A Novel Fault-Tolerance Mechanism in Software Defined Network (SDN) for Dynamic Traffic Engineering

Bi-Lynn Ong
Faculty of Electronic Engineering Technology Universiti Malaysia Perlis
Perlis, Malaysia
drlynn@unimap.edu.my

Hasnah Ahmad
Faculty of Electronic Engineering Technology Universiti Malaysia Perlis
Perlis, Malaysia
hasnahahmad@unimap.edu.my

Naimah Yaakob
Faculty of Electronic Engineering Technology Universiti Malaysia Perlis
Perlis, Malaysia
naimahyaakob@unimap.edu.my

Amiza Amir
Faculty of Electronic Engineering Technology Universiti Malaysia Perlis
Perlis, Malaysia
amizaamir@unimap.edu.my

Noor Fazreen Bakar
Faculty of Electronic Engineering Technology Universiti Malaysia Perlis
Perlis, Malaysia
fareen@unimap.edu.my

Mohamed Elshaikh Eloboid
Faculty of Electronic Engineering Technology Universiti Malaysia Perlis
Perlis, Malaysia
elshaikh@unimap.edu.my

***Abstract*—Traditionally, the network fault tolerance mechanism was insufficient to handle network traffic during network failure. The network administration needs to manage the fault tolerance mechanism manually which causes long duration of distortion and disconnection. Additionally, network administration has to monitor the switches and routers periodically. Software Defined Network (SDN) is designed to handle the fault tolerance issue during network failure. A controller in SDN has the intelligent to direct network traffic to available path during congestion. The main advantage of this controller is that this controller can work with the algorithm programmed in directing network traffic engineering. This aim of this research is to implement a fault tolerance mechanism namely topology discovery during congestion. The topology discovery algorithm is able to calculate the neighbor path and suggest an available path during heavy traffic. This topology discovery algorithm keeps calculating and thus can fully utilize available links. Having implemented this topology discovery in the SDN controller, it is believed that network can recover from network failure immediately and thus provides smooth connection to the network community.**

Keywords—SDN, fault-tolerance, topology discovery

I. INTRODUCTION

Many previous works have explored many possibilities on how new functionalities SDN can deliver. However, it has been little investigation on how to deal with an age-old yet common problem in computer networks: network failures. Network failure, such as network device failures of link disconnections, can disturb normal traffic forwarding even if there is an alternate path in the network. However, there are insufficient research and experience in building fault-tolerant mechanisms.

SDN-based network is aimed to exploree ways to cope with link failures in SDN-based networks. Hhowever, the proposed mechanisms do not completely recover from faults. The network link does recover from failures but is restricted to multi-rooted tree topologies. There are three fault domains in SDN-based networks. (1) Data plane, where a switch or link fails, (2) control plane, where the connection between the controller and switch fails, and (3) controller, where the controller machine fails [1,2]. This research plans to address these three fault domains. This research is initially focused on building a SDN-based fault-tolerant mechanisms for data plane faults. The main challenges on building a fault-tolerant mechanism based on SDN are:

- Current mechanism from old-aged networks do not work with SDN networks.

  Running current mechanism that usually operate on old-aged networks, such as rapid spanning tree protocol (RSTP), together with a SDN controller algorithms do not work well without proper planning or large modification. The reason is because two

control planes coexist. However, these two control planes are decoupled from each other. There are no interaction or information exchange between these two control planes.

• Although SDN-based network has many advantages, pure SDN-based mechanisms are not advanced.

The central controller in SDN-based network can collect routing path from routers and gather the information of the networks. With these routing path and network information, the controller can run centralized algorithms that are more efficient as compared to distributed algorithms that have limited information during link failures. Nevertheless, OpenFlow SDN is a flow-based mechanism where every first packet has to obtain information from the controller for forwarding mechanism. This is inefficient if the number of nodes increase. The reason is because the number of flows can grow exponentially. As a result, the number of forwarding mechanisms that need to be updated to recover from a failure can be tremendous and therefore increases the recovery time.

The aims to reduce the fault-tolerance and recovery time is therefore become the main objective in this research. The the later part of this research, the dynamic traffic engineering for managing the fault tolerance mechanism in SND is discussed in more details.

II. BACKGROUND

Dynamic traffic engineering is a mechanism that the Internet network engineering dealing with the problem of performance investigation and performance enhancement of operational Internet networks. Network traffic engineering consists of the mathematical calculation and algorithms that aim to measure, characterize, model, and control of Internet traffic. Optimizing the performance of an operational network, at both network traffic and resource levels, are main objectives of Internet network engineering. This is achieved by calculating traffic performance matrix, while utilizing network reliably and quality. Traffic performance matrix includes packet delay, packet jitter, packet loss ratio and throughput. A main aim of network Internet traffic engineering is to provide reliable network performance. This can be achieved by designing algorithms that follow rules, procedures and protocols. As all these are followed, this can avoid faults, errors, failures distortion and disconnection within the network infrastructure.

Fault tolerance mechanism enables the network to continue connected properly in the case of the network failure in a particular link. Networks are expected to be connected without any disconnection, even there are link failures happen between the network connections. In the traditional network, many networks have employed fault tolerance mechanisms that allow routers and switches to rapidly respond to link failures, recovering connectivity in a very short duration. At the same time, the Internet network is expected to perform much more than provide connectivity. The Internet network also must provide rigorous security and performance guarantees, even while recovering from failures.

Software-defined network (SDN) is a method for Internet network that allows network administrators to manage network services through lower-level functionality [3,4]. The traditional network does not support the dynamic, scalable computing and storage needs of more modern computing environments such as data centers [5]. SDN is designed as an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications [6,7]. This is done by decoupling or disassociating the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). This architecture enables the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. SDN mechanism is to allow network designers to construct networks that meet their end-to-end requirements, rather than forcing them to stitch together existing protocols, each with their own capabilities, features, and limitations [3,4]. In previous works, there are many researchers work on implementing failure discovery mechanism in SDN controllers. Unfortunately, network engineers are lack of experiences in designing the failure recovery rules and protocols. It is a challenge for the network engineers to come out with an algorithm that can automatically consolidating these mechanisms. Currently, network engineers are working together with the aim to come out with an algorithm that can handle the network failure. Network engineers need to have skills to enable the implementation of fault tolerance mechanism in SDN controller. Additionally, other than the fault tolerance mechanism that works to avoid network failure, there is a need to consider the security and performance issues. Therefore, network engineers are working closed together to design a sigle program that able to handle all these functions.

III. SDN FOR DYNAMIC TRAFFIC ENGINEERING

As discussed in the literature review, SDN is setting up of control plane, data plane and application plane. SDN separates the control plane and data plane of networking devices and introduces a well-defined interface, the OpenFlow protocol [8,9], between the two planes. The SDN architecture is showed in the Fig. 1 below. The OpenFlow between the control plane and the data plane acts as a controller and performs the intelligence which is to control the data flow. OpenFlow is out of networking devices and it is placed in a centralized server called controller. This OpenFlow provides centralized control over a network.

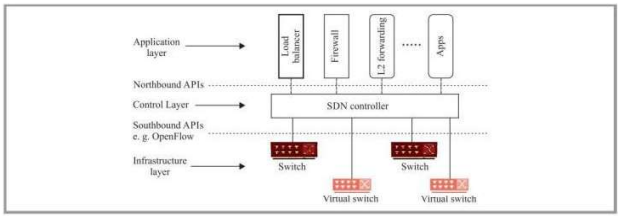

Fig. 1. SDN Architecture [8.9]

The OpenFlow controller is an operating system for the SDN network. OpenFlow performs the control applications and functions, for example routing protocols and layer 2 forwarding mechanism [10,11]. This mechanism performs the underlying network infrastructure. Therefore, OpenFlow enables the applications and network functions to treat the network as a logical entity. There are many versions of OpenFlow available for free. The most widely used SDN OpenFlow controller is the OpenFlow v.1.3 protocol. OpenFlow v.1.3 enables the controller to control the OpenFlow switches. The OpenFlow v.1.3 switches contain many flow tables and a secure OpenFlow transmission. The flow tables and a secure OpenFlow transmission are used for packet lookup and are used to forward the packets. The OpenFlow transmission is a conception layer. A secure link is establishedd between switches and the controller via the OpenFlow protocol. This transmission performs the underlying switch hardware. There are newly OpenFlow version which is OpenLow v.1.5. For OpenFlow v.1.5, a switch can have one or more OpenFlow transmission that are linked together to multiple ccontrollers. In general, SDN is a flow-based control strategy. With the OpenFlow, a controller can be defined as how the switches should performs the flows in a SDN when a source node sends.

Having OpenFlow acts as an controller, the fault tolerance mechanism is implemented in the OpenFlow protocol.The fault tolerance mechanism is able to detect the possibility if a disconnection may happen. The mechanism keeps on calculating the next node information until a fault information is detected. The mechanism reacts accordingly when fault information is detection as shown in Fig 2. below.

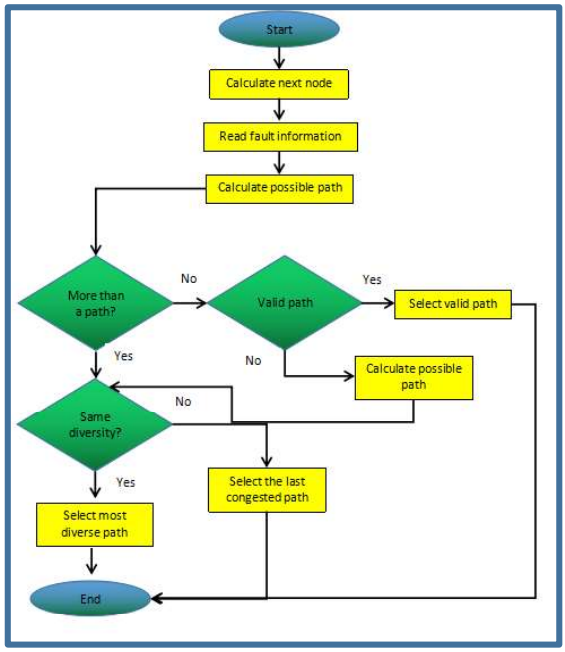

Fig. 2. Topology Discovery Fault Tolerance Mechanism over SDN

As shown in Fig. 2 above, this is the topology discovery mechanism that is able to identify available path during congestion. This topology discovery mechanism is implemented in the SND central controller. This algorithm is kept on running with the aim to suggest neighbour path to the network traffic.

Periodically, the topology discovery mechanism calculates the neighbour nodes. When there are congestion or fault information, this algorithm starts to calculate possible path from neighbour links. There are two conditions happen

a. If there is only one path available
   i. When there is only one path available, the algorithm calculates is this only path is valid.
   ii. If this is a valid path, the controller will select this valid path to forward network traffic. Although congestion happen, this is the only path available.

b. If this is not a valid path, the controller will calculate any other possible path. Then, this algorithm directs to the possibilities of multiple path.
   i. If there are more than a path available
   The controller calculates if paths are in the same diversity
   ii. If the paths are in the same diversity, the controller selects the most diverse path.
   If the paths are not in the same diversity, the controller selects the last congested path.

Having perform this topology discovery mechanism, the SND controller can identify the possible available links with neighbour nodes. Link failure can be reduced with this topology discovery mechanism in SDN controller.

This topology discovery fault tolerance mechanism is implemented over a simulation tool. The

results have shown a significant improvement. The next session presents the preliminary result of SDN fault mechanism over the traditional network.

IV RESULT AND DISCUSSION

Fig. 3 shows the preliminary result of SDN fault tolerance mechanism over the traditional network. The simulation is performed with different number of nodes.

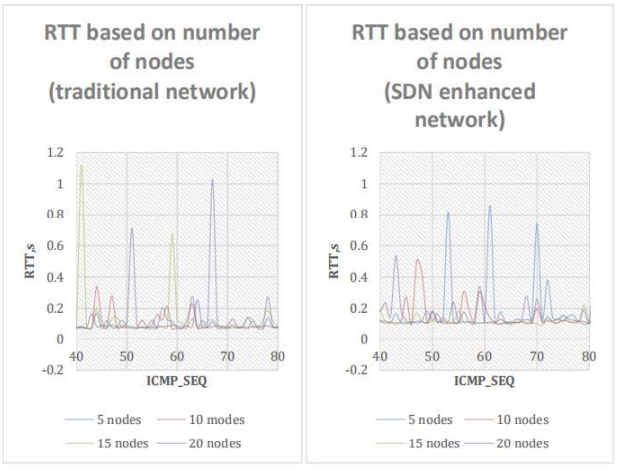

Fig. 3. Preliminary result of SDN over traditional network (RTT)

As shown in Fig. 3 above, it shows that SDN enhanced protocol performs better as compare to traditional network. Simulations are performed for 5, 10, 15 and 10 nodes. SDN enhanced protocol shows lower round trip time (RTT) compare to traditional network. As the number of nodes increase, the network needs more duration for RTT.

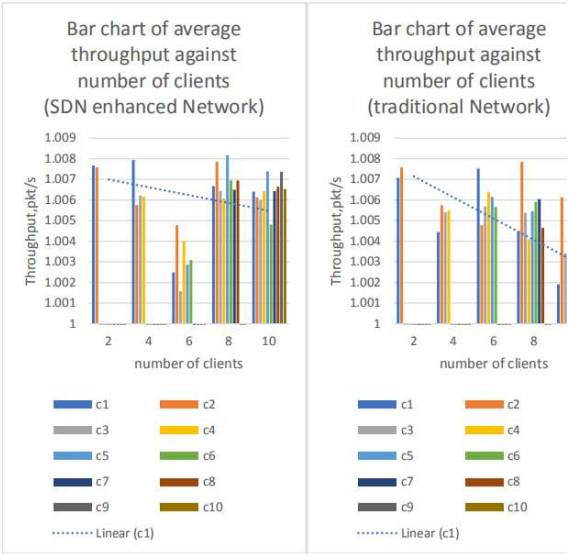

Fig. 4. Preliminary result of SDN over traditional network (Throughput)

Fig. 4. above shows the result for throughput between SDN and traditional network. Simulations are performed for 2, 4, 6, 8 and 10 clients. The results also show that SDN performs better as compare to traditional networks.

From Fig.4, it can be observed that as the number of client increases, the throughput decreases. The reason is because as the number of client increases, the bandwidth is distributed to many clients. By implementing SDN enhanced protocol, throughput is significantly increased.

IV CONCLUSION

The novelty of this research is the topology discovery mechanism that is to be implemented over the SDN architecture. The topology discovery mechanism has the ability to detect the link failure happen and a new links for the network connectivity. Having performed this mechanism, the disconnection caused by link failure is solved by implementing a new route. The implementation of this method will significantly reduce the end-to-end delay between connected nodes which can also minimize response time which could improve the overall performance.

REFERENCES

[1] H. Kim and N. Feamster, "Improving network management with software defined networking," Communications Magazine, IEEE, vol. 51, no. 2, pp. 114–119, February 2013.

[2] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," Communications Magazine, IEEE, vol. 51, no. 2, pp. 136–141, February 2013.

[3] Shailendra Mishra1 and Mohammed Abdul Rahman AlShehri, "Software Defined Networking: Research Issues, Challenges and Opportunities", Indian Journal of Science and Technology, Vol 10(29), August 2017.

[4] Tao Hu, Zehua Guo, Peng Yi, Thar Baker, Julong Lan, "Multi-controller Based Software-Defined Networking: A Survey", IEEE Access 2017

[5] T. Koponen et al, "Onix: A Distributed Control Platform for Largescale Production Networks," in Proceedings USENIX, ser. OSDI'10, Vancouver, BC, Canada, 2010, pp. 1–6.

[6] R. Ahmed and R. Boutaba, "Design considerations for managing wide area software defined networks," Communications Magazine, IEEE, vol. 52, no. 7, pp. 116–123, July 2014.

[7] S. Jain et al, "B4: Experience with a Globally-deployed Software Defined Wan," SIGCOMM Comput. Commun. Rev., vol. 43, no. 4, pp. 3–14, Aug. 2013.

[8] Arpita Prajapati, Achyut Sakadasariya, Jitisha Patel, "Software Defined Network: Future of Networking", In the Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018), 2018.

[9] D. Tuncer, M. Charalambides, H. El-Ezhabi, and G. Pavlou, "A Hybrid Management Substrate Structure for Adaptive Network Resource Management," in Proceedings of ManFI'14, Krakow, Poland, May 2014, pp. 1–7.

[10] Open Networking Foundation. Software-defined networking: The new norm for networks. ONF White paper, 2012.