Emulated OpenFlow Based Experimental Study on Middle-box Buffering Effect for Multi-path Chunked Video Streaming

Conference Paper · June 2015

4 authors, including:

Phyo May Thet
Chulalongkorn University

Jongwon Kim
Gwangju Institute of Science and Technology

C. Aswakul
Chulalongkorn University

Emulated OpenFlow Based Experimental Study on Middle-Box Buffering Effect for Multi-Path Chunked Video Streaming

Phyo May Thet [1], Nattakarn Tientrakul[1], JongWon Kim[2], Chaodit Aswakul[1]
[1] Wireless Network and Future Internet Research Group, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Thailand
[2] School of Information and Communications, Gwangju Institute of Science & Technology, Korea
phyo.ma@student.chula.ac.th, nnattakarn.t@gmail.com, jongwon@nm.gist.ac.kr, chaodit.a@chula.ac.th

Abstract

In this paper, we have implemented the emulated multi-path OpenFlow network with splitting and middle-box functionalities for investigating video buffering effects on packet delay and jitter performance for an RTP video streaming session. The mininet emulator is used to construct the multi-path OpenFlow network. The network consists of a video server, a video client, four Open vSwitches (OVSs), a remote controller and a middle-box. The POX controller is used for adding flow entries into switches and assigning periodic splitting function. The reported mean/ standard deviation of packet delay results show that the proper chunk size ratio and initial buffering time need to be considered carefully to improve the video packet delay and jitter performance.

Keywords: OpenFlow; software-defined network; middle-box; multi-path video streaming; buffering; jitter.

1. Introduction

Software-defined networking (SDN) concept [1] is the new framework that is expected to enhance the controllability for network administrators to manage their network with a set of network functions residing in controllers and communicating to data-forwarding switches via an open interface like the OpenFlow protocol [2]. Additionally, mininet is a widely used SDN platform emulator which can create a complete experimental OpenFlow network within a single computer. SDN and mininet present to researcher and developer communities a convenient testbed allowing various network functions to be investigated for the deployment of future internet applications.

Steadily increasing demands of video streaming users is one of the major concerns of network service operators worldwide. Video streaming is nowadays responsible for the major bandwidth consumption and traffic congestion on the internet network. In order to deal with these challenging issues on bandwidth hunger, stringent delay time and packet loss requirements, multi-path live video streaming via TCP has been investigated in [3]. And the usage of SDN for video streaming over a single path OpenFlow testbed has been demonstrated in our earlier research trial [4]. However, in [4], no investigations have been carried out on setting a proper initial buffering time of the middle-box responsible for streaming out packets towards the client steadily to facilitate the quality of video service during the playback time at the receiving client. To improve the smoothness of video playback, the varying of delay and jitter need to be considered [5].

For solving the requirements of video streaming users upon network congestion, the so-called middle-box becomes a new SDN enabler for addressing e.g. security management and resource allocation function in the network [6]. Especially, these requirements need to be considered thoroughly when the tests are going to be extended to international SDN testbed such as OF@TEIN (OpenFlow @ Trans-Euraisa Information Network) testbed [7]. In that kind of international testbed spanning different time zones, the network congestion is geospatially time varying and international link bandwidths are limited by each nation. These limitations become our motivation for studying on buffering effects at the middle-box via multi-path video streaming to help aggregate higher path bandwidths than those achievable by relying only on a single restrictive-bandwidth path.

As in the previous research [8], we have constructed a prototyped system with splitting and middle-box functionalities for multi-path video streaming experiment. However, the investigation of multi-path chunked video streaming over emulated OpenFlow network in [8] focuses only on the effects of chunk size ratio i.e. the proportion of time the originating video packet stream is switched alternately on two parallel paths. In this paper, the objective is to shed some lights on the effects of initial buffering at the middle-box i.e. the time period of storing first initial packets of the video stream at the middle-box before they are streamed out towards the client.

2. Middle-Box Buffering Implementation for Multi-Path Video Streaming

The topology of multi-path OpenFlow network in this experiment is summarised in Fig. 1. To implement the emulated multi-path OpenFlow network, Mininet version 2.1.0+ has been used in Ubuntu 12.04 LTS with Intel Core 2 Duo 2.8 GHz x 2 personal computer. The topology consists of a VLC server (h1), a VLC client (h2), four OpenvSwitches (OVS1, OVS2, OVS3, OVS4) and a middle-box (m1). A remote controller is the POX controller (c0) responsible for adding flow entries to every switch. A Python script is used for assigning mininet-emulated link bandwidths between hosts and switches. Two paths used in the experiment are the one with bandwidth approximately to the average server's streaming rate and the other one with a lower bandwidth than the streaming rate. Particularly, the bandwidth of path 1 (h1-s1-s2-s3-m1) and path 2 (h1-s1-s4-s3-m1) are set to 0.25 Mbits/s and 0.15 Mbits/s, respectively. Periodic packet splitting technique is adopted for transmitting video traffic between those two paths. No extra link delay between hosts and switches has been introduced so that the resultant delay represents the best case scenario obtainable before the test is later extended to OF@TEIN international testbed.

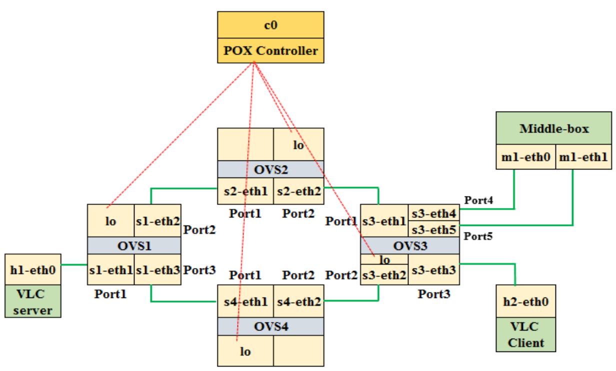

Fig.1. Topology of multi-path OpenFlow network

To study the middle-box buffering effect on multi-path video streaming, a scheduling algorithm of middle-box has been written in Python with scapy, dpkt and pcapy packages for assigning initial buffering time and performing packet transmission processes. As in [8], OVS1 performs as a splitter, feeding video chunks to path 1 (h1-s1-s2-s3-m1) and path 2 (h1-s1-s4-s3-m1) periodically by specifying *m* s for transmission via path 1 and *n* s via path 2. The splitting period is thus *m* + *n*. The time proportion of *m*:*n* is defined as the chunk size ratio. OVS3 performs as a packet combiner and marks a layer-2 MAC addresses of s3-eth1 and s3-eth2 for specifying each packet come from path 1 and path 2, respectively. Then OVS3 forwards these combined packet stream via s3-eth4 for further processing at the middle-box. Middle-box performs as a packet scheduler and classifier with two parallel buffers (buffer 1 and buffer 2). The packets coming from path 1 and path 2 are classified by their MAC addresses being marked by OVS3 and queued at the buffer 1 and buffer 2, respectively. The middle-box uses the RTP timestamp values for reordering the packets in buffer 1 and buffer 2, and forwards these packets from m1-eth1 to s3-eth5, and then towards s3-eth3. Finally, s3-eth3 forwards the packets to the client.

3. Video Streaming Scenarios and Results

Unlike in [8], a Frozen video clip with H.264 video codec, 320x144 resolution, total video/audio bit rate 246 kbits/s and 2-minute playback duration has been used in the experiment. These settings have been chosen to allow the dynamic range of testable input parameters with available performance of computer hardware specification. VLC server program has been used to stream out the video by RTP mode. The initial buffering time varies from 0, 5, 10, 15 to 20 s. The chunk size ratio of 20:10 and 10:20 have been used. The packet scheduling rate of middle-box is set to 25 packets/s to match with the server video transmission rate.

Wireshark software is used to capture all the transmitting and receiving packets on both Ethernet interfaces of server and client. To investigate the effect of initial buffering time and chunk size ratio on packet delay, we have tested three times with the same parameter settings and computed the mean/standard deviation of packet delay. Figs. 2 and 3 depict the mean/standard deviation of packet delay in our experiments. These are the average results from three experiments with chunk size ratio (20:10 and 10:20) and initial buffering times (0, 5, 10, 15, 20 s).

From Fig. 2, when initial buffering time increases for both chunk size ratio settings, the mean packet delay increases with its upper bound at the initial buffering time. The mean packet delay of chunk size ratio 20:10 case is lower than that of 10:20 case. In case of 20:10, most of the packets are streamed out through path 1 and use only a short period of time on path 2. The former path has enough capacity to carry the whole incoming traffic than the latter path. When compare to the packet delay results between 20:10 and 10:20 cases, it is clear that proper chunk size setting is important to reduce packet delay due to the higher delay of 10:20 in which using the lower bandwidth path is more than the higher bandwidth path.

Moreover, we studied the jitter performance in terms of the standard deviation of packet delay. The standard deviation of packet delay results from Fig. 3 demonstrates that jitter performance of 20:10 case is better than that of 10:20 case. In order to obtain the best jitter performance, the proper initial buffering time and chunk size ratio settings need to be considered carefully. The reason that

20:10 case is better than 10:20 is because path 1 has enough 0.25 Mbits/s capacity to carry the whole video traffic 0.246Mbits/s while path 2 has only 0.15 Mbits/s. Finally, it is noted that, since the middle-box can handle all the video packets in this two-minute video streaming experiment, the packet loss ratio is zero in most experiments. And the high performance machine is recommended for operating the middle-box function in real network since middle-box has to process a number of functions on fast-coming packets.

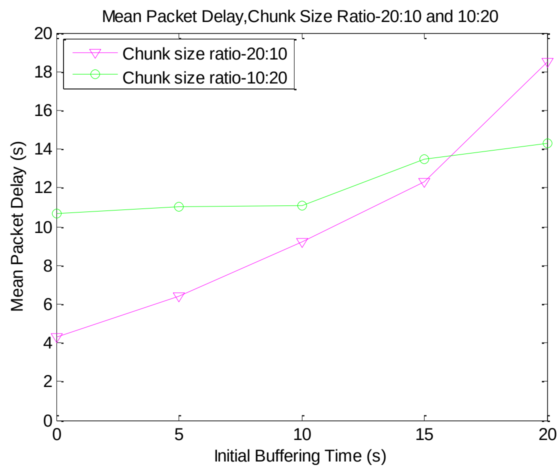

Fig.2. Mean packet delay vs initial buffering time for chunk size ratio – 20:10 and 10:20

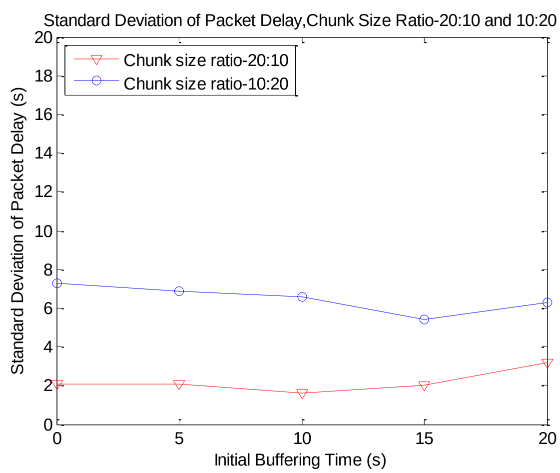

Fig.3. Standard deviation of packet delay vs initial buffering time for chunk size ratio – 20:10 and 10:20

4. Conclusions and Future Work

In this paper, we have implemented a prototyped multi-path OpenFlow network with splitting and middle-box functionalities for investigation of the middle-box buffering effects on video packet delay and jitter performance. The RTP video session has been streamed out with various initial buffering time over multi-path OpenFlow network. According to the experimental results, we observe that the proper chunk size ratio and initial buffering time need to be considered carefully to improve the mean/standard deviation of packet delay. Moreover, it is recommended to use the high performance machine for middle-box processing since the maximum possible packet scheduling rate of middle-box depends directly on computer hardware specifications. As for the future work, we will extend our experiments into international OF@TEIN testbed [6] and study the buffering effects over stochastic bandwidth capacity and link delay.

Acknowledgment

This research has been supported by the Special Task Force for Activating Research (STAR) Funding in Wireless Network and Future Internet Research Group, Chulalongkorn University, Thailand, and is based on the collaborations of OF@TEIN project. The first author wishes also to acknowledge the scholarship for International Graduate Students in ASEAN Countries, Chulalongkorn University.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, "OpenFlow: enabling innovation in campus networks," *Proceedings of ACM SIGCOMM Computer Communication Review*, pp. 69-74, Seatle, WA, USA , August 17-22, 2008.

[2] O. M. E. Committee, "Software-defined networking: The new norm for networks," *ONF White Paper*, Palo Alto, US, April 13, 2012.

[3] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath live streaming via TCP: scheme, performance and benefits," *Proceedings of ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, pp. 25, 2009.

[4] P. Panwaree, J. Kim and C. Aswakul, "Packet delay and loss performance of streaming video over emulated and real OpenFlow networks," *Proceedings of 29th International Technical Conference on Circuit/ Systems Computers and Communications (ITC-CSCC)*, pp.777-779, Phuket, Thailand, July 1-4, 2014.

[5] R. Pereira, and E. G. Pereira, "Client buffering considerations for video streaming," *Proceedings of Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on IEEE*, pp. 595-600, 2013.

[6] Z. A. Qazi, C. C. Tu, L. Chiang, R. Miao, V. Sekar, M. Yu, "SIMPLE-fying middlebox policy enforcement using SDN," *Proceedings of ACM SIGCOMM Computer Communication Review*, pp.27-38, ACM, August 2013.

[7] J. Kim, B. Cha, J. Kim, N. L. Kim, G. Noh, Y. Jang and S. M. Kang, "OF@TEIN: An OpenFlow-enabled SDN testbed over international SmartX rack sites," *Proceedings of the Asia-Pacific Advanced Network*, APAN, pp. 17-22, Daejeon, South Korea, August 19-23, 2013.

[8] P. M. Thet, P. Panwaree, J. Kim and C. Aswakul, "Design and functionality test of chunked video streaming over emulated multi-path OpenFlow network," *Proceedings of 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Hua Hin, Thailand, June 24-26, 2015.