Journal of Science & Technology 131 (2018) 069-075

SDN-based Adaptive Routing for Video Streaming over HTTP

Pham Hong Thinh [1,2], Nguyen Duc An [1], Tran Duc Tan [1], Truong Thu Huong[1]*
Pham Ngoc Nam [1]
[1]*Hanoi University of Science and Technology, No. 1, Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam*
[2]*Quy Nhon University, Binhdinh, Vietnam*
Received: May 28, 2018; Accepted: November 26, 2018

Abstract

OpenFlow/Software-Defined Networking (SDN) is a new networking paradigm that virtualizes network infrastructure by decoupling the control and data plane logic of traditional network devices. The controller of SDN has the overall look about network topology and hence provides flexibility to network operators to implement its own routing approaches. However, it could not control the way client works. In this paper, we propose a bitrate adaptation algorithm at the client side in real time combined with a SDN-based dynamic path selection for HTTP-based video streaming. Experimental results show that the proposed method can provide users with better Quality of Experience (QoE) with lower number of rerouting times, higher average quality and smoother video quality than existing non-SDN methods under different scenarios of bandwidth fluctuations.

Keywords: Adaptive streaming, Adaptive routing, DASH, SDN.

1. Introduction

In recent years, video streaming transmission with suitable bitrate like HTTP adaptive streaming (HAS) has become one of the standards for over-the-top video streaming. In HAS, the rate adaptation algorithm is deployed at the client side. In each server, video content is encoded in many different quality levels and divided up in many different segments. Besides, the client could select a qualitative level which is suitable for each video segment depending on network conditions. At present, in the system of Dynamic Adaptive Streaming over HTTP (DASH) [1], the main operation mechanism is request-respond. After each segment has been downloaded, the client requests the server to select a suitable version for the next segment and the server will respond a segment corresponding to that version to the client. Normally, video segments are set to a fixed-duration from 2s to 10s [2]. In [3], the authors find the adaptation operation corresponding to an adapted version which obtains the maximum utility and the initial delay that is lower than the delay constraint. The utility of the video streams is computed using a utility model that takes into account impacts of both video perceptual quality and the initial delay [4]. In some recent studies [5], [6], [7] various adaptation methods have been proposed. In [8], the authors investigate typical adaptation methods in the context of live video streaming. Although most previous studies have achieved success in bitrate adaptation of video streaming over HTTP, collaborative dynamic routing in network has not been mentioned.

On the other hand, in the recent tendency, OpenFlow/Software-Defined Networking [10] has been changing the present network architecture by decoupling control and data plane. SDN allows to control, manage, and supervise with a concentrate and logical way in order to bring out a flexible platform. After that, different routing algorithms have been developed to satisfy various applications of users. In SDN, the controller is the main control unit which is a key component of a network. It can be linked to any units in the topological network so it is easy to determine a routing path. Hence, it easily sends and receives related information to switching devices. The controller and the switches communicate to each other via the OpenFlow [11] protocol in which the controller instructs the switches by simply updating their flow table. A part from adding flow table entries, the controller can also delete/modify existing flow entries. It can also ask for real-time traffic statistics using the OpenFlow protocol.

A Video over Software-Defined Networking (VSDN) solution is presented in [12] in which a network architecture is proposed to select an optimum path among multiple paths, increasing ability of the Internet to provide end-to-end quality of service (QoS) for real time interactive video applications. X. Jin et al. [13] presents a line routing mechanism for transferring data among switches by a controller. They also define a rule for determining a

* Corresponding author: Tel.: (+84) 915.114.425
Email: phthinh@ftt.edu.vn

routing line for respective data flows depending on topological condition or changing network.

In [14], Hilmi et al. employ a scheme to classify data flows into three types, which are QoS level-1, QoS level-2 flows, and best-effort flow, and define different routing rules for each of them. In [15], the authors propose an adaptive routing approach for video streaming with QoS including base-layer packets and enhancement-layer packets of video bit streams. In this approach, the packet loss rate of the QoS level-1 flow is 77.3% which is lower than that of OpenQoS [14] when the network load increases and enhances at least 51.4% of coverage under various network loads for the shortest path and the feasible path. However, the two previous studies determine suitable data flows at only two quality levels and best-effort flow. Moreover, the context of dynamic adaptive video streaming at the client side has not been yet considered. Our approach combines the video quality adaption with dynamic path routing.

In the literature, path assignment frameworks for improving the performance of adaptive HTTP streaming over SDN have been proposed in [13] and [16]. However, in these methods the controller routes the path after the client has completely downloaded every video segment. This leads to the controller being overloaded when great number of clients accesses at the same time or when the network load increases.

Table 1. Comparison of related work with the proposed method

Approach	SDN architecture	Adaptive bitrate streaming	Dynamic routing
[2], [5], [7]	No	Yes	No
[14], [15]	Yes	No	Yes
[13], [16]	No	Yes	Yes, rapid-fire routing
Proposed	Yes	Yes	Yes, with flexible rerouting mechanism

Besides, in order to help the controller to reduce its workload, in this paper we combine all advantages of both HAS and SDN techniques at the controller. The client side only determines its buffer, throughput and demands the network to provide the most suitable video version. Meanwhile, the SDN controller with a global view of the whole network topology only specifies network states to provide the most suitable routing path to the client. Our proposed solution not only adapts bitrate for the client, but also finds an optimal routing path in SDN. In our work, if the network could afford a certain throughput, the client does not need to ask the SDN controller to select a new routing path. It will select the most suitable video version for itself depending on the present throughput and current buffer level. Conversely, if the buffer and throughput at the client side are reduced, the client will request the SDN controller to provide a new suitable routing. A routing path between the client and the server is chosen by the controller depending on the available bandwidth, jitter, and link delay. The main goal of this research is try to maintain video quality as good and smooth as possible in order to provide the users with the best possible Quality of Experience (QoE). Table 1 shows comparison of related work with the proposed method.

The rest of the paper is organized as follows. In Section 2, we present the principles of our method as well as the description of the adaptation algorithm at the client side and dynamic path routing over SDN. The experimental results and discussion are given in Section 3, and finally Section 4 concludes the paper.

2. Problem formulation

As can be seen in Fig. 1 [12], the SDN controller is a key component of the network, which has an important task of deciding the transition of data flows automatically. In addition, the controller also controls and administers all different data flows as well as supplies flexible directions to build routing rule in the network. Furthermore, because of the centralized management, the controller always has an overview of the network topology and can modify its routing path based on a certain network state. Constraint based routing algorithms and implementations have been studied in detail in [17].

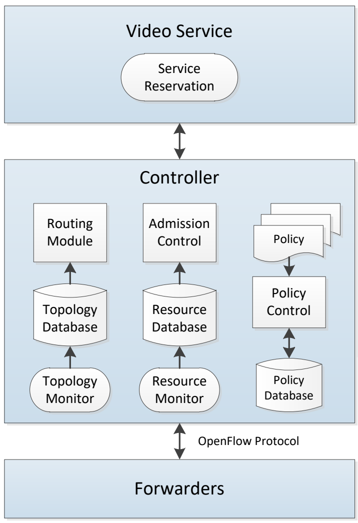


Fig. 1. Video over software defined networking (VSDN) architecture [12].

In this paper, we propose a system which does not only adapt bitrate for a client over HTTP but also finds an optimal routing path in SDN for video streaming. Normally, the client and the server exchange information together over HTTP by the request-response rule in order to select the most suitable version. Hence, when the throughput at the client-side decreases to a certain level which causes the client to satisfy with a corresponding lower version, the client will send a request to the SDN controller in an emergency manner. Immediately, the controller analyzes, evaluates and provides a new optimal routing path for the client.

Table 2. Symbols used in the paper

Symbol	Description
B_i	Buffer level at current time.
B_{low}	The low buffer threshold.
B_{high}	The high buffer threshold.
B_{max}	The buffer size.
i	The current segment index.
T_i	The actual throughput of segment *i*.
T_i^e	The estimated throughput for segment *i*.
I_i	The version of segment *i*.
P_{i+1}	The optimal path re-routing.
R_n	The bitrate of version *n*.

For video streaming with alternative CBR (Constant Bitrate) bitrates over HTTP, our purpose is to determine a version which can provide a smooth video and a good buffer stability. Table 2 shows symbols and definitions used in this paper.

To select the version for the next request, one way is to estimate the throughput based on the throughput of previous received segments. In this article, we follow the adaptive algorithm over buffer and throughput as presented in [5] where the throughput of the last segment is used as the estimated throughput.

$$T_{i+1}^e = T_i \quad (1)$$

For the new bitrate decision, we divide the method into three cases: switch-up, stable and switch-down that correspond to the situation where the client switches up, maintains, and decreases the version, respectively. Accordingly, we divide the buffer into three ranges with three thresholds from B_{low} to B_{max} ($B_{low} < B_{high} < B_{max}$).

In the "switches up" case, if the current buffer B_i is greater than or equal to B_{high} and the estimated bitrate of the next version at the next segment is less than the estimated throughput, we will increase the version; otherwise, the client will maintain the current version.

As for the "stable" case, when the current buffer B_i is in the range from ($B_{low} < B_i < B_{high}$), the current version will be maintained when its bitrate at the next segment is less than estimated throughput; otherwise, the version is switched down by one version.

Finally, the switch–down case is concerned with the following condition. When the current buffer B_i is less than B_{low}($B_i < B_{low}$) or the throughput remains its value less than T^0 in the next 5 seconds, we select the version which reduces to the lower level compared with the current version and its bitrate at the next segment is less than the estimated throughput. In this case, the client will notify the SDN controller about the client and bitrate situation. The controller computes an optimal path for transmitting the video. The details of the proposed method are shown in Algorithm 1.

In the switch-down case, when the client requires a video streaming service, it will ask for a connection to the nearest switch. Instantaneously, the switch will check its flow-table. If there is no corresponding entry in the switch's flow table, the switch will encode required information in a *PacketIn* message (which has been defined in OpenFlow) and then directing it to controller. The next step is that, the controller will analyze the message and check the current network state as well as video streaming scheme including the video server and routing line supplied for the client.

Algorithm 1. Adaptation Algorithm

Input B_{low}, B_{high}
Output I_{i+1}, P_{i+1}

//Switch-up case
if $B_i \geq B_{high}$ **then**
  if $R_{I_i+1}^e < (1-\mu)T_{i+1}^e$ **then**
  //where μ is a safety margin in the range [0, 1]
    $I_{i+1} = I_i + 1$
  else
    $I_{i+1} = I_i$
  end if
//Stable case
else if $B_i \in [B_{low}, B_{high})$ **then**
  if $R_{I_i+1}^e < (1-\mu)T_{i+1}^e$ **then**
    $I_{i+1} = I_i$
  else
    $I_{i+1} = I_i - 1$
//Switch-down case
else $B_i < B_{low}$ or $T_i^e < T^0$
  //Request for a new path in SDN
  for ($m = I_i$; $m \geq 1$; m--)
    if $R_m < (1-\mu)T_{i+1}^e$ **then**
      $I_{i+1} = m$
    end if
  end for
end if

To find the most suitable path, three constraint parameters including the available bandwidth, delay, and jitter as presented in [17] are considered. After finding an optimal routing path, the controller passes routing information to the corresponding switches to set up new selected routing via *FlowMod* messages which is defined in OpenFlow protocol. If the new path differs from the previous path, it sends the corresponding forwarding rules to the switches. Otherwise, when the controller cannot find a better path, it notifies the client and the client will try to keep the quality of the video as high as possible.

3. Performance evaluation

3.1. Experiment setup

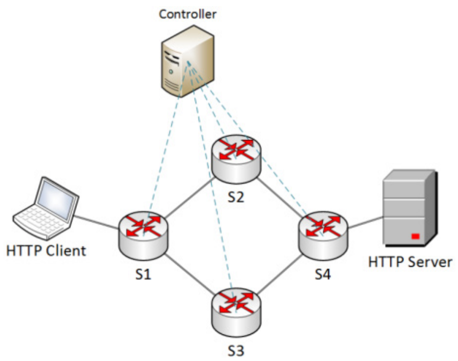


Fig. 2. Experimental setup with exemplary topology.

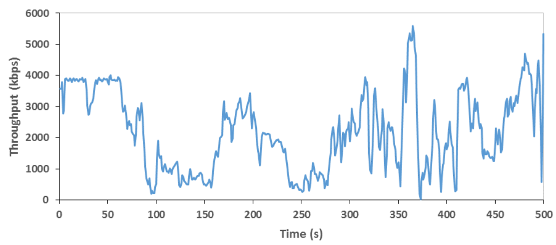


Fig. 3. Bandwidth trace is used in experiment [22].

Fig. 2 illustrates our testbed setup comprising a video HTTP server and a HTTP client operating over a virtual network. To provide the variation and complexity that can occur in a deployment network, we emulated the transportation network using Mininet [18]. The SDN controller is implemented using the POX [19] software. On the server side, an HTTP enabled server is installed in Ubuntu 14.04. The client is run in Java environment on a Windows 7 note book with 2.4 GHz and core i5 CPU. The channel bandwidth is simulated using DummyNet [20]. As we focus on on-demand streaming, the buffer size of the client (B_{max}) is set to 30s (i.e., 30 segment durations). For comparison, the reference method which is aggressive method [7] is implemented. In the aggressive method, the bitrate is decided as the highest bitrate that is lower than estimated throughput.

The experiments are implemented with buffer thresholds (B_{low} =*15s,* B_{high} =*25s*). The test video consists of 500 segments and is encoded into 16 bitrate versions. All video segments have the same length of 1 seconds. The video bitrate set R is referred to DASH dataset [21], i.e., R = {200, 250, 300, 400, 500, 600, 700, 900, 1200, 1500, 2000, 2500, 3000, 4000, 5000, 6000}.

In our experiments, two bandwidth scenarios are investigated: a simple bandwidth and a complex bandwidth obtained from a real mobile network (Fig. 3).

3.2. Simple bandwidth case

In bitrate adaptation for video streaming, one of the most important cases is when the throughput drops suddenly. In order to observe the client behavior more clearly in this case, we run the experiments with a simple bandwidth which has two levels of bandwidth, 600 kbps and 2500 kbps. The experiments are done in two cases, namely adaptive video streaming over protocol HTTP without SDN (i.e., the reference aggressive adaptation method) and with SDN (i.e., our proposed method).

In the first case “adaptive video streaming over protocol HTTP without SDN”, when the current buffer B_i is less than B_{low}=*15s*, the client will select a version which satisfies two following conditions. The first condition is that its bitrate has to be the highest and the second one is that the chosen bitrate must be lower than the throughput predicted previously.

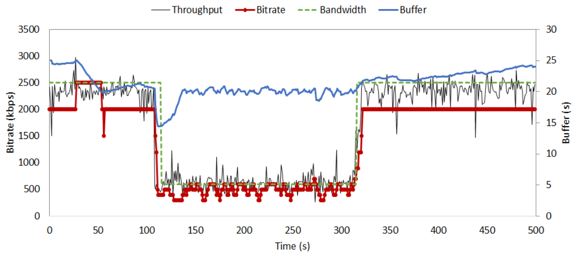
In the second case “adaptive video streaming over protocol HTTP with SDN”, when the current buffer B_i is less than B_{low}=*15s* or when the throughput remains its value less than T^0=1000kbps in the next five seconds, the client will ask for a new routing path.

3.3. Complex bandwidth case

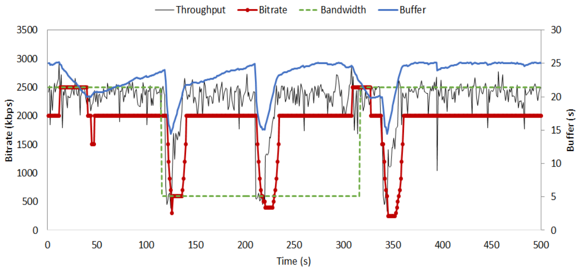
Next, we evaluate the adaptation method under a time-varying bandwidth obtained from a mobile network [22]. The bandwidth demonstrates a mobile network with strong fluctuations, from 100kbps to 6000kbps (Fig. 3). The experimental results are shown in Fig. 5.

3.4. Discussion

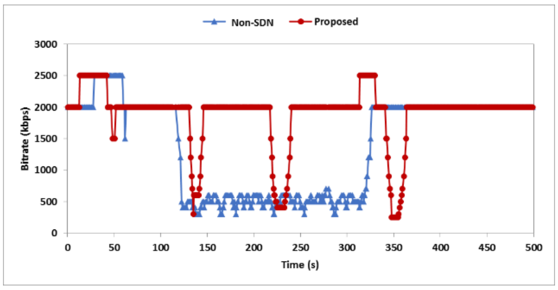
In Fig. 4-5, we show the simulation results for transmitting adaptive video stream over the HTTP protocol in the cases of with and without dynamic routing mechanism exploited in the SDN controller under different scenarios of bandwidth fluctuations.



a) Adaptation results over HTTP without SDN.



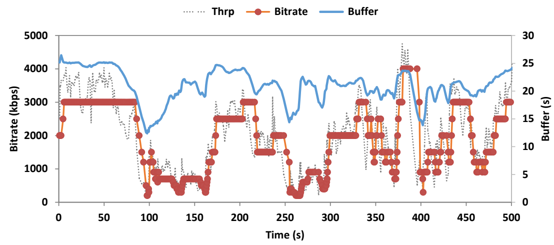
b) Adaptation results over HTTP with SDN.



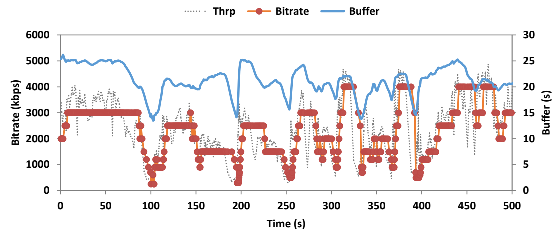
c) Resulting bitrate level of two methods.

Fig. 4. Comparisons of the two methods with a simple bandwidth trace.

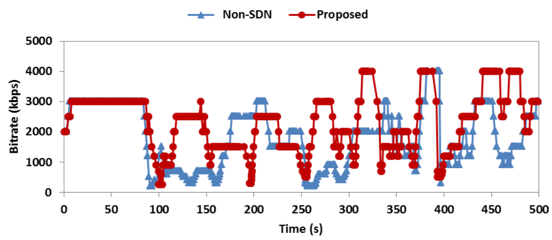
When compared with previous client based adaptive video streaming methods as presented in [2], [5] and [7], our results show that the video performance has been significantly improved in the case where the client buffer is less than $B_{low} = 15s$. In addition, when the throughput decreases, the selected version of our method is usually higher than or equal to that of the other methods. In both simple and complex bandwidth cases, the adaptation results have been computed and represented in table 3-4 and the Fig. 4-5. The table values compare the HTTP-based adaptive methods without SDN-based routing and our proposed method in terms of the number of quality switches, the average bitrate, the lowest bitrate and the buffer levels. As can be seen from the table, our results provide a much higher average bitrate. Besides, the number of quality switches is lower while the average and lowest buffer level are almost the same. In the simple case, the lowest bitrate figure as can be seen from the table 3 of the proposed method is smaller than that of the aggressive one due to abrupt changes in the available bandwidth. However, in real network condition, our method still maintain the exceeding worst case of video version.



a) Adaptation results over HTTP without SDN.



b) Adaptation results over HTTP with SDN.



c) Resulting bitrate level of two methods.

Fig. 5. Comparisons of the two methods with real bandwidth trace.

Table 3. Statistics of adaptation results in the simple case

Metrics	Non-SDN methods	Proposed method with SDN
Average bitrate	1393	1881
Number of switch down	44	21
Average buffer	21246.7	22968.9
Lowest buffer	14414	14431
Lowest bitrate	300	250

Table 4. Statistics of adaptation results in the complex case

Metrics	Non-SDN methods	Proposed method with SDN
Average bitrate	1777.4	2139.3
Number of switch down	60	54
Average buffer	20972.1	21216.3
Lowest buffer	12392	13485
Lowest bitrate	200	250

The proposed adaptive algorithm is the combination of both HTTP and SDN technologies to improve the quality of video through parameters that affect the quality of experience (QoE) of clients. As in table 3 and table 4, the *Average Bitrate* of the proposed method can reach 1881kbps and 2139.3kbps, while the non-SDN methods can reach only 1393kbps and 1777.4kbps, respectively. The *Number of Switch Down* in the proposed method is only 21 (as illustrated in Table 3) and 54 (in Table 4), while for the non-SDN methods is 44 (in Table 3) and 60 (in Table 4). So the Quality of Experience (QoE) here can be considered as proportional to the *Average Bitrate* and inversely proportional to the *Number of Switch Down*.

According to the re-routing scheme for segment as in [16], a video was divided into 500 segments which take 500 rerouting times to operate during 500s. This implies that the controller has to work continuously at every segment. Hence, the controller might be overloaded when large number of clients accesses at the same time. On the contrary, in our method, the number of re-routing is very low (i.e., 3) even there is no rerouting if the buffer is more than $B_i \geq 15s$.

4. Conclusion and future work

In this paper, we have presented a solution for enhancing the quality of video by combining video bit-rate adaptation algorithm and dynamic routing method for adaptive streaming over a SDN network. We also combine the advantages of both HAS and SDN techniques in streaming video over the Internet. Besides, another advantage of our method is that it helps to reduce the load of the SDN controller. The experimental results have shown that the proposed method provides the user with a better QoE in terms of the number of quality switches and the average video bitrate in comparison with other existing methods.

In the future, we could widen the network scale and increase the number of clients access with different resolutions. By using algorithm combination, the dynamic traffic shaping problem when there are multiple clients competing for access to a shared bottleneck link or accessing on a gateway in home networks to improve video streaming will be solved with fairness, efficiency and quality for all clients.

Acknowledgments

Our thanks to the Vietnam National Project, code DTDLCN.38/15 for supporting this research.

References

[1] T. Stockhammer, Dynamic Adaptive Streaming over HTTP – Standards and Design Principles, Proc. Second Annu. ACM SIGMM Conf. Multimed., no. i, pp. 133–143, 2011.

[2] T. C. Thang, H. T. Le, H. X. Nguyen, A. T. Pham, J. W. Kang, and Y. M. Ro, Adaptive Video Streaming over HTTP with Dynamic Resource Estimation, J. Commun. Networks, vol. 15, no. 6, pp. 635–644, 2013.

[3] D. V Nguyen, D. M. Nguyen, H. T. Tran, N. P. Ngoc, A. T. Pham, and T. C. Thang, Quality - Delay Tradeoff Optimization in Multi - Bitrate Adaptive Streaming, 2015 IEEE Int. Conf. Consum. Electron. (ICCE), Las Vegas, vol. 0, pp. 66–67, 2015.

[4] D. V. Nguyen, H. T. Le, A. T. Pham, T. C. Thang, J. Y. Lee, and K. Yun, Adaptive home surveillance system using HTTP streaming, 2013 Int. Jt. Conf. Aware. Sci. Technol. Ubi-Media Comput. Can We Realiz. Aware. via Ubi-Media?, iCAST 2013 UMEDIA 2013, pp. 579–583, 2013.

[5] T. C. Thang, Q. Ho, J. W. Kang, A. T. Pham, and S. Member, Adaptive Streaming of Audiovisual Content using MPEG DASH, IEEE Trans. Consum. Electron., vol. 58, no. 1, pp. 78–85, 2012.

[6] S. Akhshabi, A. C. Begen, and C. Dovrolis, An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP, MMSys '11 Proc. Second Annu. ACM Conf. Multimed. Syst., pp. 157–168, 2011.

[7] D. V Nguyen, H. T. Le, P. N. Nam, and T. Anh, Adaptation Method for Video Streaming over HTTP / 2, IEICE Commun. Express, vol. 1, no. May, pp. 1–6, 2015.

[8] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, An evaluation of bitrate adaptation methods for HTTP live streaming, IEEE J. Sel. Areas Commun., vol. 32, no. 4, pp. 693–705, 2014.

[9] H. T. Le, N. P. Ngoc, T. A. Vu, A. T. Pham, and T. C. Thang, Smooth-bitrate adaptation method for HTTP streaming in vehicular environments, IEEE Veh. Netw. Conf. VNC, vol. 2015–Janua, no. January, pp. 187–188, 2015.

[10] Open Networking Foundation (ONF), 'Software Defined Networking: the new norm for network', White paper, 2012.

[11] N. McKeown et al., OpenFlow: Enabling Innovation in Campus Networks, ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, p. 69, 2008.

[12] H. Owens and A. Durresi, Video over Software-Defined Networking (VSDN), Comput. Networks, vol. 92, no. December, pp. 341–356, 2015.

[13] X. Jin, H. Ju, S. Cho, B. Mun, C. Kim, and S. Han, QoS Routing Design for Adaptive Streaming in Software Defined Network, 2016 Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS), Phuket, pp. 1–6, 2016.

[14] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, An

optimization framework for QoS-enabled adaptive video streaming over openflow networks, IEEE Trans. Multimed., vol. 15, no. 3, pp. 710–715, 2013.

[15] T. Yu, Adaptive Routing for Video Streaming with QoS Support over SDN Networks, 2015 Int. Conf. Inf. Netw. (ICOIN), Cambodia, pp. 318–323, 2015.

[16] C. Cetinkaya, E. Karayer, M. Sayit, and C. Hellge, SDN for Segment based Flow Routing of DASH, 2014 IEEE Fourth Int. Conf. Consum. Electron. Berlin, pp. 74–77, 2014.

[17] O. Younis and S. Fahmy, Constraint-based routing in the internet: Basic principles and recent research, IEEE Commun. Surv. Tutorials, vol. 5, no. 1, pp. 2–13, 2003.

[18] B. Lantz, B. Heller, and N. McKeown, A Network in a Laptop: Rapid Prototyping for Software-Defined Networks, Proc. Ninth ACM SIGCOMM Work. Hot Top. Networks - Hotnets '10, pp. 1–6, 2010.

[19] POX. [Online]. Available: http://www.noxrepo.org/pox/about-pox/.

[20] L. Rizzo, Dummynet: A Simple Approach to the Evaluation of Network Protocols, ACM SIGCOMM Comput. Commun. Rev., vol. 27, no. 1, pp. 31–41, 1997.

[21] S. Lederer, C. Müller, and C. Timmerer, Dynamic Adaptive Streaming over HTTP Dataset, Proc. ACM MMSys'12, pp. 89–94, 2012.

[22] C. Müller, S. Lederer, and C. Timmerer, An evaluation of dynamic adaptive streaming over HTTP in vehicular environments, Proc. 4th Work. Mob. Video - MoVid '12, p. 37, 2012.