International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

QoS-aware Traffic Management in Software Defined Networking

May Thu Zar Win[a]*, Khin Than Mya[b]

[a,b]*University of Computer Studies, Maubin, Myanmar*

[a]*Email: maythuzarwin@ucsy.edu.mm*

[b]*Email: khinthanmya@ucsy.edu.mm*

Abstract

Software defined networking (SDN) provides effective traffic management solution by separating control and data planes, global centralization control, and being programmable. And, the traditional shortest path routing cannot provide effective traffic engineering because it only aware shortest path. The constraint-aware routing is more efficient than the traditional shortest path routing, however, it needed to estimate constraints such as link capacity, delay, jitter, and so on and it cannot guarantee the future traffic demands. This paper proposed QoS-aware traffic management method in SDN to guarantee the QoS-aware traffic by selecting the optimal path based on the estimated constraints. First, the proposed traffic management method categorized traffic classes: QoS-aware traffic and non QoS-aware traffic classes. Then, the proposed method estimated the QoS parameters and calculated the optimal path based on the estimated parameters. Finally, the QoS-aware traffic routed with the optimal path and non QoS-aware traffic simply routed through the shortest path. The proposed method is validated by using network emulator, Mininet and SDN controller, ONOS. The experiment results of throughput and packet loss show that our proposed method outperformed the other two traffic management methods.

Keywords: Traffic Management; QoS; SDN; ONOS; Link Capacity.

---

* Corresponding author.

1. Introduction

Effective network management solutions are required not only to harmonize the dramatic growth of transferring information and requirements of applications but also to provide the best services to users and improve network performance. The legacy networks mostly implement in dedicated appliances and hardware that lead to limited innovations for both management and configuration aspects. The networks also create multi-vendor equipment. Therefore, network administrators need to have a wide knowledge of all devices from different vendors because different vendors have different syntaxes and commands. Moreover, the traditional networks cannot provide the complexity of control protocols, complex traffic engineering (TE) tasks, and interconnecting of a huge number of smart devices [1]. Software Defined Networking (SDN) is an architecture that overcomes the above issues of the traditional networks by taking advantage of global centralization control, decouples of the control and data planes, and enabling innovation through the network programmability [2]. Although SDN has presented to enable network innovation, and to program new applications according to the user requirements, there have still problems to encounter in traffic management, Quality of Service (QoS) based routing, and security. This paper focuses on the traffic management methods in SDN. The traditional Shortest Path First (SPF) routing is efficient, fast, and simple, but it not only can cause traffic congestion but also cannot guarantee future traffic demands [3]. SPF does not consider whether the incoming traffic is QoS-aware traffic or not. It only takes account of the minimum hop-count and does not achieve load balancing. The constrained-based routing algorithms alleviate congestion and improve the resource utilization. However, the calculation time and the flow rule installation time become non-negligible factors. In SDN architecture, OpenFlow switches can support flow management more effective because of their multiple flow table pipelines. The authors in [4, 5] proposed traffic management methods which considered a trade-off between the load balance and latency for multiple flow tables usage at switch level and controller level. For a wider range of Internet applications, routing algorithms based on the delay and link utilization have become more important to fulfill the user requirements. The authors in [6, 7] tried to solve the problem of setting up the bandwidth guarantee tunnels in networks. A simple solution is proposed in [6], where firstly they prune all the links with insufficient bandwidth, and then the solution chooses a path with the smallest delay path. The authors in [7] proposed the algorithm that calculates the shortest disjoint paths, defines critical links for bottleneck traffics, and avoids the links for future demands. Deng et.al [8] mentioned one of the traffic management solutions that guaranteed multiple QoS requirements of high-priority IoT applications and selected the better routing paths by adapting the current network status. The authors in [9] proposed a routing scheme for SDN-based cloud data centers. They classified applications as class1, 2, and 3. Then, they used different routing algorithm for different classes. Jarschel and his colleagues [10] proposed a DPI-based application-aware path selection method for YouTube video streaming. Firstly, they defined the threshold for a YouTube streaming application and then monitored buffered playtime and stalling. When currently buffered playtime reaches below the threshold, their method calculated the least load path and reroute the flow through the least congested path. Their method only applied YouTube streaming and required an additional machine to continuously monitor the buffered playtime. Different application traffic needs different routing based on their QoS requirements. Therefore, we proposed QoS-aware traffic management method which performed two different routing. For QoS-aware application traffic, our proposed method used constraints-aware routing. These constraints are minimum delay and maximum link utilization. When the non

QoS-aware application traffic entered the network, our proposed method simply routed through the minimum hop-count path. The main objective of our proposed method is to choose the optimal path for QoS-aware application traffic. The remainder of the paper is organized as follows. Section 2 presents the background theory of SDN. The detailed process of proposed QoS-aware traffic management method is mentioned in section 3. Section 4 presents the implementation of the proposed method and discusses the experimental methods and results. Finally, section 5 concludes the paper.

2. Software Defined Networking Architecture

Software Defined Networking (SDN) is an architecture that decouples control and data function from the network devices such as switches and routers. This architecture also has global centralization control and enabling innovation through the network programmability. Figure 1 describes the difference between the traditional internets and SDN architecture. This figure shows clearly the integration of control and data functions in traditional networks and the separation of control and data functions in SDN.

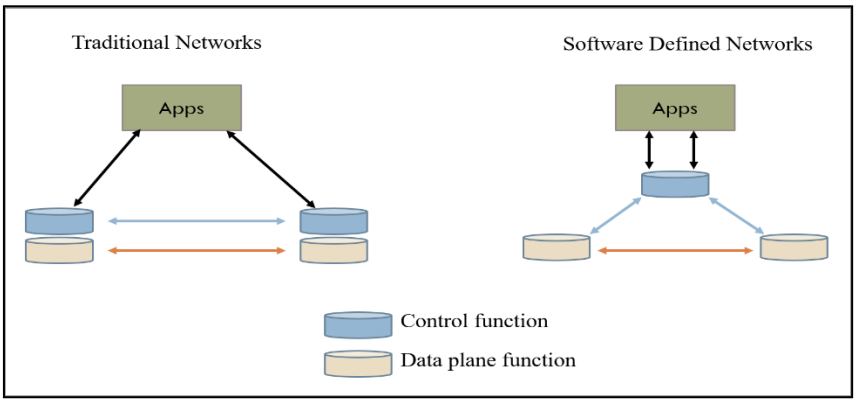


Figure 1: Traditional Network Architecture Versus SDN Architecture

SDN architecture involves the following three main layers: the application layer, control layer, and data plane layer as shown in figure 2. The SDN applications are programmed to support all kinds of network services such as traffic engineering, firewall, routing, load balancing, etc. The control layer is a core layer of the SDN architecture that extracts the data plane layer information and communicates to the application layer with an abstract view of the network topology, including statistics and events. The application and control layers communicate by using northbound APIs. The data plane layer consists of network nodes which can forward and process the data path. Communications between the data plane and control layers use a standardized protocol, called OpenFlow. The SDN Controller defines the data flows that take place in the SDN Data Plane.

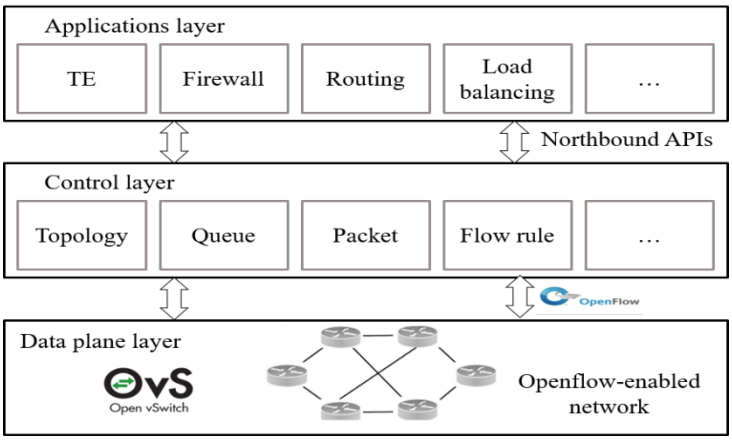


Figure 2: Software Defined Networks Architecture

3. The Proposed QoS-aware Traffic Management Method

The proposed system contains the following four main modules: topology discovery, measurement module, classification module, and management module. Figure 3 depicts the overall design of the proposed system. Topology discovery module discovers the network nodes and constructs the network topology. The others modules used the topology information to measure the QoS parameters and manage the network. The classification module classifies the incoming traffic whether the QoS-aware traffic or not.

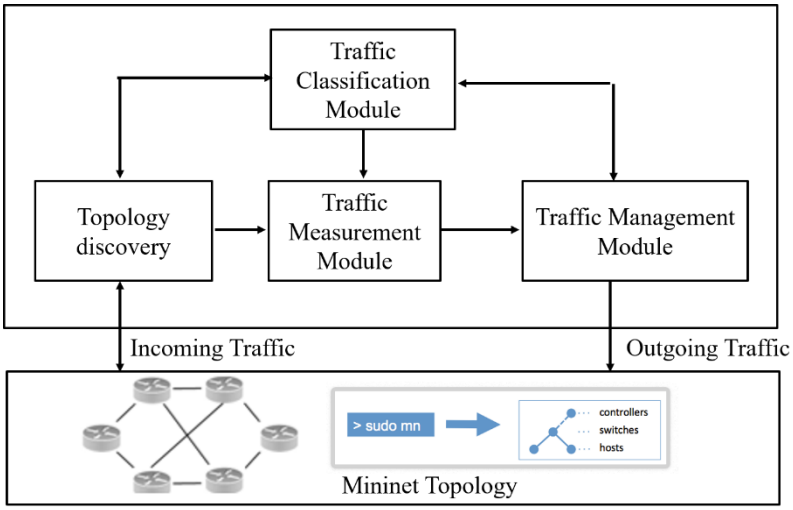


Figure 3: The Proposed System Design

The measurement module calculates the link utilization and link delay by using topology information and port statistics information from the controller. In the traffic measurement module, we estimate link utilization and link delay. The link utilization of the link is obtained by subtracting link load from link capacity as shown in equation (1):

$$Link\ Utilization = Link\ Capacity - Link\ Load \qquad (1)$$

Link capacity of the link is the configured link bandwidth. We can get the link load of the link by adding the port statistics information (port statistics of transmitted and received bytes counter) of source switch and destination switch through the following equation (2):

$$Link\ Load = Src\_Port\_Stats\_ of\_Transmitted\_Bytes\_Conut + Dst\_Port\_Stats\_of\_Received\_Bytes\_Count \quad (2)$$

After calculating each link load along a given path, the link utilization of the link can be derived as equation (3):

$$Link\ Utilization_{src \rightarrow dst} = min_{link_i \in Path}\ Link\ Utilization_{link_i} \quad (3)$$

The link utilization of the path is the minimum link utilization of a link along the given path, where $link_i \in Path$. In this work, we used maximum link utilization of a path as the QoS parameter constraint. To estimate the link delay, we send probe packets from source to destination switch of the link. Therefore, the link delay of a link estimated as the equation (4).

$$T_{link\_delay} = T_{total} - (T_{controller\_to\_Srcswitch} + T_{controller\_to\_Dstswitch}) \quad (4)$$

$$T_{total} = T_{controller\_to\_Srcswitch} + T_{Srcswitch\_to\_DstSwitch} + T_{Dstswitch\_to\_Controller} \quad (5)$$

T_{total} is the time duration to send a probe packet from the controller to source switch plus source switch to destination switch plus destination switch to the controller. We got one-way delay by subtracting the delay time of the controller to source switch, and controller to destination switch from the total time, T_{total}. The total delay for the path is the sum of each link delay along the path through the network as in equation (6):

$$T_{path\_delay} = \sum_{link\ \epsilon\ Path} link_i \quad (6)$$

where, T_{path_delay} is the total delay for a path and $link_i$ is the i^{th} link of the link delay and that link include in that path, $link\ \epsilon\ Path$.

Figure 4 describes the overall process of our proposed method. As illustrated in figure 4, when the classified traffic is entered the network, our proposed method first check whether the QoS-aware traffic or not. If the classified traffic is the QoS-aware traffic, the method extracted all possible paths between source and destination switches. Then, it calculated the link utilization for all the extracted paths and selected the maximum link utilization paths. After that, it also calculated the link delay for the selected maximum link utilization paths. And, our proposed method selected an optimal path which has maximum link utilization and minimum link delay path. Finally, it installed flow entries to the switches along the optimal path and forwarded the QoS-aware traffic through the optimal path. If the classified traffic is non QoS-aware traffic, the method calculated the minimum hop-count paths and selected the shortest path. Then, it installed flow rules and forwarded the incoming traffic to the destination.

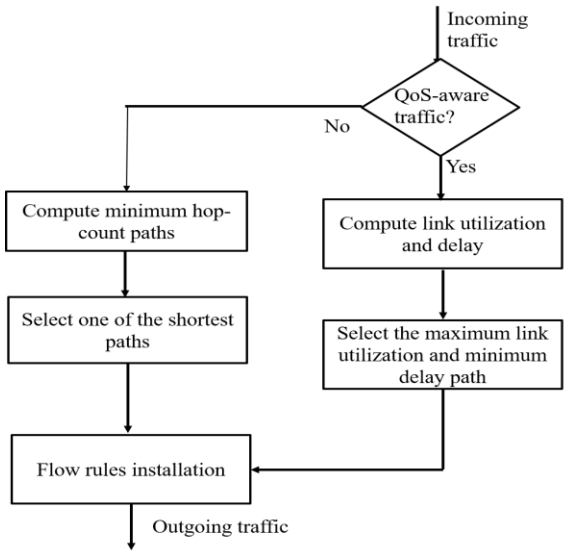


Figure 4: The Overall Process of the Proposed System

4. Implementation and Evaluation of the Proposed System

In order to assess the performance of the our proposed QoS-aware traffic management method, the experimental testbed has to be designed. This work applied ONOS [11] controller as the SDN controller and Mininet [12] as the network emulator. As the proposed method is implemented by using ONOS controller, it run in testbed topologies to compare and evaluate the results with other traffic management methods. Figure 5 illustrates the logical testbed design of this work.

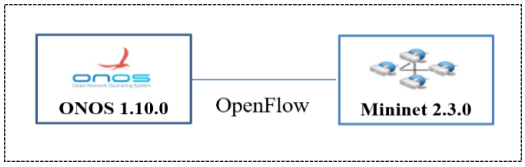


Figure 5: Logical Testbed Design of Proposed System

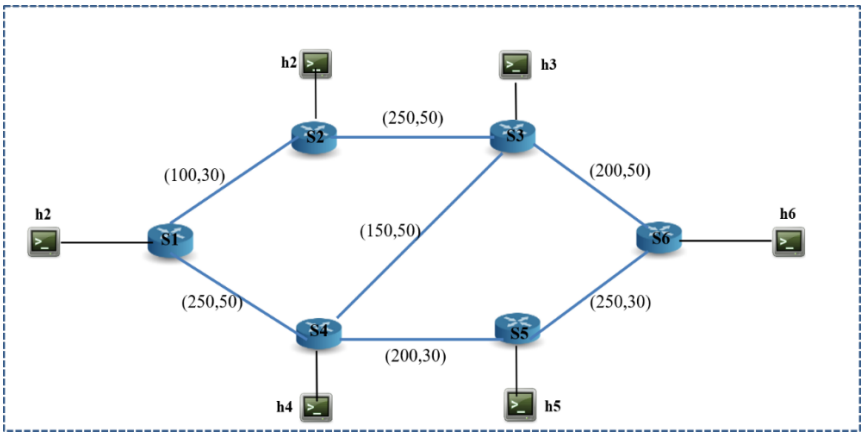


Figure 6: Physical Test Topology

In figure 5, ONOS is connected to the Mininet via OpenFlow protocol. All the switches in the Mininet topology are logically connected to the controller. Figure 6 depicts the physical test topology. In the test topology, there are six switches and hosts. The two numbers on each link denotes the configured link bandwidth (Mbps) and link delay (ms). Table 1 depicts the requirement and specifications of hardware and software tools that are used in the experimental testbed.

Table 1: Hardware and Software Tools that are Used in Experimental Testbed

Hardware		**Software**	
Name	**Specifications**	**Name**	**Version**
CPU	Core TM i5-4210U CPU @ 2.40GHz	ONOS	1.10.0 (Kingfisher)
RAM	8.00 GB	Mininet Emulator	2.2.1
Operating System	Linux 16.04 LTS	Open vSwitch	2.9.2
Number of PCs	2	OpenFlow Protocol	Version 1.3

4.1 Experimental Methods

To highlight the outcome of the proposed QoS-aware traffic management method, the following three methods are compared with two different scenarios.

(1) QoS-aware Traffic Management Method

The proposed QoS-aware traffic management method considered two different routing methods. When QoS-aware application traffic entered the network, the proposed method routed this traffic through the path which has maximum link utilization and minimum delay. If the incoming traffic is non QoS-aware application traffic, our proposed method routed it through the shortest path.

(2) Link Utilization-aware Routing

In this work, we used one of the types of constrained-aware routing, so called link utilization-aware routing. When the incoming traffic entered the network, the link utilization-aware routing firstly estimated the link utilization of all possible paths between source and destination switches and selected the maximum link utilization path. Then, it routed the traffic through the selected path.

(3) Shortest Path Routing

Shortest path routing is fast and simple. Whenever the incoming traffic entered the network, this routing simply forwarded the traffic through the shortest path.

4.2 Experimental Scenarios

There are two different experimental scenarios that have been carried out in this work. These scenarios are

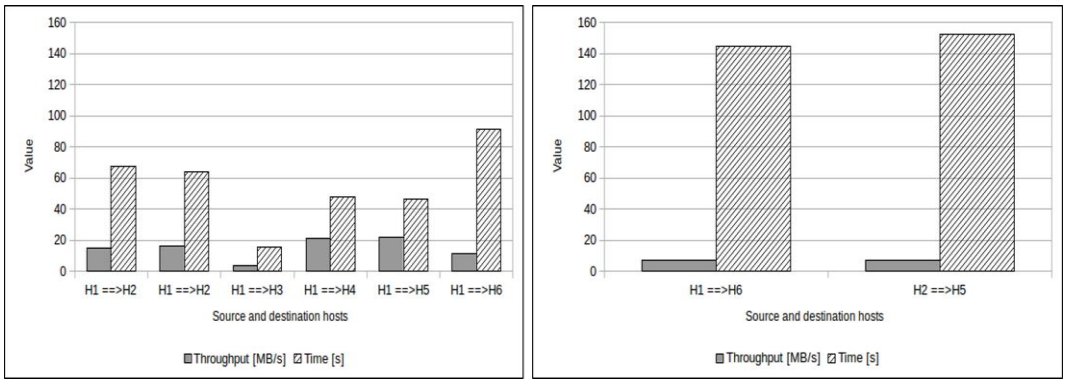
conducted according to table 1 and tested with figure 6, physical test topology. The experimental results calculated based on the average results of five running time.

(1) Scenario 1: Analyzing Methods by Transferring Files

In scenario 1, experiments have carried out according to parameters settings of table 2 by using test topology, figure 6. This scenario aims to analyze the three methods by file transferring and present how the three methods handled with the file transferring process.

Table 2: The Experimental Parameters Values for Scenario 1

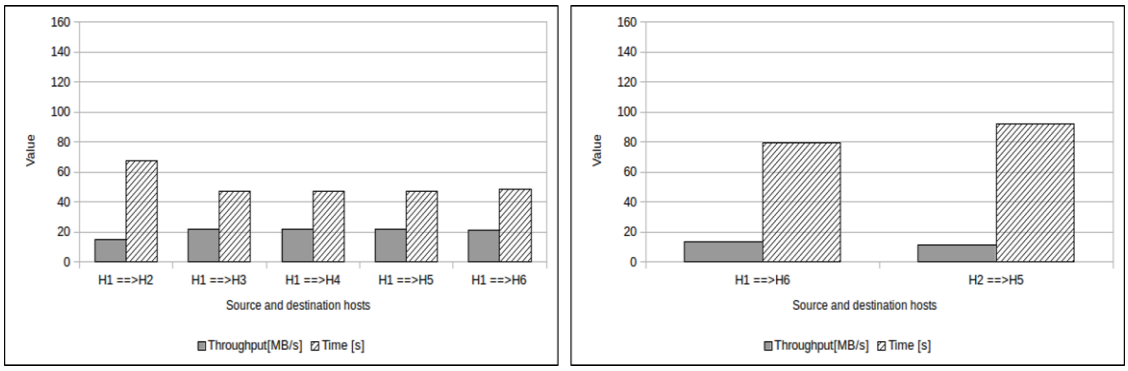
Scenario 1	FTP Server	FTP Client	Number of Flows	File Size	Run time Process
Test 1	H1	H2, H3, H4, H5, H6	25	1G	(one by one)
Test 2	H1, H2	H6, H5	10	1G	Parallel



(a) Test 1

(b) Test 2

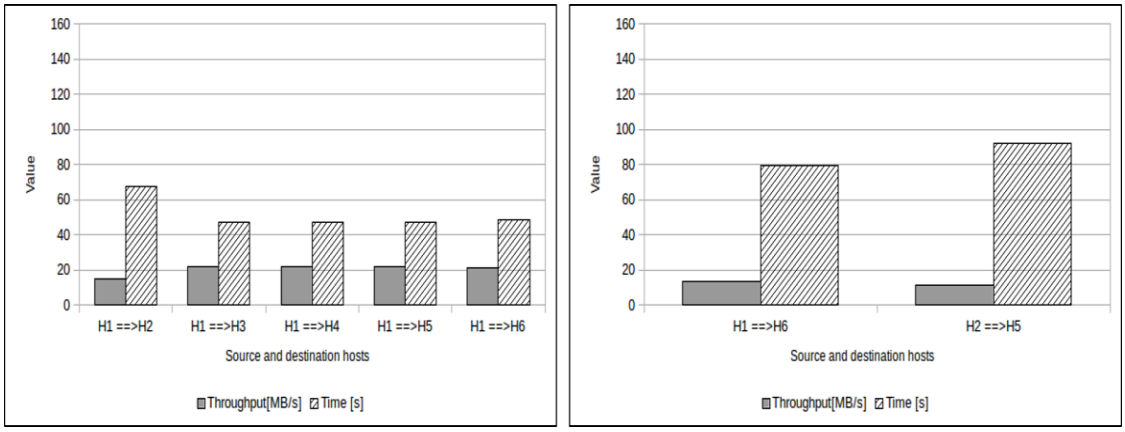
Figure 7: Throughput and Time Results of Scenario 1 by Shortest Path Routing



(b) Test 1

(a) Test 2

Figure 8: Throughput and Time Results of Scenario 1 by Link Utilization-aware Routing



(a) Test 1 (b) Test 2

Figure 9: Throughput and Time Results of Scenario 1 by QoS-aware Traffic Management Method

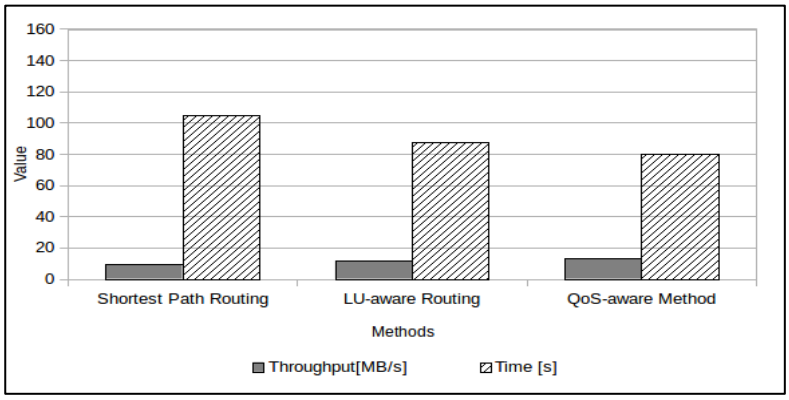


Figure 10: Throughput and Time Results of Test 2 in Scenario 1 by the Three Methods

In test 1, host h1 served as an FTP server and other hosts served as the clients. Clients are downloaded a file whose size is 1G (1024Bytes). To parallel accessing, test 2 generated cross traffic between hosts h1, h6, h2, and h5. For this test, hosts h1 and h2 served as FTP servers and hosts h5 and h6 performed as the clients. Figures 7, 8, and 9 depicted the throughput and time results for accessing FTP servers by shortest path routing, link utilization-aware routing, and QoS-aware traffic management method, respectively. The value of time bar showed the time duration that took when downloading the file and the throughput bars showed which MB/s are used to download file from the FTP server. Figure 10 summarized the throughput and time results for test 2, parallel accessing FTP servers (h1 and h2) from the clients (h5 and h6). According to the figure 10, the shortest path routing has downloaded a file with lower throughput but taken highest amount of time. Link utilization-aware routing has got higher throughput than the shortest path routing. The proposed method and Link utilization-aware routing got nearly the same throughput results with the significant difference is 1.146. The throughput result of Link utilization-aware routing is a little smaller than the QoS-aware traffic management method. However, our proposed method took less time to accomplish this file transferring process.

(1) Scenario 2: Testing with Different Application Traffic Flows

This scenario used different application traffic flows such as, video streaming and haptic streams. The main goal of this test is analyzing how the three methods perform to conduct these different types of application traffic flows (QoS-aware and non QoS-aware). The experimental parameters tested with scenario 2 are as shown in table 3. To apply video streaming, this test used VLC server and client. Host h1 served as the VLC server which stream the video to the VLC client host h6.

Table 3: Experimental Parameters for Scenario 2

Application traffic flows	Types of traffic	Options
Video streaming	QoS-aware traffic	VLC Server (h1), VLC Client (h6)
Haptic stream	Non QoS-aware traffic	Generated by using D-ITG

The streamed video file size is 101.1 MB and the duration for the video is 11 minutes and 25 seconds. VLC server streamed the video by using HTTP protocol. This scenario also analyzed packet loss rate of video streaming by applying the three methods (shortest path routing, link utilization-aware routing, QoS-aware traffic management). To analyze packet loss rate of video streaming, this scenario checked packet retransmission of TCP connection.

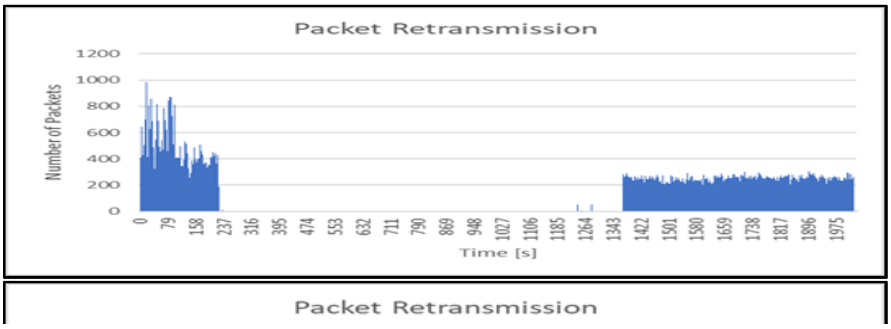


Figure 11: Video Streaming with Shortest Path Routing

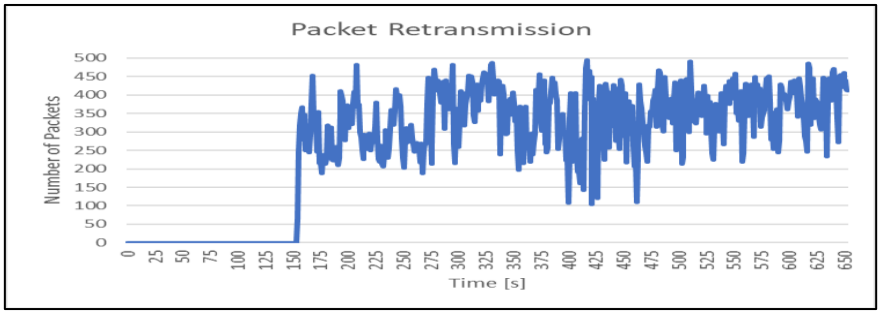


Figure 12: Video Streaming with Link Utilization-aware Routing

Figures 11, 12, and 13 showed the occurrence of packet retransmission, alternatively, the occurrence of packet loss when the VLC server h1 was streaming video file to VLC client h6. According to the packet retransmission analytical results, the shortest path routing got the highest number of retransmitted packets (nearly 1000 packets) and the packet loss rate is nearly 20% as shown in figure 11.

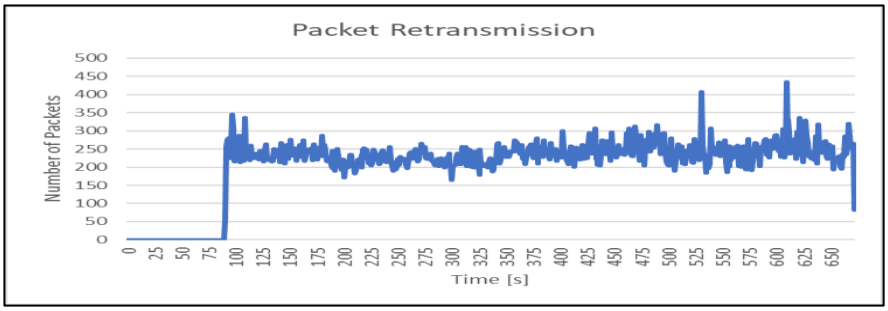


Figure 13: Video Streaming with QoS-aware Traffic Management Method

When video streaming is conducted by Link utilization-aware and QoS-aware traffic management, the highest number of retransmitted packets are nearly 500 packets and less than 450 packets, respectively. For the QoS-aware traffic flows (video streaming), the proposed method considered two QoS parameters constraints such as link utilization and link delay. Video streaming application required less delay and more bandwidth path to maintain its video quality. The calculation of link capacity and link delay needed the significant time, however, the proposed method guaranteed fast packet transmission and packet loss. According to the analytical results of figures 11, 12, and 13, our proposed method not only got a smaller number of retransmitted packets than the others but also got less time duration to transmit the video file.

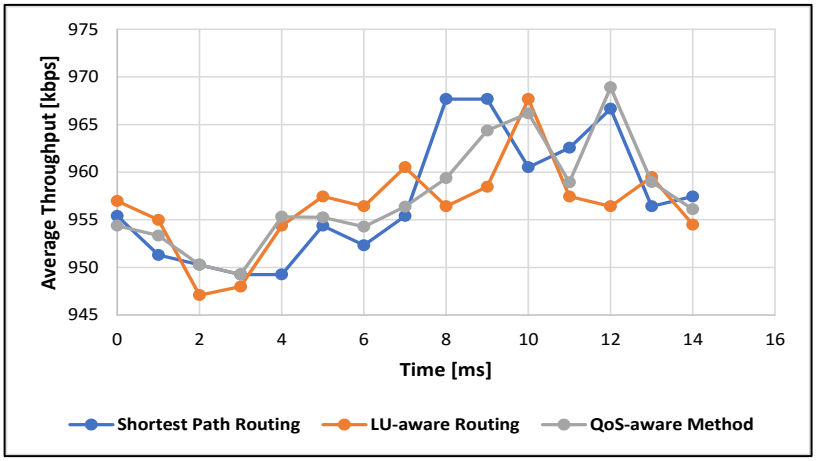


Figure 14: Comparative Results of Haptic Stream by the Three Methods

To conducted the haptic stream, scenario 2 used D-ITG [13]. Figure 14 showed the throughput versus elapsed time of haptic streams by applying the three methods. In this work, we defined haptic stream as the non QoS-aware application traffic and the generated haptic stream has smaller average bit rate and size. When the haptic

stream is conducted by using shortest path routing, shortest path routing does not need traffic classification and measurement steps, it finds the shortest paths and simply forwarded the incoming traffic through one of the shortest paths. When haptic stream is conducted by using link utilization-aware routing, the routing does not classify types of application traffic flows but it calculated the link utilization. This calculation and installing flow rules took time. The proposed method, QoS-aware TE classified types of application traffic flows then forwarded the traffic by using two different traffic management methods. The classification process and calculating QoS parameters may also take a significant amount of time. Therefore, according to the figure 14, the average throughput results of the shortest path routing outperformed the others because the traffic volume of haptic stream is smaller than the link capacity and shortest path routing does not need to classify traffic and calculate QoS parameters. For the QoS-aware traffic flows, the throughput results of our proposed method outperformed the others.

5. Conclusion

In this paper, we proposed a QoS-aware traffic management method in SDN. To get the effective traffic management, the proposed method firstly classified types of QoS-aware traffic flows. Then, the proposed method routed the classified traffic through the path based on the types of traffic flows and estimated QoS parameters in traffic measurement. The main goal of our proposed method is to select the optimal path for QoS-aware application traffic flows. When the QoS-aware application traffic flows entered the network, the proposed method calculated the link utilization and link delay then forwarded the traffic flows through the maximum link utilization and minimum link delay path. The classification of traffic flows and calculation of QoS parameters took a significant time. However, for the QoS-aware application traffic, our proposed method guaranteed packet transmission and packet loss. The experimental results demonstrated that the average throughput results of the proposed method outperformed the others two methods when the conducted traffic is QoS-aware application traffic. And, the proposed method also reduced packet loss rate than the others. When the incoming traffic volume is smaller than the link capacity, the shortest path routing got fast packet transmission.

6. Recommendations for Future Work

Although our QoS-aware traffic management implemented to fulfill its objectives, there has still left some works for future extension. The proposed QoS-aware traffic management took account of the network QoS constraints such as available bandwidth and delay. For the future work, we will not only compute other QoS parameters but also compare compare our proposed method with other traffic management methods and study with complex network typologies and other NOS.

References

[1] Sinh, D., Le, LV., Lin, BSP., and Tung, LP. “SDN/NFV-A new approach of deploying network infrastructure for IoT,” IEEE Wireless and Optical Communication Conference, pp. 1-5, 2018.

[2] Open Networking Foundation. “Software Defined Networking: the new norm for networks,” Web. White Paper, 2014.

[3] Jmal, R. and Fourati, LC. "Implementing shortest path routing mechanism using Openflow POX controller," IEEE International Symposium on Networks, pp. 1-6, 2014.

[4] Cheng, LC., Wang, K., and Hsu, YH. "Application-aware routing scheme for SDN-based cloud data centers", Ubiquitous and Future Networks, pp. 820-825, 2015.

[5] Fonseca, P., Bennesby, R., Mota, E., and Passito, A. "A replication component for resilient OpenFlow-based networking", Network Operations and Management Symposium, pp. 933–939, 2012.

[6] QoS routing mechanisms and OSPF extensions, IETF, RFC 2676, 1999.

[7] Kodialam, M. and Lakshman, TV. "Minimum interference routing with applications to MPLS traffic engineering," IEEE INFOCOM, vol. 2, pp. 884 - 893, 2000.

[8] Deng, DC., and Wang, K., "An application-aware QoS routing algorithm for SDN-based IoT networking", IEEE Symposium on Computers and Communications, pp. 186-191, 2018.

[9] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, pp. 1228-1234, 1996.

[10] Jarschel, M., Wamser, F., Hohn, T., Zinner, T., and Tran-Gia, P. "SDN-based application-aware networking on the example of YouTube video streaming", European Workshop on Software Defined Networks, pp. 87-92, 2013.

[11] ONOS [Online]. Available at: https://onosproject.org.

[12] Mininet [Online]. Available at: http://mininet.org.

[13] Avallone, S., Guadagno, S., Emma, D., Pescapè, A., and Ventre, G., "D-ITG Distributed Internet Traffic Generator", First International Conference on the Quantitative Evaluation of Systems, pp. 316-317, 2004.