Software Defined Networks Security: Link Failure Analysis in SDN

M A H. Al-Junaidi, Mohd Nazri Bin Mohd Warip, R Badlishah Ahmad, Anuar Mat Safar, I Zunaidi, Wan Khairunizam, Z M Razlan, A B Shahriman.

1Universiti Malaysia Perlis (UniMAP), Kampus Pauh Putra, 02600 Arau, Perlis, Malaysia. 2Universiti Sultan Zainal Abidin. 3Faculty of Technology, University of Sunderland, St Peter's Campus, Sunderland, SR60DD, United Kingdom

Abstract. SDN is the new network technology. Although SDN still new, it is inseparable from the issue of a link failure. By using SDN, link failure scenario can be simulated and collection of data can be analysed. Before creating link failure scenario, several topologies were needed to show similarities with traditional network. There are many topologies that can be created by using SDN because SDN already provide virtual network devices. Each network device will be build inside virtual network topology. With use of desired topology then link failure scenario can be implemented. SDN also need a controller that acts a “brain” to the network. The controller will be provide a routing algorithm to control the behaviour of the network when one or more links become fail to connect with host or controller. One of suitable controller was Ryu which can adapt to any traffic handler and already provide application resemble to network behavior. All software components will be simulated inside Mininet emulator, one of realistic virtual network application.

1. Introduction

In order to comply with all of these criteria, networking protocols have evolved significantly over the last few decades. However, the way traditional networks set up the deployment of one protocol to realize these needs in the worldwide organization is quite challenging. Traditional network configuration is indeed a time-consuming and error-prone because many steps are needed for the network administrator to add or remove a single device. A traditional network that consists of a modem, router, switch and several clients which have an Ethernet connection. This configuration approach makes it much more complex for a network administrator to deploy a consistent set of policies. As a result, organizations are more likely to encounter some failure in network links. To resolve this issue on a traditional network, it is now the time to introduce a new viewpoint on network management which is called Software Defined Networking (SDN) [1]. In recent years, the growth of SDN has gathers the attention of the research community. Mostly because of the flexibility it presents in its model of centralized management, which facilitates the development of solutions according to demand of the network by client [2]. Currently, traditional network does not have a good solution to solve the link failure scenario as soon as possible. It takes time for network administrator to identify the problem before handling the situation in traditional network. The challenge for large data center are getting complex because of the public demand for internet services. In fact, large data center network are prone to have a serious network problem such as data link failure. Whenever the link failure scenario happens, network administrator has a hard time to identify or analyze the problem. The reason is because there are thousands of routers over the Internet and it also depends to the network topology. Thus,
it is difficult to overcome the problem during network failure in many arrangements of the various elements of a computer network. The objective is to study the performance of SDN in the case of link failure in terms of packet loss, throughput, end-to-end delay, and packet delivery ratio and how the link failure affects the overall performance and, to understand and analyse several related methods used to solve the link failure scenario in SDN and, to evaluate the overall performance of the selected methods in custom network topologies. The network topology will be designed in the simulation and the data will be analysed.

Network company's endeavoured to adapt to distributed situations by growing new specialized arrangements of routing protocols. Be that as it may, there is just restricted data on the execution of each, and reasonable execution correlations are not generally accessible. Recently, the designs of utilizing Mininet was not frequently exist in execution assessments of OpenFlow. The simulation apparatus show the reasonable undertaking of planning, building and testing for clients with down to word systems when growing true frameworks. This will enable framework fashioners to decide therightness and productivity of a plan before the framework is really conveyed. Another favourable position of test systems are they permit the assessment among different system measurements and approval instruments to acquire comes about that are not tentatively quantifiable on bigger geologically appropriated designs.

1.1. Mininet is an emulator platform
Using OF protocol, runs a collection of end-hosts, switches, routers and links on a single Linux kernel by using lightweight virtualization. Mininet components function as real network components [3].

1.2. Floodlight Controller
Floodlight Controller In [4] experiment, authors choose Floodlight to compare and analyse the routing convergence time of legacy network. Floodlight also been used in [5] for multipath routing. The GA and Dijkstra algorithm for solving the BCMPO problem has been load inside Floodlight controller. With the algorithm, Floodlight can choose routing paths automatically according to the whole network status scenario. Figure 1 show the structure of core module, which is responsible for transforming received OF packets into events. [6].

Figure 1. SDN architecture [11]
2. Motivations
SDN components proceeding with Mininet, the Mininet creating Software Defined Networks (SDN) components, alter, share SDN and Mininet with different networks and make cooperations. Selection. Controller Ryu is a component-based of SDN framework. Ryu needs to be installed so Mininet has a controller component to allow communication between devices. To begin ensure system has python installed, by executing python on the terminal, which would turn into the python shell, besides displaying the version.

2.1 Wireshark
Wireshark is a network analysis tools that will captures packets in real time and display the readable data. Other than that, Wireshark also includes filters, color-coding and other features that show network traffic and inspect individual packets. This section will discuss about the system flowchart, simulation parameter, topology design, and the routing algorithm used in the system.

2.2 OpenFlow
In this section, the OpenFlow-enabled network architecture for basic OpenFlow network topologies is implemented using Mininet. Mininet supports 5 built-in network topologies, namely, minimal, single, linear, tree and reversed topology [7]. The default topology is minimal topology which is predefined with one OpenFlow kernel switch connected to two hosts and OpenFlow reference controller, whereas number of switches and hosts can be changed for other topologies using the command-line interface (CLI). In this section will be focus only to three basic which are single, linear and tree topology. Table 1 shows the value of parameter and explanation inside CLI.

2.3 Network topology
Network topology is one of data centre topology that provides the efficient communication between the hosts. This custom topology has a capability for multipath routing. The topology used will be 4-pod topology as shown in Figure 2 The hierarchical layers of the network include core switch, aggregation switch, edge switch and host. A topology having 16 hosts is designed in Mininet using CLI command as below:

```
$ sudo mn --topo=custom,4.py --switch=ovsk --controller=remote
```

![Figure 2. OpenFlow-custom Topology, 4 pods having 16 H](image-url)
Each switch been renamed to show different between switch in each layer. Assume the controller at the top of the layer, core switch will become first layer contain of 4 switch start from 1001~1004. Both second layer (aggregation switch) and third layer (edge switch) each has 8 switch and will start from 2001~2008 and 3001~3008 respectively. Host also will be rename from default h1-h16 to h001~h016. For the host IP address also will be change from default 10.0.0.1~10.0.0.16 to 10.1.0.1~10.8.0.2. The links connection of all switches and hosts are shown in Figure 2.

2.4 Ryu Application Routing Algorithm
Ryu Application Routing Algorithm the most important thing in simulating SDN network is the controller. As been discussed in previous section, Ryu will be main controller used to run the simulation. This section will cover about routing algorithm in the Ryu controller. The selected routing algorithm will determine the behavior of host or switches when a link failure scenario happens. Ryu controller has already provides many applications to be run inside controller. All Ryu applications have their own function and behavior for switches and also hosts so that it can meet user requirements. In the next section, each application will be introduced to be used for the desired topologies.

![Figure 3. Ryu Application]

3. Simulating SDN on mininet and floodlight controller.
A standout amongst the most important communication protocols is the Internet Protocol Suite, TCP/IP, utilized for correspondence over the web and comparative systems. This systems administration shows an availability between two hubs in the system, by determining how information ought to be transmitted, designed, tended to, directed and got at the goal. The TCP/IP framework architecture was produced after the OSI reasonable model on the grounds that the internet was executed utilizing the favored TCP/IP suite amid the spread of the World Wide Web [8]. In addition, most sellers and large programming consumers favored using this model instead of OSI because of the way that TCP/IP can meet worldwide benchmarks in which the OSI model couldn't give. Table 1 demonstrates a next to each other examination of the OSI, TCP/IP, and SDN models.
The floodlight [9] is controller of an OpenFlow for big company networks in light of programming of Java language and dispersed under the Apache permit. The main undertaking began as of the Beacon controller [10] and now it is bolstered by an engineers’ group and furthermore by Big-Switch Networks, a jump up that products business equipment controller.

In this section, a comparative performance analysis of a topology is done based on results obtain after execution of networks. No link failure simulation will be done in the basic topologies because result give almost guarantee failure. Thus, the basic topology will be simulated with fully working link. The performance analysis is done by comparing all network topologies on the basis of bandwidth utilization, packet transmission rate, end-to-end delay (time required to transmit packet from source node to destination node) and throughput. Here, performance analysis for topologies discussed for a topological designed network of eight hosts. Bandwidth utilization for topologies is tabulated in Table 1 and is shown in Figure 5.
4. Results and discussions.

This section will discuss the analysis of link failure scenario in topology. Topology with 4 pods will be simulated in Mininet environment. Several scenarios have been created to satisfy the link failure scenario so that the performance can be clearly measured. The performance analysis is done by comparing the analysis with and without link failure based on throughput, end-to-end delay, packet loss rate, and packets delivery ratio. For topology, several switches will be created to be stopped thus make all interconnected links become fail so it can match with the scenario that already been discuss in the previous chapter. Topology, 10Mbps of bandwidth will be set in the network. There are 3 selected aggregation switches from the total number of 8 switches in layer 2 will be turned off and also for the 2 core switches from the total number of 4 core switches. Table 1 shows the collected bandwidth using Iperf tool for each switch scenario with and without failure.

| Scenario             | Average Utilized Bandwidth (Mbps) | Overall Performance (%) |
|----------------------|-----------------------------------|-------------------------|
| All switch start     | 8.95                              | 89.5%                   |
| 1 aggregation switch down | 8.31                              | 83.1%                   |
| 2 aggregation switch down | 7.82                              | 78.2%                   |
| 3 aggregation switch down | 7.66                              | 76.6%                   |
| 1 core switch down   | 7.58                              | 75.8%                   |
| 2 core switch down   | 7.99                              | 79.9%                   |

Table 1. Comparison of Bandwidth in Switch Failure Scenario

![Figure 6. Bandwidth Chart for Topology](image)
Based on the result in Table 1, core switches will give more influence for the decrease in bandwidth compared to aggregation switches. It is because of the core switches placed at the first of layer and has more influence links connected with other switches during new routing data path algorithm. In Figure 6 clearly, show that more switches are turned off will incur the decrease of bandwidth utilization in the network topology. The comparison of the number of switches down with performance can be seen in Figure 6. It is because aggregation switch connects closer to the host so the connection possibility between two hosts will reduce. The more aggregation switches down; the connection possibility reduces more. So it affects the connection bandwidth in the entire network. The same can be applied with the core switch that also has been also down. Thus, the more core switch is down, the bandwidth will also decrease.

| No of Transmitted Packet | All switch start | 1 aggregation switch down | 2 aggregation switch down |
|--------------------------|------------------|--------------------------|--------------------------|
| 100                      | 90               | 83.1                     | 78.2                     |
| Throughput (kbps)        |                  |                          |                          |
| 3 aggregation switch down| 77               | 76                       | 80                       |
Throughput for topology will be calculated based on bandwidth and RTT. The result can be referred to in Table 2. Total throughput when there are none switch failures are 90 kbps. By referring to Table 2, the more switch been turned off the less outcome of throughput value. The value of throughput will affect accordingly to the value of bandwidth simulation result. The value of 1 core switch down versus 1 aggregation switch down clearly shown that core switch gives more effect to the network when been turned off. Figure 7 shows the throughput values in graphically.

### Table 3. Packet Delivery Ratio Chart for Topology

| No of Transmitted Packet | All switch start | 1 aggregation switch down | 2 aggregation switch down | 3 aggregation switch down |
|--------------------------|------------------|---------------------------|---------------------------|---------------------------|
| 100                      | 100%             | 31%                       | 18%                       | 11%                       |
| core switch down         | core switch down | core switch down         |                           |                           |
|                          | 60%              | 15%                       |                           |                           |
The packet delivery ratio is the ratio of packets successfully received by the host to the total transmitted packets. The total number of successful packets received and the total number of packets transmitted can be referred to in ping statistics from the ping test in Mininet environment. Table 3 shows the packet delivery ratio in topology based on the simulation scenario. By referring to Figure 8, the packets delivery ratio will decrease when there are more switches turned off. The highest PDR is for 1 core switch in layer 1 of topology which is 60%. The lowest PLR is for 3 aggregation switch down which deliver only 8% from 100 packets been transmitted to the host 016.

5. Conclusions
The purpose of the performance analysis of SDN in link-failure scenario study has been achieved with the help of research from journals and papers by previous researchers. As the advanced technology approach, SDN is the trend that will replace existing traditional networks, especially in data center. Study of SDN with link failure scenario will greatly improve aspects in computer networking. Although SDN still new to the computer networking world, with applying this SDN proved it can offer many data analysis in long terms for the future. There are so many methods to implement link failure scenario in SDN. Each method has already been compared and the suitable method has been successfully used to simulate link failure scenario. By using the chosen method, the study of performance analysis in case of link failure in term of throughput, end-to-end delay, packet loss rate and packet delivery has been simulate and the data has been analyze and discuss throughout this research project. There are five topology that has been discuss in this research analysis which are single, linear, tree, triangle and fat tree topology. Each of topology can be used to simulate link failure scenario but only several topologies will give a detail result of network performance. All topology has been designed and data from each of it has been discussed in previous chapter. Hence, the research performance analysis objective has been fully achieved.

6. REFERENCES
[1] “Traditional VS Software Defined Networking.” [Online]. Available:
http://www.mavenspire.com/blog/traditional-vs.-software-defined-whats-the-differences.

[2] Open Networking Foundation. (2013). SDN Architecture Overview. Onf, 1–5.

[3] Kaur, K., Singh, J., & Ghumman, N. S. (2014). Mininet as Software Defined Networking Testing Platform. International Conference on Communication, Computing & Systems (ICCCS–2014), 3–6.

[4] Zhang, H., & Yan, J. (2015). Performance of SDN Routing in Comparison with Legacy Routing Protocols. Proceedings - 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2015, 491–494.

[5] Liu, Y., Pan, Y., Yang, M., Wang, W., Fang, C., & Jiang, R. (2015). The Multi-Path Routing Problem in the Software Defined Network, 250–254.

[6] Isaia, P., & Guan, L. (2016). Performance benchmarking of SDN experimental platforms. IEEE NETSOFT 2016 - 2016 IEEE NetSoft Conference and Workshops: Software-Defined Infrastructure for Networks, Clouds, IoT and Services, 116–120. https://doi.org/10.1109/NETSOFT.2016.7502456

[7] Kaur, K., Singh, J., & Ghumman, N. S. (2014). Mininet as Software Defined Networking Testing Platform. International Conference on Communication, Computing & Systems (ICCCS–2014), 3–6.

[8] Henderson, T. R., & Riley, G. F. (2006). Network Simulations with the ns-3 Simulator. Proc. Sigcomm, 527. https://doi.org/10.1.1.415.6550

[9] Pal, C., Veena, S., Rustagi, R. P., & Murthy, K. N. B. (2014). Implementation of simplified custom topology framework in Mininet. 2014 Asia-Pacific Conference on Computer Aided System Engineering, APCASE 2014, 48–53. https://doi.org/10.1109/APCASE.2014.6924470

[10] S. Liao, X. Hong, C. Wu, B. Wang and M. Jiang, "Prototype for customized multicast services in software defined networks," 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, 2014, pp. 315-320.