



Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: Sumadi Fauzi
Assignment title: Penelitian Dosen
Submission title: Analysis of failover mechanism in SDN
File name: Kepangkatan_Fauzi_B11.pdf
File size: 466.39K
Page count: 6
Word count: 2,287
Character count: 12,162
Submission date: 19-Jul-2023 08:48PM (UTC+0700)
Submission ID: 2133565388

The thumbnail shows the first page of the document. On the left, there is a vertical grey bar with the text "AIP Conference Proceedings" and the AIP Publishing logo. The main content area has the title "Analysis of failover mechanism in SDN" in bold. Below the title, it says "Cite as: AIP Conference Proceedings 2453, 030016 (2022); <https://doi.org/10.1063/5.0094300>" and "Published Online: 25 July 2022". The authors are listed as "Fauzi Dwi Setiawan Sumadi, Jurdan Wahyu Adi Saputra and Mahar Fauzrahman". There are icons for "View Full" and "Export Citation". At the bottom, there is a blue banner for "AIP Author Services English Language Editing" with the text "High-quality assistance from subject specialists" and a "LEARN MORE" button. The footer contains the citation information and the ID "2453, 030016".

Analysis of failover mechanism in SDN

by Sumadi Fauzi

Submission date: 19-Jul-2023 08:48PM (UTC+0700)

Submission ID: 2133565388

File name: Kepangkatan_Fauzi_B11.pdf (466.39K)

Word count: 2287

Character count: 12162

Analysis of failover mechanism in SDN

3

Cite as: AIP Conference Proceedings 2453, 030016 (2022); <https://doi.org/10.1063/5.0094300>
Published Online: 25 July 2022

1

Fauzi Dwi Setiawan Sumadi, Jurdan Wahyu Adi Saputra and Mahar Faiqurahman




View Online




Export Citation



 Author Services

English Language Editing
High-quality assistance from subject specialists

[LEARN MORE](#)



3

AIP Conference Proceedings 2453, 030016 (2022); <https://doi.org/10.1063/5.0094300>

2453, 030016

© 2022 Author(s).

Analysis of Failover Mechanism in SDN

Fauzi Dwi Setiawan Sumadi^{a)}, Jurdan Wahyu Adi Saputra^{b)}, Mahar Faiqurahman^{c)}

⁴
Department of Informatics, Universitas Muhammadiyah Malang, Malang, Indonesia

Corresponding author: ^{a)}fauzisumadi@umm.ac.id
^{b)}jurdan.aho@gmail.com
^{c)}mahar@umm.ac.id

¹
Abstract. Software Defined Network (SDN) has been widely utilized for resolving the traditional network problems by separating the control and forwarding mechanism. However, the implementation of centralized network management is vulnerable to a single point of failure leading to a comprehensive problem such as., unreachable network. Therefore, this paper proposes an analysis of the failover approach on high availability controller, performed by utilizing Heartbeat and DRBD (Distributed Replication Block Device), with the main objectives of directing the management process into the secondary controller, during the occurrence of a crash on the main controller. The data replication process was performed by the DRBD in real-time. The experiment's results indicate that RYU gains the shortest failover and failback time at 1.3 s than the other controllers (POX and OpenDaylight). In terms of the Quality of Service (QoS), RYU also maintains the jitter, throughput, and packet loss variables which is better than POX and OpenDaylight.

INTRODUCTION

²
Software Defined Network (SDN) contains a network architecture concept to design, manage, and implement a computer network by separating the control and forwarding plane, controlled through one controller application [1]. SDN aims to increase network availability, simplify the network management process, reduce network costs, and develop network innovation [2]. In addition, SDN controller is responsible for managing the flow of data on the whole network including: Beacon, Onix, ONOS, OpenDayLight, Open Contrail, Ryu, POX, and Floodlight [3]. One of the protocols used for the communication between the forwarding device and the controller is the OpenFlow protocol, providing a standard for the controller to command, learn, and create a specific command for the dataplane devices.

However, the SDN concept is deemed vulnerable to single-point failure, probably degrading the network performance or even experiencing an unreachable state. The general example includes where one of the controllers in the SDN experiences down or inactivity hindering proper connection to overcome this problem, it is thus essential to have a system which could handle it, which is the High Availability maintaining services or applications running and can recover from component or system failures with a minimum application termination impact. One type of High Availability concept is failover availability, employing the two servers: the primary server and the backup server with identical data on each server. When the system with this concept usually runs, only the main server is in charge of serving all users. However, when the main server goes down and the backup server detects it, the backup server will replace the function of the main server. Therefore, the failover mechanism provides two or more connection lines when one path is down by diverting to the other. Prior related studies have attempted to investigate the failover possibility in SDN [1-6]. Paper in [1] proposed a dynamic failover mechanism utilizing network hypervisor of OpenVirtex, which mainly discussed the link backup process during the link failure event. This paper however did not concern about the controller crash event. The second paper [2] presented FCF-M method for handling multi-domain failures, deployed in EstiNet. The authors in [3] implemented Heartbeat to perform a failover mechanism using NOX13 of lib controller in High Availability Controller Architecture (HAC). The results indicated that the HAC could maintain the performance, despite numerous link stress. Paper [4] introduced a comprehensive, fast recovery of link failure using a backup path for resolving data and control channels. The authors further installed backup paths on the group table in the OpenFlow switch to maintain data channel recovery and proposed Control Plane Spanning Tree (CST) to retain and restore controller state to the affected switch. Paper in [5] utilized Heartbeat as media for developing the Fast and Load-aware Controller Failover (FLCF) emulated in EstiNet using OpenDaylight controller. The authors in [6] presented the two methods, including the Greedy failover and the Prepartitioning failover. The system was emulated in Mininet using Heartbeat, suggesting sending more LLDP messages to reduce the processing time.

Based on the previous related works, this paper was directed to analyze several controller performances, including-

ing RYU, POX, and OpenDaylight, for implementing the failover and failback process, contributing the comparative analysis using both Heartbeat and DRBD.

RESEARCH METHOD

The research was conducted using the emulation method in Mininet [7] environment. In general, the failover process was directly handled by the Heartbeat [8] when the main controller was down/crash, allowing the Heartbeat to automatically direct the main controller role to the backup controller. In addition, DRBD [9] performed a comprehensive backup from the main controller to the real-time backup controller on the SDN environment using the Python application installed in the controller. The emulation topology is depicted in Figure 1.

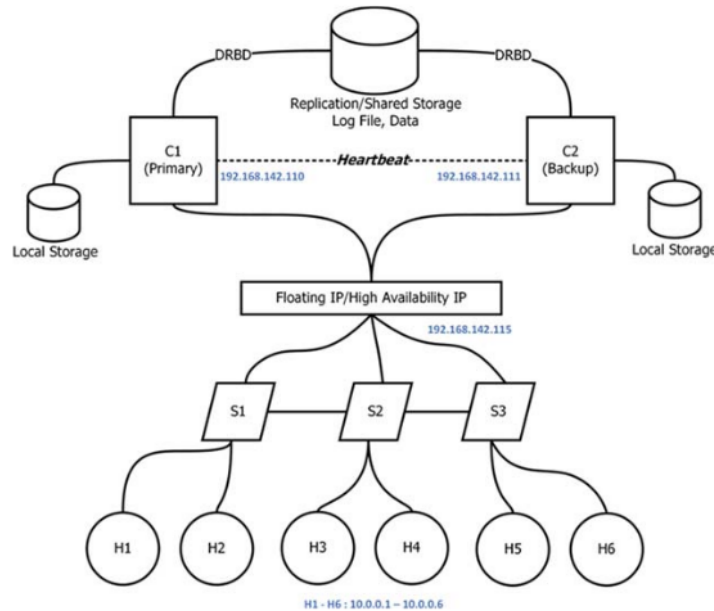


FIGURE 1. Emulation's topology

The SDN infrastructure implemented on Mininet utilized a simple topology where there were two controllers (C1 and C2), including the main controller and the backup controller, three switches (S1, S2, and S3), six hosts (H1-H6), with their respective functions. The controller was in charge of controlling the network and specifying the forwarding mechanism to the switch. The controllers deployed in this research were RYU [10], POX [11], and OpenDaylight [12]. Switch (Open Virtual Switch/OvS [13]) served to perform forwarding functions based on the Flowrule, set by the controller. The host was utilized to test network performance by sending the data. OpenFlow was deployed to connect between the control layer and the forwarding layer.

In terms of the failover procedure, the three types of VM (Virtual Machine) controllers include, including the primary controller VM (C1), the backup controller VM (C2), and the Mininet VM (OvS and the Hosts). Generally, the DRBD would provide a real-time data synchronization process on an identical disk partition for storing the primary data, duplicated in a replication disk. The failover and failback procedure were handled by Heartbeat, where the Mininet VM could directly access the controller through the Floating IP/High Availability IP (192.168.142.115). When the Heartbeat detected the primary controller in a downstate, it would automatically execute the failover procedure by performing the resource takeover from the replication disk located on the main controller.

Meanwhile, the secondary/backup controller would automatically deploy the failback procedure specified on the Heartbeat configuration. The Mininet VM would also redirect its connection to the backup controller through Floating IP. Subsequently, the backup controller operated as the new primary controller while the primary controller would reboot and functioned as the new backup controller.

In order to analyze the impact of the failover and the failback process on several types of controllers (RYU,

POX, and OpenDaylight), some variables were calculated, including the required time to perform the failover and failback, the jitter, packet loss, and throughput. The calculation process for acquiring the failover and failback was derived from the log of the active controller. At the same time, the QoS variables were extracted from Wireshark and Iperf during normal packet transmission between Host 1 and Host 3 as a client.

RESULTS AND DISCUSSION

The experiment's results are categorized into two main variables, which include: the time for performing both the failover and failback and the QoS during the specified processes. As illustrated in Table I, the average time extracted from RYU pointed at 1.3 s. RYU demonstrated the fastest controller to implement the failover and failback, followed by OpenDaylight and POX, respectively. This result might occur since the program complexity and modularity in RYU was less than the other controllers. In terms of the QoS calculation, the experiment was executed by employing the Iperf application by sending packets from H3 that functioned as a client.

TABLE 1. The average time of the failover and failback process in s

Number of Experiment	RYU	POX	OpenDaylight
1	1	24	2
2	2	33	1
3	1	14	2
4	1	28	1
5	1	10	2
6	2	29	1
7	1	27	2
8	1	26	1
9	2	28	1
10	1	14	2
Total	13	233	15
Average	1.3	23.3	1.5

Meanwhile, H1 was pointed as a U11 server that would send UDP traffic for 200 seconds. The results indicated similar pattern for QoS variables, illustrated in Table 2, Table 3, and Table 4. The average throughput, jitter, and packet loss values were directly proportional to the time value for deploying the failover and failback.

TABLE 2. The average of throughput in kbps

Number of Experiment	RYU	POX	OpenDaylight
1	943	491	328
2	549	890	549
3	1051	855	335
4	1052	571	868
5	1052	760	929
6	1051	115	931
7	1051	194	442
8	544	769	404
9	985	907	1017
10	1051	636	414
Total	9329	6188	6217
Average	932.9	618.8	621.7

Since the processing delay occurred during the mentioned processes might affect the communication between regular clients, therefore RYU has gained the best performance than the other controllers. The average throughput, jitter, and packet loss produced by the RYU failover process were 932.9 kbps, 0.02 ms and 0.3 %. It was thus possible to implement a failover process for handling the crash event in the SDN environment.

TABLE 3. The average of jitter in ms

Number of Experiment	RYU	POX	OpenDaylight
1	0.139	0.009	328
2	0.009	0.008	549
3	0.006	0.016	335
4	0.017	0.009	868
5	0.005	0.007	929
6	0.011	0.011	931
7	0.008	0.007	442
8	0.011	0.005	404
9	0.01	0.008	1017
10	0.008	0.012	414
Total	0.224	0.092	6217
Average	0.0224	0.0092	621.7

TABLE 4. The average of packet loss in percentage

Number of Experiment	RYU	POX	OpenDaylight
1	0.27	16.4	0.32
2	0.31	9.3	0.33
3	0.3	6.7	0.31
4	0.24	8.7	0.34
5	0.32	4.6	0.3
6	0.37	14	0.31
7	0.31	13	0.32
8	0.26	15	0.31
9	0.34	14	0.31
10	0.3	9.7	0.42
Total	3.02	111.4	3.27
Average	0.302	11.14	0.327

CONCLUSION

Based on the results and discussion section, this study concluded that the most responsive controller for performing both the failover and failback processes was RYU, confirming that RYU could maintain the performance during the crash and reboot event. This mechanism might be obtained because RYU provided less modularity on its component. Future research is encouraged to combine the load balancing using the failover mechanism in supporting the distributed applications in the SDN environment.

5

ACKNOWLEDGMENTS

The researchers of this study would like to express their profound gratitude to Universitas Muhammadiyah Malang and UMM Informatics Laboratory for adequately supporting this project.

REFERENCES

1. K. Ko, D. Son, J. Hyun, J. Li, Y. Han, and J. W. Hong, "Dynamic failover for SDN-based virtual networks," 2017 IEEE Conference on Network Softwarization (NetSoft), 1–5 (2017).
2. Y.-C. Chan, K. Wang, and Y.-H. Hsu, "Fast Controller Failover for Multi-domain Software-Defined Networks," 2015 European Conference on Networks and Communications (EuCNC), 370–374 (2015).
3. V. Pashkov, A. Shalimov, and R. Smeliansky, "Controller failover for SDN enterprise networks," 2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), 1–6 (2014).
4. R. Hwang and Y. Tang, "Fast Failover Mechanism for SDN-Enabled Data Centers," 2016 International Computer

- Symposium (ICS), 171–176 (2016).
5. K. Fang, K. Wang, and J. Wang, “A fast and load-aware controller failover mechanism for software-defined networks,” 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), 1–6 (2016).
 6. M. Obadia, M. Bouet, J. Leguay, K. Phemius, and L. Iannone, “Failover mechanisms for distributed SDN controllers,” 2014 International Conference and Workshop on the Network of the Future (NOF), Vol. Workshop, 1–6 (2014).
 7. K. Kaur, J. Singh, and N. S. Ghumman, “Mininet as Software Defined Networking Testing Platform,” International Conference on Communication, Computing & Systems (ICCCS–2014) (2014).
 8. M. A. Aglan, M. A. Sobh, and A. M. Bahaa-Eldin, “Reliability and Scalability in SDN Networks,” 2018 13th International Conference on Computer Engineering and Systems (ICCES), 549–554 (2018).
 9. S. Park, I. Y. Jung, H. Eom, and H. Y. Yeom, “An analysis of replication enhancement for a high availability cluster,” *Journal of Information Processing Systems*, **9**, 2 (2013).
 10. S. Asadollahi, B. Goswami, and M. Sameer, “Ryu controller’s scalability experiment on software defined networks,” 2018 IEEE International Conference on Current Trends Advanced Computing, ICCTAC 2018 (2018).
 11. S. Kaur, J. Singh, and N. S. Ghumman, “Network Programmability Using POX Controller,” International Conference on Communication, Computing & Systems, (2014).
 12. J. Medved, R. Varga, A. Tkacik, and K. Gray, “OpenDaylight: Towards a model-driven SDN controller architecture,” *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, WoWMoM 2014* (2014).
 13. B. Pfaff et al., “The design and implementation of open vSwitch,” *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2015*, 117–130 (2015).

Analysis of failover mechanism in SDN

ORIGINALITY REPORT

18%

SIMILARITY INDEX

17%

INTERNET SOURCES

5%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	9%
2	repositor.umm.ac.id Internet Source	3%
3	Submitted to Universitas Negeri Surabaya The State University of Surabaya Student Paper	2%
4	Evi Dwi Wahyuni, Danang Agung Permadi, Eko Budi Cahyono. "Augmented reality as a media for Reog Ponorogo art figure introduction", AIP Publishing, 2022 Publication	1%
5	Achmad Fauzan Hery Soegiharto, Mulyono, Muhammad Rasyid Ridho, Lilis Yuliati, Fudhail Abdul Munir. "The role of the fuel-flame separator in stabilizing the flame of liquid fuel in a meso-scale combustor", AIP Publishing, 2022 Publication	1%
6	Muhammad Arief Nugroho, Vera Suryani. "AADC 3: Active-Active Distributed Controller	1%

with 3-in-1 Asynchronous Heartbeat Synchronization Method in Software-Defined Networks", 2021 9th International Conference on Information and Communication Technology (ICoICT), 2021

Publication

7	watermark.silverchair.com Internet Source	1 %
8	arxiv.org Internet Source	1 %
9	dehesa.unex.es:8443 Internet Source	<1 %
10	aquila.usm.edu Internet Source	<1 %
11	www.mdpi.com Internet Source	<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On