Effective Traffic Rerouting for Video Streaming in Software Defined Networking

Hla Myo Su
Faculty of Computer Systems and Technologies
University of Computer Studies, Yangon
Yangon, Myanmar
hlamyosu@ucsy.edu.mm

Aung Htein Maw
Faculty of Computer Systems and Technologies
University of Information Technology
Yangon, Myanmar
ahmaw@uit.edu.mm

***Abstract* — Video streaming exceeds all other traffic types on the internet. Now, it occupies a significant portion of total internet traffic. The transmission mechanism used by the video stream affects not only network traffic but also the users' Quality of Experience (QoE). Software Defined Networking (SDN) is a manageable, programmable, adaptable, and cost-effective emerging network architecture, which is very suitable for the dynamic natured high-bandwidth of modern applications. The OpenFlow protocol is a fundamental element in the creation of SDN solutions. Traffic engineering is a favorite mechanism used by network administrators when optimizing the existing networks and providing more services to customers is needed. This paper proposes a traffic rerouting approach in the video streaming flow management framework on SDN architecture, which can effectively reroute video streaming traffics to reduce packet loss and improve bit rates. It can calculate the current link utilization over the network links and it uses least-cost path rerouting rather than the default routing, reactive forwarding in the ONOS controller. The proposed method is validated by using the popular Mininet network emulation environment and OpenFlow features. The experimental results verify that the improvement of performance by the measurement of video streaming bit rates and packet loss of video traffic.**

Keywords—SDN, ONOS, OpenFlow, Video Streaming

I. INTRODUCTION

Video streaming service is very popular in modern networks and demands a large number of resources, so accurate video traffic models are needed. The concept of streaming has quickly become favorite because it allows video viewers to view the data without having to download the entire video first. Since then, the technologies that affect video streaming have been developing rapidly. Consequently, many new techniques have been evolved for video transmission. Although providing high-quality service is the most challenging task, researchers are trying to solve this challenge by providing a more efficient network in which bandwidth limitations, congestion, and unsatisfied users are restricted. In the modern multimedia networks, the new challenges shift from technology-oriented services to user-oriented services, demonstrating the importance of QoE. However, no one can deny that video streaming has become a new super-popular sphere of entertainment. The new Visual Networking Index (VNI) by Cisco predicts that 60% of the global population will be internet users and video will make up to 82% of all IP traffic by 2022 [1]. Thus, this paper focuses on video traffic.

Traditional networking is usually hardware-based and rooted in the fixed-function network devices. It is expensive, time-consuming and difficult in changing. Software Defined Networking (SDN), a category of programming technologies that enables more intelligent provisioning, centrally control, and policy-based management of network resources. By expanding the traditionally closed network platforms and implementing a common SDN control layer, administrators can consistently control the integrated network and its equipment without considering the complexity of the underlying network technology.

Video streaming is the continuous video files transmission from the server to the client. The media file being played on the client device will be stored remotely while streaming, and transmitted over the Internet a few seconds at a time. Streaming takes place in real-time, which is more effective than downloading the entire media files. If a video file is downloaded, save a copy of the entire file to the hard drive of the device, and the video cannot be played until the file has completely downloaded. If it is changed to streaming, the browser will play the video without actually copying and saving it. If too much data is sent over the network, it may cause network congestion and even degradation of streaming performance.

In large traffic networks, traffic congestion is a key issue. When the traffic transmitted by the network link exceeds its capacity, it will be overloaded. So, the data packets are either lost or delayed, which reduces the network performance. This will ultimately reduce the performance of the network application or service and bring an unsatisfactory experience to the end user. In general, the information about the available bit rates and packet loss rates will benefit many administrators and users of applications and network infrastructures. So, measuring the traffic statistics of video streaming hosts and rerouting the optimal path involves a critical path in adjusting to the current network conditions.

This paper compares the experimental results of video stream rerouting that uses the least-cost path rerouting rather than the default routing, reactive forwarding in Open Network Operating System (ONOS) controller, thereby reducing packet loss and increasing the transmission bit rate. Video traffic is most of the traffic load on the Internet. It may cause network congestion, which is the reason for the degradation of client video quality and also affects QoE.

The rest of this article is mentioned below. Section 2 will state the related works. In the next Section 3, the overview of background knowledge of the system and video streaming flow management framework are explained. Then, Section 4 reports the testbed platform and experimental results of rerouting video traffics. Finally, Section 5 discusses the conclusion of this paper.

II. RELATED WORK

In [2], Dynamic Adaptive Streaming over HTTP (DASH) runs on Hypertext Transfer Protocol (HTTP) using Transmission Control Protocol (TCP) as the transport layer. When more than one DASH clients compete for bandwidth, it can cause player instability, unfairness between the bit rates requested by player, and insufficient network bandwidth utilization.

HTTP Adaptive Streaming (HAS) traffic shaping is introduced in [3] to enhance users' QoE in home network. The

literatures tried traffic shaping in the home gateway based on expected bit rate of each video stream. Although this technology is not dynamic, traffic shaping is performed at the starts of the session, and when a new video stream begins streaming or the video stream leaves the session, the traffic shaping does not change.

For enhancing the delivery of video content and improve the end users' QoE, [4] proposed Server and Network Assisted DASH (SAND) architecture. It can obtain QoE metrics from the clients and return network-based metrics to enhance its overall clients' QoE. The third-party measurement server in SAND is called DASH-Aware Network Elements (DANE) , which can measure information of different including CDN, ISP and content providers. Therefore, it is difficult to implement an architecture.

NAVS (Network Assisted Video Streaming) proposed by Abuteir et al. [5] aims to better QoE by reducing player instability, maximizing fairness between clients, and improving video quality. Adaptive logic is an algorithm based on bandwidth estimation. NAVS evenly distributes the available bandwidths between video clients and separates the channels by additional bandwidth between video streams so that each client can stream video without any interference from others. But, this separation decreases the bit rate and leads to underutilization.

III. Proposed Effective Video Streaming flow management application Framework

The background theory of the system is depicted in Fig. 1.

A. *Background*

SDN can respond to traffic demands based on real-time network status, it can also be programmed to handle future traffic demands. It can configure any type of route in the network, and the overhead is very low. The SDN application takes care of calculating the route and allocating the bandwidth for the whole network. It is expected to reduce the complexity of statically defining networks; make it easier to automate network functions; and allow simpler provisioning and policy-based network resources management. SDN uses open API software applications to program network behaviors in a centralized control manner.

The switch is connected to the controller, and the controller uses the OpenFlow protocol to guide the switch. The OpenFlow switch contains one or more flow tables, group tables, and meter tables. The flow table and group table are used in the lookup or forwarding phase to forward the packet to the appropriate port. The meter table is used to perform simple QoS operations (such as rate-limiting) to complex QoS operations (such as DiffServ). The OpenFlow channel is used to link to an external controller.

ONOS controller adopts a powerful structure for global network state size, high performance, low latency, availability, and scalability requirements of the enlarge network.

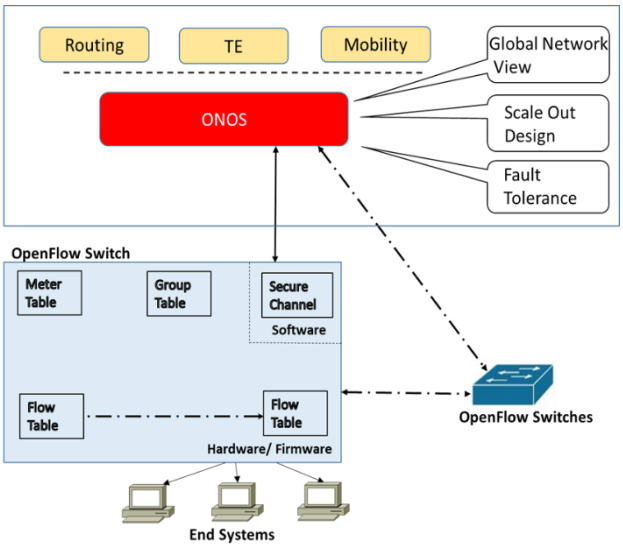

Fig. 1. OpenFlow and ONOS controller Communication Components

B. *Improved Video Streaming Framework*

The video streaming flow management application framework is proposed with the measurements of packet loss and transmission bit rates within the network to build an enhanced video streaming QoE. The video stream monitoring and dynamic traffic shaping method is proposed, which depends on the collected network traffic statistics. The controller includes five functional modules, as shown in Fig. 2.

1) Receive and Classify Requests

When the controller receives a request from any client, it detects the type of request whether the video streams or not. The PacketProcessor examines the packet by context.inPacket( ) function. The packet_in message or context packet chooses which application will receive it. ONOS controller defines packet director levels when one more application is running on it. Therefore, it is set packet director level 1 to the proposed application and level 2 to reactive forwarding application default in ONOS controller. This means that every first context packet from every device will receive from the proposed application firstly and if it does not handle this packet, redirect to the next reactive forwarding application. Therefore, the reactive forwarding application makes a decision and install it. The later sequence of flow is pass through via this rule with priority 10. When the proposed application received video streaming flow and it makes a new decision for this flow and installs it with priority 5001. The later sequence of flow will pass through via the rule of the proposed application, and the old rule is automatically removed after 10 seconds in idle timeout.

It extracts the source address, source port, destination address, and destination port of the requested packet. Then, the packet's source address, destination address, and port numbers are matched with video server's address and port number. If a video steaming request is detected, then it directs to the video streaming flow management application. After that, it is sent to the next module, Media Presentation Description (MPD) analyzer in the proposed application.

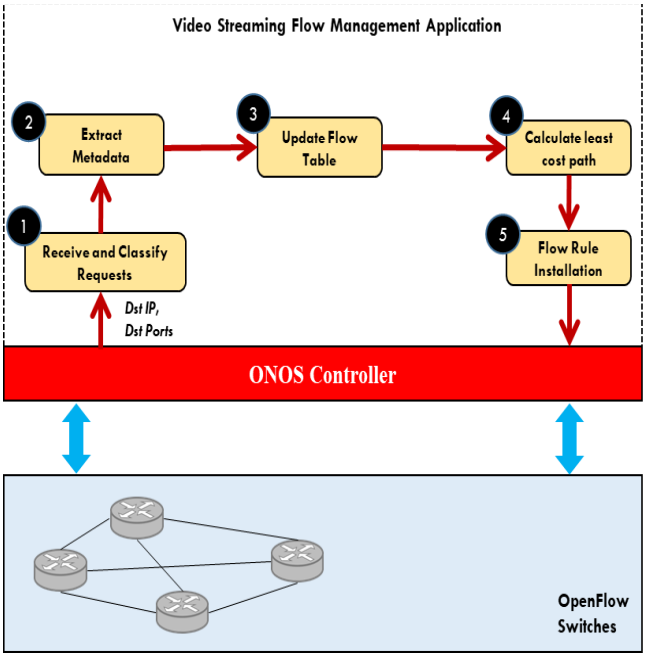

Fig. 2. Proposed System Architecture

2) Extract Metadata

If the request is a video stream, the Media Presentation Description (MPD) file concern with the requested video stream is sent to MPD analyzer module for extracting metadata values. The metadata contains available bit rate, video length, number of chunks, and chunks' URL.

3) Update Flow Table

The Active Video Flow Info table keeps new request from video clients. When video flow is inactive for a period of time, it means that the video streaming has stopped and will be refreshed by deleting it from the Active Video Flow Info table. Then the information of the active video flow is also deleted.

4) Calculate the Least-Cost Path

It will cause network congestion due to the access of limited bandwidth. When one or more video clients will compete for bandwidth, it may cause performance degradation and also affect users' QoE. In order to solve the traffic congestion problem and to improve QoE, any two video streams should have equal bandwidth, and the bandwidth assigned to each video streams is less than total bandwidth. Either the new video stream starts or the existing video stream stops, the suitable new bandwidth value for each stream should be recalculated to obtain fairness between the video clients.

5) Flow Rule Installation

The optimal path for the video requests is chosen among the available paths based on the current traffic statistics, then by using the FlowRuleService of the ONOS controller and new flow rules are set to the corresponding network devices. When a new flow arrives at the OpenFlow switch, some header fields are needed to match against the rules in the table. If there are no rules, the default table miss entry is applied. Table miss entry sends the first packet of flows to the controller via OF packet_in message.

IV. Experimental Testbed and Performance Evaluation

In order to improve video streaming performance, this experiment uses current bandwidth utilization based least-cost path rerouting. It measures the bit rate and the packet loss rate of the video stream. The well-known network emulator, Mininet, that can configure the rapid prototyping to create SDN controllers, OpenFlow switches, hosts, and network links on a single computer virtually.

The testbed topology in this experiment is shown in Fig. 3. This topology connects one video server (h1), six video clients (h2, h3, h4, h5, h6, and h7), one controller (C0), and five OpenFlow switches (S1, S2, S3, S4, and S5). In addition, an SDN controller to monitor the network traffic of the video server and streaming clients. While currently not implemented, the eventual goal of this testbed is to provide the considerable facts for the huge network traffic congestion with the number of users, servers, and the client applications.

VLC media player is used on both sever-side and client-side of video streaming. Moreover, the simple HTTP is bound on sever-side at port 8080. The video codec H.264 is used to encode the bit rate of this adaptive video streaming. VLC player streams out the video packets by using TCP mode. The Wireshark analyzer has captured all packets transmitted between the server and clients.

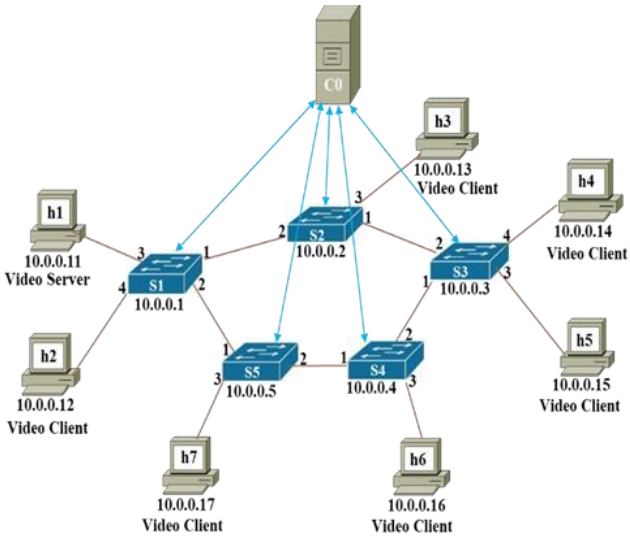

Fig. 3. Emulation Testbed Topology for System Implementation

TABLE I. HARDWARE REQUIREMENTS

No	Name	Specifications
1.	Laptop	Core i7-8565U CPU @ 1.80GHZ with RAM 8GB

TABLE II. SOFTWARE REQUIREMENTS

No	Name	Specifications
1.	Operating System	Ubuntu 16.04 LTS (64 bits)
2.	Mininet Emulator	Version 2.3
3.	ONOS controller	Version 1.8
4.	OpenFlow Protocol	Version 1.3
5.	VLC Media Player	Version 2.2.2
6.	Wireshark Analyzer	Version 2.6.10
7.	Animation Video (Big Buck Bunny)	9.56 Minutes ➢ 480 x 360p, 40.4 MB ➢ 1280 x 720p, 89.4 MB

The transmission of large amounts of data through the network will cause traffic congestion and even affect the users' QoE in the video stream. The critical demand on that situation is to monitor the network traffic and to reroute the traffic dynamically to the appropriate path for solving the requirements. The statistics of the traffic and the existing network conditions are the major parts in the route selection. According to the real-time network conditions, an optimal path is conditionally selected by the path selector. Most literature reviews are based on the link utilization to select suitable paths for traffic flows.

A. Estimation Current Bandwidth Utilization

In order to choose the best path for rerouting the video stream, the current load on the link is needs to be calculated. The traffic statistics of every OpenFlow switch are gathered by polling at fixed intervals of 10 second. It is collected the statistics for each flow, each table, and each port from all connected switches in the network. Then, the statistics such as received bytes and sent bytes of each link on each path is counted to estimate the link utilization of this path. The total number of bytes sent and the total number of bytes received for a specific port on each link means the total number of bytes transmitted (or total bandwidth usage). The current total number of bytes (λ) of each link on each path can be computed as (1).

$$\lambda = src\_port.Received\_bytes( ) + src\_port.Sent\_bytes( ) + dst\_port.Received\_bytes( ) + dst\_port.Sent\_bytes( ) ; \quad (1)$$

Equation (1) can be used repeatedly to calculate the total transmitted bytes of each path until there are no more links on the path. Suppose that the graph $G = (V, E)$ is network topology, where V represents the node set and E represents the edge set. Let P_q is the set of all possible paths in the source-destination pair q. It is acceptable for any path, $I \in P_q$ has many M_i sub-paths, and each sub-path $j \in M_i$ has many L_{ij} links. μ_{kji} and λ_{kji} used to represent the link capacity and the link load on the link k of sub-path j of path i, and the link utilization (LU) is given by (2).

$$LU_{kji} = \frac{\lambda_{kji}}{\mu_{kji}} \quad (2)$$

Where;
LU_{kji} = Link Utilization of sub-path j of path i on link k
μ_{kji} = Link capacity of sub-path j of path i on link k
λ_{kji} = Link load of sub-path j of path i on link k

Solve the link load problem by polling the corresponding port of switch using the standard OpenFlow mechanism.

B. Calculation the Least-Cost Path

In order to choose the least-cost path, all available shortest paths are firstly calculated, and then the cost c_k of each path is computed by using the link utilization and the current bandwidth usage as mentions in (3).

$$c_k = \frac{\lambda_k}{LU_k} \quad (3)$$

Algorithm 1. Least_Cost_Path Rerouting Algorithm

Input : Request for stream

Output : Least_Cost_Path

Initialization: $\lambda = 0$, $\mu = 0$;

If context.inPacket( ) is video_requested_packet **Then**

Pass to Video Streaming Flow Management Application

Find all possible shortest paths from Server to Client

For each path p in all possible shortest paths **Do**

For each link $k \in 1, 2, \ldots, n$ **Do**

Compute current bandwidth usage λ , for k

$\lambda_{kji} \leftarrow \lambda_{kji} + \lambda_k$

End For

Calculate Link Utilization: $LU_{kji} = \frac{\lambda_{kji}}{\mu_{kji}}$;

Find Least_Cost_ Path $c_k = \frac{\lambda_k}{LU_k}$;

End For

Install flow rules to Least_Cost_Path Rerouting;

Else

Use Reactive Forwarding Default Routing;

End If

Subsequently selecting the least-cost path from all possible paths, it is used the FlowRuleService in the ONOS controller to put the new flow rules into the corresponding device. By means of the video streaming bit rate and the received packet, the experimental evaluations are compared with the reactive forwarding default routing in the ONOS controller.

Fig. 4. Transmission Bit Rates Comparison at Each Host in 360p Low-Resolution

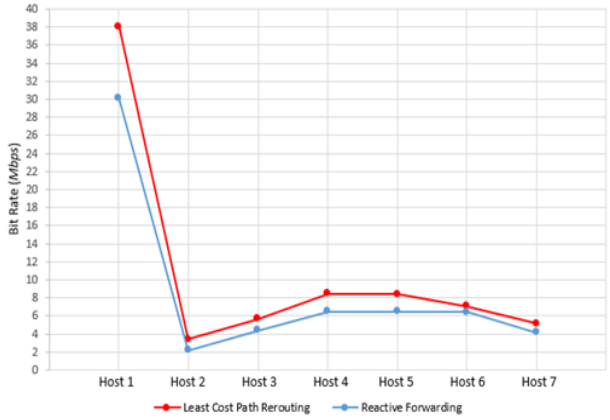

Fig. 5. Transmission Bit Rates Comparison at Each Host in 720p High-Resolution

The ONOS controller supports DijkstraGraphSearch as a module. The reactive forwarding routing is a hop-by-hop-based shortest path routing to find the shortest path between the source and the destination pairs. Even if the shortest path is not the path with the best performance, the shortest path between the source and the destination is used to route the packets. This comparison result is shown in Fig. 4, for the 360p low-resolution, and in Fig. 5, the 720p high-resolution, by this experimental testbed topology. The video clients, h3 and h7 have approximately equal bit rates because they are hop-count 1 routing hosts. And the other video clients, h4, h5, and h6 also get the equally same bit rates according to the hop count based routing, they are hop-count 2 link hosts.

The least-cost path allows the path selector to send traffic over a longer but less congested link, ignoring the shortest path rule. Before choosing the path for video transmission, it firstly computes the current total transmission bytes on the links. And then, depending on the current traffic statistics such as link utilization, link load, and link capacity, the route is chosen as the optimal path in streaming video. This rerouting method reduce the traffic bottlenecks and provide the administrators for using the maximize of the current network resources. The resulted bit rate at each video host of the proposed rerouting method is also shown in Fig. 4 and Fig. 5 compare with the default routing bit rate. The least-cost path rerouting bit rate is higher than the rest because this method is realistic based on the current traffic statistics.

The received packets at each video host is compared, the 360p low-resolution in Fig. 6, and the 720p high-resolution in Fig.7, respectively. According to these figures, compared with the reactive forwarding default route in the ONOS controller, the least-cost path rerouting scheme receives more data packets. This shows that the proposed method reduces packet loss and tends to improve the users' QoE.

As shown in Fig. 8, in the 360p low-resolution video stream, the least-cost path rerouting method has a bit rate slightly higher than reactive forwarding by 18.98%. The bit rate of the rerouting method is significantly increased by 26.62 in 720p high-resolution video streaming, as depicted in Fig. 9. As the result, the proposed least-cost path rerouting is more effective in high-resolution video streams.

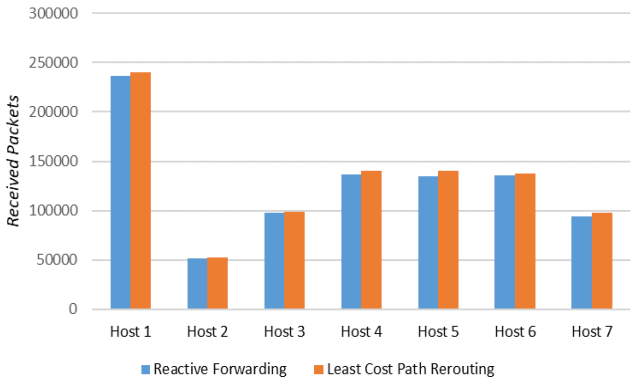

Fig. 6. Received Packets Comparison in 360p Low-Resolution

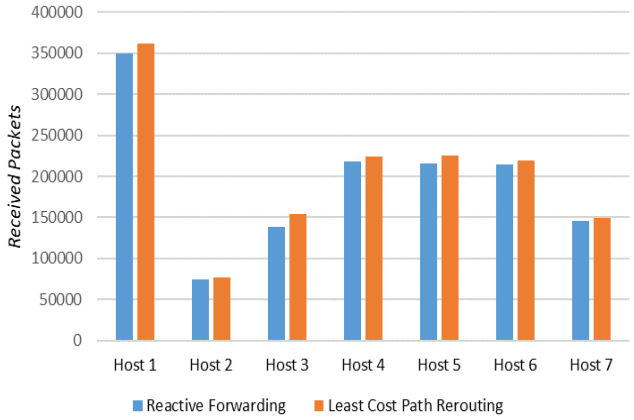

Fig. 7. Received Packets Comparison in 720p High-Resolution

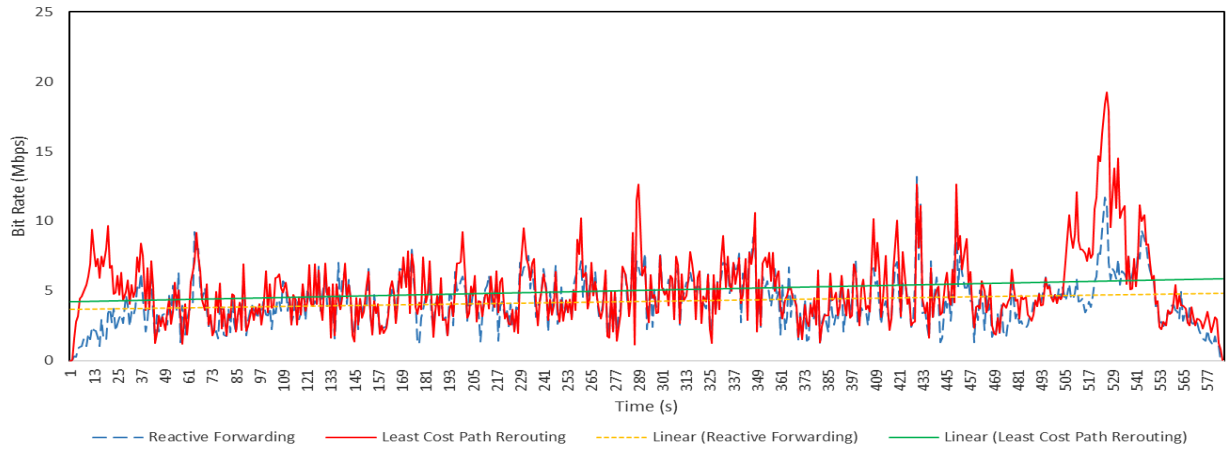

Fig. 8. Average Bit Rates Comparison in 360p Low-Resolution

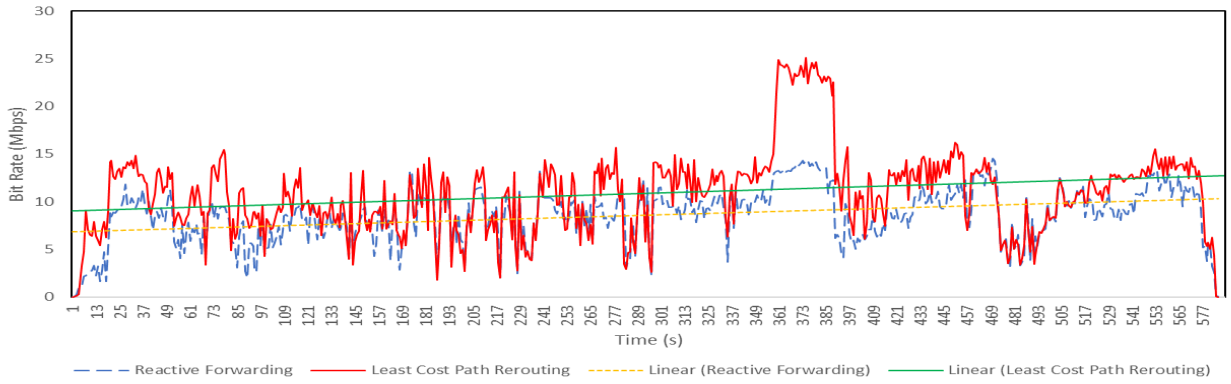

Fig. 9. Average Bit Rates Comparison in 720p High-Resolution

V. Conclusion

The video streaming flow management application using the least-cost path rerouting method is proposed in this paper. It is different from the reactive forwarding default routing in the ONOS controller since the proposed method use the link load utilization based routing instead of hop count based routing. The comparison of experimental results is evaluated by the number of received packets and the transmission bit rate with two resolution of animation videos. The proposed least-cost path rerouting method improves video streaming transmission bit rates to 18.98% in 360p low-resolution and 26.62% in 720p high-resolution, respectively. This improvement can reduce the network traffic congestion and advantage in implementing the effective video streaming flow management application to enhance QoE. The proposed application is implemented and validated by using Mininet for network emulation and ONOS for the SDN controller. In order to simplify the comparison and analysis, the test platform topology in the current simulation only contains one video server. This is insufficient in the real world. Based on the proposed rerouting, future work will be to add more video servers and real-world deployment.

References

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, “Cisco Visual Networking Index (VNI) complete forecast update,” 2017-2022. December 2018.

[2] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," in IEEE MultiMedia, vol. 18, no. 4, pp. 62–67, April 2011.

[3] R. Houdaille and S. Gouache, “Shaping HTTP adaptive streams for a better user experience,” in Proceedings of the 3rd Multimedia Systems Conference, New York, USA, pp. 1–9.

[4] ISO/IEC JTC1/SC29/WG11 (MPEG) Report, Technical report, ISO, November 2013.

[5] R. M. Abuteir, A. Fladenmuller, and O. Fourmanx, “SDN based architecture to improve streaming in home network,” IEEE 30th International Conference on Advanced Information Networking and Applications, 2016, pp. 220–226.

[6] W. Stallings, “Software-defined networks and OpenFlow,” The Internet Protocol Journal, vol. 16, No. 1, pp. 2–14, March 2013.

[7] R. Houdaille and S. Gouache, “Shaping http adaptive streams for a better user experience,” in Proceedings of the 3rd Multimedia Systems Conference, ACM, 2012, pp. 1–9.

[8] C. Xu, B. Chen, and H. Qian, “Quality of service guaranteed resource management dynamically in software defined network,” Journal of Communication vol. 10, No.11, pp. 843–850, November 2015.