitoring the main controller and if indicated to be down will move the resources to the backup controller with the DRBD application. From the simulation result shows that OSCP Clustering Failover and Failback average Time is 17 seconds while Heartbeat-DRBD is 23-45 seconds depends on how many switch and host are. While QoS parameters on both methods have similar value. It can be concluded that the High Availability system with OSCP Clustering method is more stable and good rather than Heartbeat-DRBD method to apply in a network.

Keywords - OSCP, Heartbeat, DRBD, Availability, QoS

Copyright © 2018 JURNAL INFOTEL

All rights reserved.

I. INTRODUCTION

Software-Defined Network (SDN) is an architecture for modern networks that are essential to managing an increasing and complex device. SDN can design, manage and implement a concept of data flows from a control system separated from hardware [1]. The control plane consists of a controller, while the data plane consists of switches, routers, and other network devices. The purpose of SDN is to increase network availability, network management, reduce network operating costs, and develop network innovation. The controller is a hardware or software that has a function to regulate the data flows between two entities, namely data plane and control plane.

There are many types of controllers such as Beacon, Onix, ONOS, OpenDayLight, OpenContrail, and Floodlight [2]. The type of controller used in this research is OpenDayLight Controller.

High Availability (HA) is a system that has failover capability, ensure that the application will run continuously and not interrupted with the failure of one subsystem because there is a subsystem replacement to backup data automatically. With High Availability, the system can maintain the downtime or service malfunction so it can make for better services.

On pre-existing researches, the controllers such as pox, floodlight, beacons use the HA system using

heartbeat and DRBD, but in the OpenDayLight controller, there is a feature that is capable of developing clustering, where this feature will afford a high availability system that is easy to configure and much more stable. However if using this feature the number of controllers needed is will be more, so the cost of developing the network will be more expensive.

In this research, the system will be analyzed with two different methods of high availability namely OSCP Clustering and Heartbeat-DRBD using complex topology of the network. So in the future people will know the best way to use high availability system when using an OpenDayLight controller.

The previous related research was conducted by M. Purwadadi. Namely with the title "High Availability Controller Software Defined Network Using Heartbeat and DRBD" [3]. this study more focused in the analysis using the POX controller, whereas this study was tested using an OpenDayLight controller.

Papers [3] use a simple topology and not measure for QoS parameters it only analyzes the failover and failback time if used different controllers. The result shows that the POX controller is better than OpenDayLight Controller according to that topology.

Another research for Heartbeat-DRBD is Reference from research conducted by Irpan Bahsyar [4]. Examine server performance by building High Availability cluster system using DRBD. The analysis performed is measuring downtime in the mail server and FTP server on the main server and backup.

Papers [4] works on the different architecture it used mail server and FTP server to measuring failover time using heartbeat-DRBD and the result is downtime for FTP server is faster than the mail server

In this research, the HA system will use to different method and topology. First HA with OSCP Clustering and the second is HA with heartbeat-DRBD. The topology that used in this research is generated by using topology command in mininet. the topology used is topo tree,4 and topo tree,5. Topo tree,4 has 16 hosts and 15 switches while topo tree,5 has 32 host and 31 switches.

In this study also measures QoS parameters such as delay, packet loss, and throughput. Where QoS will be analyzed during failover and failback process occurs. That process will measure with a third application named Wireshark for more accurate data.

The other parts of this paper will discuss: Section II will discuss the system design of HA with OSCP Clustering and Heartbeat-DRBD method. Section III will show the result of the research. Section IV discusses the results and section V is the conclusion.

II. RESEARCH METHOD

A. Software Defined Network

Software Defined Network (SDN) is a new paradigm in controlling and managing computer networks. One important feature of SDN is the ability to provide a network-wide abstraction. Keller et al. discuss the idea of the "platform as a service" model for networking. According to the authors, it is a common trend to decouple infrastructure management from service management [5]. The basic concept of SDN is the separation between the control plane in charge of determining how a packet will be forwarded to the data plane in charge of forwarding the packet. With a separate control plane, administrators can modify the protocol along the network device. Data plane represents a traffic router device in SDN architecture. Because the controller requires communication with the network infrastructure, a protocol is needed to control and manage interfaces between various network devices. The most widely used protocol is OpenFlow [5].

Controllers take decisions that commonly used by control planes, such as routing, switching, Quality of Service (QoS), and others. Automation is possible with the development of APIs that detect changes of flow in the network. For example, if a traffic is moved from one switch to another, then routing or Quality of Service (QoS) will also change automatically [6, 7].

The control plane is part of the SDN that forms the forwarding table based on the data set. The forwarding table will be used by the data plane to forward the traffic between the ingress and port on a device. The dataset used to store network topologies called Routing Information Base (RIB). After the RIB formed, the RIB will be used to program Forwarding Information Base (FIB) which is used by the data plane to forward the packet. In its implementation, the controller can work with three modes, reactive mode, proactive mode, and hybrid mode.

Data plane is part of the SDN that works with datagrams (on cable, fiber, or wireless) through a series of operations performed at link-level. The datagram will be forwarded based on the FIB table that has been programmed by the SDN controller. Data plane is the lowest layer of the SDN that contains network devices such as routers, physical/virtual switches, access points etc. Data plane can communicate with SDN controller using OpenFlow protocol [7].

Another important advantage of SDN is that it enables innovation and flexibility. If the control and data plane are managed by hardware network devices, there is little room for innovating and experiment, as the software or firmware of those devices cannot be easily modified. Instead, by having access to a software component to manage the control plane, many ideas can be explored.

B. OpenDayLight

Gopal, I., (2013) states that OpenDayLight is an Open Source software project under the auspices of the Linux Foundation. The goal is to implementing and innovating Software Defined Networking (SDN) through the creation of a common framework among industry [7]. OpenDayLight is an implementation of the Software-Defined Network (SDN) concept and utilizes the following tools: Maven, JAVA interface and REST API.

ODL contain several component projects and the inclusion of three layers. In top layer is business and network logic applications reside that use the controller to gather network intelligence, run algorithms to control and monitor network behavior. The core is the framework layer where the controller exists [8]. The architecture of OpenDayLight is shown by Fig.1.

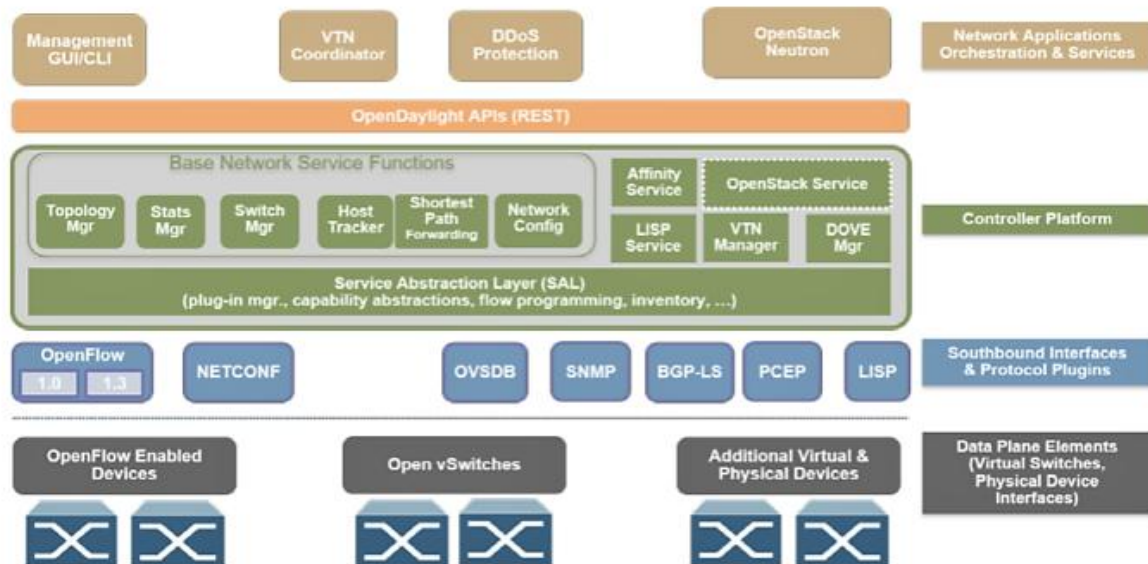


Fig.1. Architecture of OpenDayLight [9]

C. Mininet

Mininet is a network emulator which runs a collection of end-hosts, switches, routers and links on a single Linux kernel. It is an approach of using operating system virtualization features, and processes to allow it to scale up to hundreds of nodes. Users can implement a new network feature or a new architecture, test it on large topologies with application traffic and then deploy the same code and test scripts into a real production network [12].

D. Heartbeat

Heartbeat is an application that can detect if the main server down then Heartbeat will automatically direct the primary server role to the backup server. Heartbeat runs the initialization script to run another service when Heartbeat starts or it can also turn off other services when Heartbeat is turned off. Heartbeat

also performs an inspection of a path connection to ensure the availability of the node.

In SDN architecture, northbound application program interfaces (APIs) [10] are used to communicate between the SDN Controller and the applications running on the network. The northbound APIs can be used to provide new application and enable automation of the network to arrange with the needs of different applications via SDN control plane programmability.

Southbound application program interfaces (APIs) provides communication between the SDN Controller and the switches and routers of the network. Southbound APIs provides efficient control over the network and enables the SDN Controller to dynamically make changes according to real-time requirements and demands [11].

E. DRBD

DRBD (Distributed Replicated Block Device) is a storage block device designed to build High Availability systems. The performance of DRBD (Distributed Replicated Block Device) is done by mirroring all through the network and using the network as a transmission medium to synchronize data between disks in real-time. DRBD can be analogous to the RAID-1 mechanism (mirroring, or other RAID type using mirroring principle), which duplicates data through the network. This data duplication is done in the block devices mechanism, not in the form of raw data. If RAID-1 duplicates the contents and data of a hard disk or partition to another disk or partition, DRBD does the same thing, it's only done over the network. DRBD and harddiskRAID are mutually supportive. The different of DRBD and harddiskRAID is DRBD has one advantage over

harddiskRAID, that backup servers are separated with the backup source. This separation brings preventive benefits, if there is a problem on one server, the other server will act as a replacement server. If the primary server is restored, the control will be restored to the primary server [13].

DRBD works in several ways, among others [14] First Uptime is Replication done continuously when the application modifies data on the block device. Second By synchronous mirroring, the file system on the active node is notified that the writing process has been successful when writing has been done on both block devices of each node. Synchronous mirroring (protocol C DRBD) is an option for the local network so as not to lose a single transaction in case of crashes during writing on the active node. The third is by Asynchronous, the file system will be notified that the writing completes when the data is finished writing on the local disk. Asynchronous is required when doing remote mirrors.

F. Quality of Service (QoS)

It is the ability of a network to provide good service by meeting several criteria of parameters such as delay packet loss and throughput. QoS (Quality of Service): "the collective effect of service performance which determines the degree of satisfaction of a user of the service"[15]. There are several QoS parameters.

Delay is the time it takes data to travel from distance to destination Delay can be affected by distance, physical media, congestion or long processing time. According to TIPHON, the magnitude of the delay can be classified as follows [15]:

Table 2. Delay TIPHON Standard

| Delay Category | Delay (ms) | Index |
|----------------|------------|-------|
| Very Good | <150 | 4 |
| Good | 150 - 300 | 3 |
| Average | 300 - 450 | 2 |
| Bad | >450 | 1 |

Packet loss is a parameter in QoS that describes a condition that determines the number of packets lost, can occur due to collision and congestion on the network. This has an effect on all applications because retransmission will reduce overall network efficiency even if there is a sufficient amount of bandwidth available for the application. The amount of packet loss is specified as follows [15]:

Table 3. Packet Loss TIPHON Standard

| Packet Loss Category | Packet Loss (%) | Index |
|----------------------|-----------------|-------|
| Very Good | 0 | 4 |
| Good | 3 | 3 |
| Average | 15 | 2 |
| Bad | 25 | 1 |

Throughput represents the total number of successful arrival packets observed at the destination over a specified time interval. Usually, throughput is always associated with bandwidth. Because the throughput can indeed be called the bandwidth in the actual conditions. Bandwidth is more fixed while throughput is dynamic depending on the current traffic. The amount of throughput is calcified as follows [15]:

Table 4. Throughput TIPHON Standard

| Throughput Category | Throughput (%) | Index |
|---------------------|----------------|-------|
| Very Good | 100 | 4 |
| Good | 75 | 3 |
| Average | 50 | 2 |
| Bad | <25 | 1 |

G. Heartbeat-DRBD System Design

In this research Heartbeat and DRBD will work together as a third application that provides High Availability system. Controllers version that used in this research is OpenDayLight Boron SR 4, each controller has the same data configuration in Heartbeat and DRBD so in this method there are two controllers that identical. The system design is shown in Fig.5.

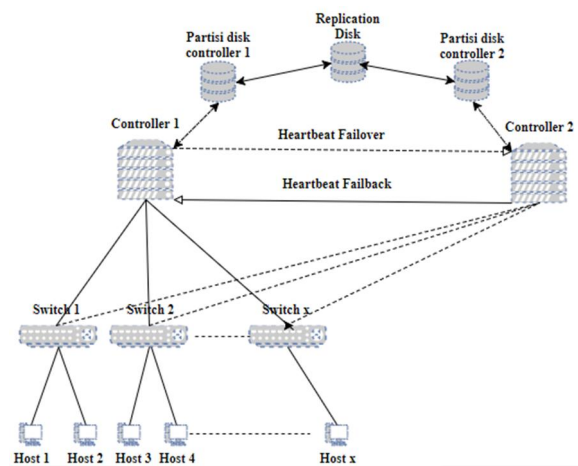


Fig.5 Heartbeat-DRBD System Design

Fig.5 shows that every controller has one partition disk. When Master is up (Controller 1) the replication disk active in disk partition 1 with name /dev/drbd0. The heartbeat will monitor the performance of controller 1. when controller 1 down then heartbeat will send a signal to DRBD on slave controller to do failover and then disk partition 2 will become active so DRBD will move replication disk that contains resource to disk partition 2.

The testing scenario of QoS parameters occurs when controller master is going down by shutdown the opendaylight and heartbeat system the DRBD will start to replicate the data. while failover happens the

traffic will be given to the network with an iperf command for 20 seconds from host 1 to host 16 for 16 host topology and from host 1 to host 32 for 32 host topology by the time traffic is given the Wireshark application will capture the packet data and measure the QoS parameters value.

H. OSCP Clustering System Design

Oscp clustering is a feature that is already in OpenDayLight and ready to install. By using SCP clustering the controller will form a cluster with an identical controller inside the cluster. In this research, OSCP Clustering uses three controllers of OpenDayLight Boron SR4. The System Design is shown in Fig.6

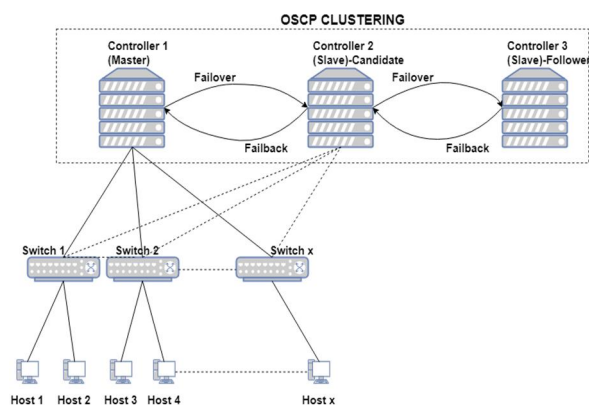


Fig.6. OSCP Clustering System Design

In Fig.6 Shows that in this research there is three controller as one master and two slaves, the slaves have two states first as a candidate and the second one as a follower. Different is the failover only occurs from master to slaves-candidate when the controller 1 down role as a master moved to controller 2. When it occurs in controller 3 roles as a slave-candidate and controller 1 role as slave-follower.

In this research, the scenario that used is failover time will be analyzed when it occurs from controller 1 to controller 2 and from controller 2 to controller 3. The failback time is analyzed when it occurs from controller 3 to controller 1.

The testing scenario of QoS parameters occurs when controller master is going down by shutdown the opendaylight. While the controller slave-candidate is going to be a master the data-shard from it machine will start to generate the network data from the previous master then the failover is happens, the traffic will be given to the network with iperf command for 20 seconds from host 1 to host 16 for 16 host topology and from host 1 to host 32 for 32 host topology by the time traffic is given the Wireshark application will capture the packet data and measure the QoS parameters value.

III. RESULT

Based on the simulation result, the failover and failback time comparison between OSCP Clustering and Heartbeat DRBD is shown in Fig.7. In the simulation, the conditions were predetermined as follows:

- The Topology of the network is Tree with 16 hosts 15 switches and 32 hosts 31 switches
- Traffic will be given to the network when failover and failback occurs for 20 seconds
- The traffic will be measured from host 1 to host 16 and from host 1 to host 32
- The QoS parameters that are measured is a delay, packet loss, and throughput.

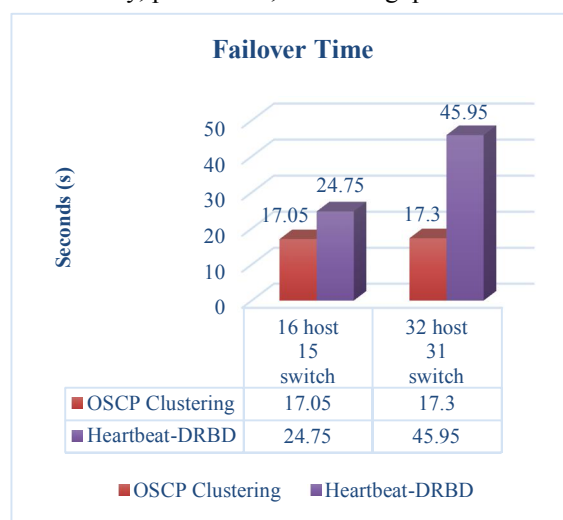


Fig.7. Failover Time Result

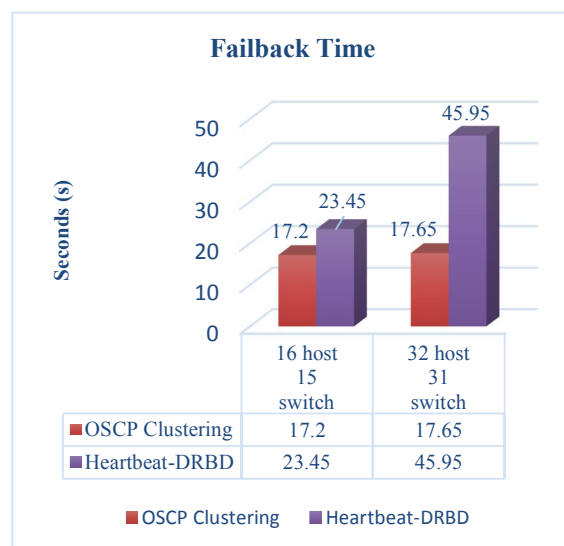


Fig.8. Failback Time Result

The failover and failback time that occur on each method is similar but if we compare between both methods it has different values.

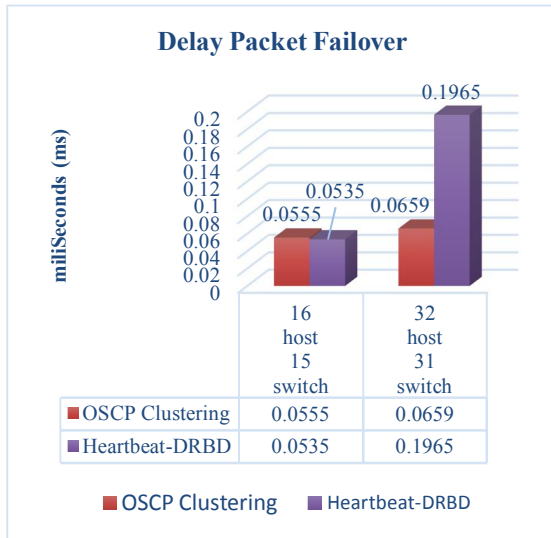


Fig.9. Delay Packet Failover

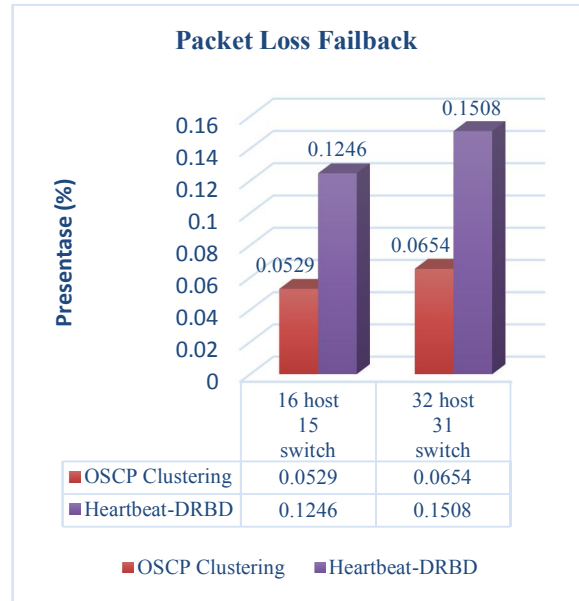


Fig.12. Packet Loss Failback

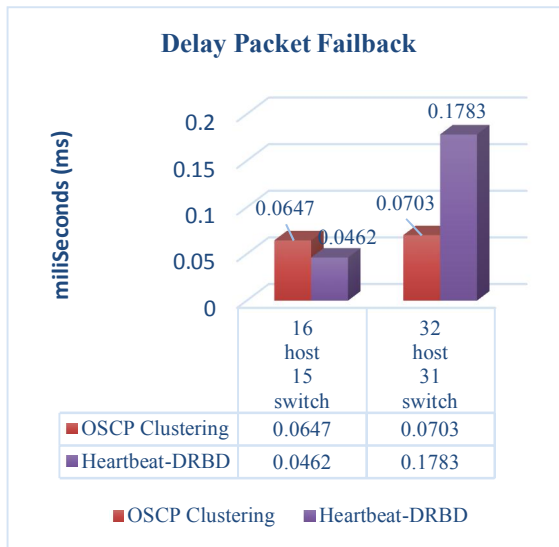


Fig.10. Delay Packet Failback

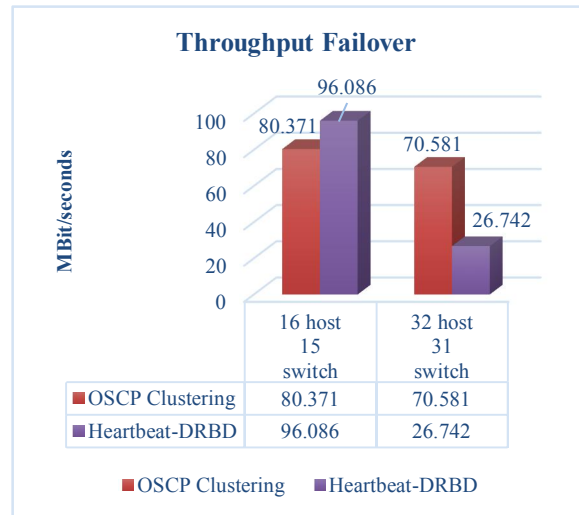


Fig.13. Throughput Failover

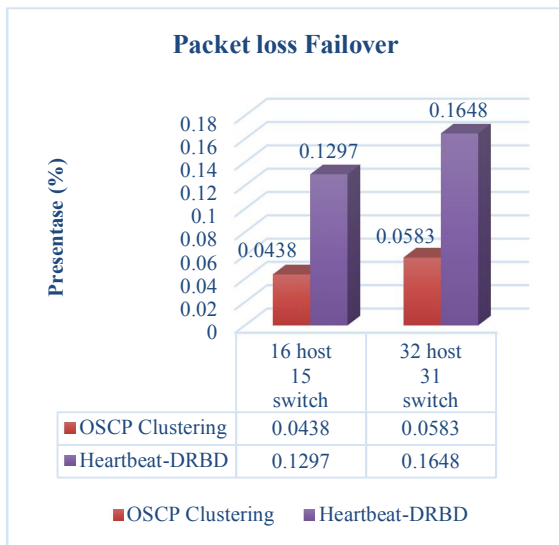


Fig.11. Packet Loss Failover

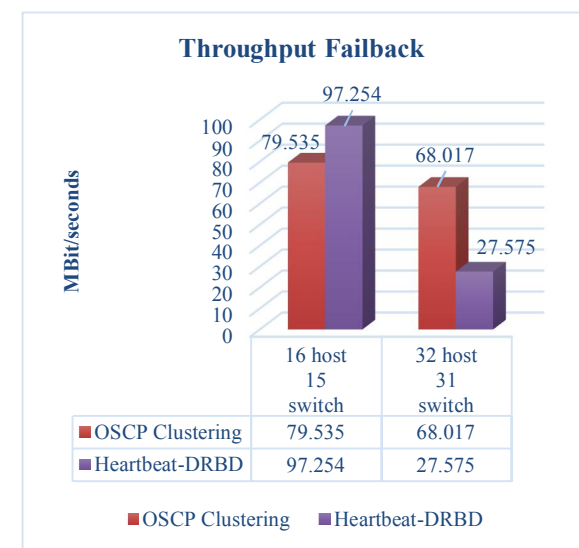


Fig.14. Throughput Failback

IV. DISCUSSION

In Fig.7, the comparison between OSCP Clustering and heartbeat-DRBD shows that both methods have different time for failover depend on how many hosts and switch in the network is. For OSCP clustering 17.05 needed for failover in the first topology for the second topology its need 17.3 seconds for failover. For heartbeat-DRBD, the failover time is increased significantly for the second topology its increases about 20 seconds.

In Fig.8, the failback time is not too different from the failover process. OSCP clustering failback time remains stable within 17 seconds while the heartbeat-DRBD still increases 20 seconds. This indicates that the clustering feature provided by OpenDayLight in addition to providing high availability feature, it also provides scalability.

In Fig.9 and Fig.10 shows that delay packet from OSCP Clustering its about 0.055ms to 0.070ms for failover and failback time while heartbeat-DRBD is formed 0.046ms to 0.196ms. Both methods are a good performance based on TIPHON standardization where its show <150ms for the very good index.

In Fig.11 and Fig.12, the comparison of packet loss from both method also shown very good index based on TIPHON standardization where its shown <0% packet loss.

In Fig.13 and Fig.14, the result of Throughput comparison for both methods is slightly different for the second topology. OSCP Clustering throughput is more stable, it's around 68 to 80 Mbit/sec while heartbeat-DRBD is around 26 to 97 Mbit/sec. It shows that the heartbeat-DRBD is very sensitive with the addition of host and switches. each addition will affect the performance of the controller that will use more resources so that the resulting throughput will go down very drastically. This is different from OSCP Clustering which has a scalability system where the addition of hosts and switches will not affect too much for the performance of the controller.

When failover and failback occur in OSCP Clustering every controller has its own database of the system called data-shard so the new master controller already knows the configuration of the network that controlled. This is different from the heartbeat-DRBD where data from the network resides on the master disk controller partition when the failover occurs the new master controller will read data of the network residing on the partition disk that has just been replicated by DRBD from the previous master controller, so it takes a longer time. The number of hosts and switches then the replicated data will be bigger so it will take a longer time to do failover and failback.

On the previous research, the performance of POX controller using Heartbeat-DRBD is longer to take failover and failback time because the OpenDayLight

controller has faster run-up time compared to POX controller.

V. CONCLUSION

Based on the results of High Availability performance research using OpenDaylight with Heartbeat-DRBD proves that this application is very sensitive with the numbers of host and switch. According to DRBD function, it only provides the availability of the storage disk not the resources of the controller while High Availability using OSCP Clustering feature is offering scalability to increase the resources of the controller. It can be conducted that Heartbeat-DRBD is more suitable to use in database servers for its High Availability system and OSCP Clustering is very affordable to use in SDN controller due to its stability and scalability.

REFERENCES

- [1] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software Defined Networking," *IEEE Commun. Surveys & Tutorials*, Vol. 17, no. 1, pp. 27-51, Jun. 2014.
- [2] Marowsky-Bree, L, "A New Cluster Resource Manager for Heartbeat," Proceedings of UKUUG LISA/Winter Conference High-Availability and Reliability, *The UK's Unix & Open Systems User Group*, Bournemouth, 2014.
- [3] Purwiadi, M., Yahya, W., & Basuki, A. "High Availability Controller Software Defined Network using Heartbeat and DRBD, *ONF White Paper*, pp. 2297-2306, 2018.
- [4] Irpan Bahsyar, "implementation and analysis performance of high availability server system using a distributed replicated block device (DRBD)". TelkomUniversity, p. 201, 2017.
- [5] Lara, A., Lara, A., Kolasani, A., & Ramamurthy, B. "Network Innovation using OpenFlow: A Survey Network Innovation using OpenFlow: A Survey". *CSE Journal Articles Computer Science and Engineering, Department of*, vol. 16, no. 1, pp. 493-512, 2014.
- [6] F. Granelli, et al. "Software-defined and virtualized wireless access in future wireless networks: scenarios and standards," *IEEE Communications Magazine*, Vol. 53, no. 6, pp. 26-34, Jun. 2015.
- [7] D. Zhao, M. Zhu, and M. Xu, "Supporting One Big AP illusion in enterprise WLAN: An SDN-based solution," *Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1-6, Oct. 2014.
- [8] Asadollahi, S., Goswami, B., & Gonsai, A. M. "Implementation of SDN using OpenDayLight Controller". *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 218-227, 2017.
- [9] Abdillah, D., Sibaroni, Y., & Ummah, I. "Design and analyze a Virtual network based on Software- Define Networking (Sdn) Design and Analysis of Virtual Network Based on Software- Define Networking". *E- Proceeding of Engineering*, vol. 3, no. 1, pp. 1247-1252, 2017.

- [10] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Es-teve Rothenberg, "Software-Defined Networking: A Comprehensive Survey". *Arxiv:1406.044v3 8th*, Oct. 2014.
- [11] Jyothi, K. S., & Girish, L. "Effective disaster recovery using Software-defined networking". *International Journal of Engineering Trends and Technology (IJETT)*, vol. 24, no. 5, pp. 265–271, 2015.
- [12] B. Lantz, B. Heller, N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks". In *HotNets*. ACM, 2010.
- [13] Mohammad, F. "Implementation Of Disaster Recovery Plan Without The Failed System Using Drbd And Heartbeat On Data Center Fkip Uns". *Jurnal on Networking and Security (IJNS)*, Vol. 2, No 2, pp. 64–69, Apr. 2013.
- [14] Miers, K. (n.d.). "High Availability with Linux Using DRBD and Heartbeat". *Development of technology information and computer science journal*. Vol. 2, No. 8. pp. 1-22, Oct. 2004.
- [15] Pranata, Y. A., Fibriani, I., & Utomo, S. B. (n.d.). "Analysis and Optimation Performance Quality of Service on Communication Services Data Using NS - 2 di PT. PLN (Persero) Jember". *ISSN: 1410-2331*, Vol 20, No. 2, pp. 149–156, Jun 2016.