A SDN-Based Solution Towards Smooth Adaptive Playback for Dynamic Video Streaming over HTTP

Mickaël Cunha
ISCTE – Instituto Universitário Lisboa,
Instituto de Telecomunicações
Lisboa, Portugal
mickael_cunha@iscte-iul.pt

Jose Moura
ISCTE – Instituto Universitário Lisboa,
Instituto de Telecomunicações
Lisboa, Portugal
jose.moura@iscte-iul.pt

Paulo Nunes
ISCTE – Instituto Universitário Lisboa,
Instituto de Telecomunicações
Lisboa, Portugal
paulo.nunes@iscte-iul.pt

***Abstract*— This paper studies the scenario of providing smooth adaptive playback for dynamic video streaming over HTTP in constrained networks. We propose and evaluate a bandwidth control loop programmed into a new management service operating over an SDN controller. Experimental results based on emulation show that the proposed low-complexity solution reduces the number of times a DASH video client switches among the diverse video representations, thus enhancing the video quality presented at each client, despite unfair competitive UDP traffic.**

Keywords—video streaming, SDN, DASH, QoE

I. Introduction

Considering the Internet's traffic, the video streaming is already responsible for a significant amount of the total traffic [1]. In addition, a critical factor in today's video traffic ecosystem is the user perceived Quality of Experience (QoE) [2]. There are various factors that strongly influence the QoE, namely: i) video bitrate; ii) rebuffering rate; and iii) switching rate among video versions. Therefore, it is very important to provide an adequate QoE to the end-user, essentially when the network is congested [3].

Several technologies have been developed to support a fair video distribution with adequate QoE. From these, Dynamic Adaptive Streaming over HTTP (DASH) [4] is widely used. It offers several benefits:

- The protocol was developed over HTTP especially because HTTP can solve issues regarding Network Address Translation and other middleboxes [5];
- HTTP has become the main protocol on the Web [6]. In addition, by allowing DASH servers to use Web servers, the operational costs are reduced; and, by using hierarchical web caching, further performance gains are achieved, such as small data access latency and traffic reduction on the access link [4];
- Relying on TCP fairness, it guarantees that streaming traffic consumes only a fair fraction of the network bandwidth when sharing it with other traffic [4].

Although it represents a great advance in video streaming, it has been reported that DASH applications use estimation tools that are inaccurate [7][8]. In addition, DASH algorithms tend to be unstable, especially when sharing a network bottleneck among a significant number of users. Therefore, novel solutions are required to enhance DASH streaming.

In the current work, we use the Software Defined Networking (SDN) paradigm [9] to enhance DASH streaming in networking scenarios with constrained available resources. The research question that serves the main basis for our work is as follows: "Can we smooth the user-side control loop of DASH with a second network-side control loop based on the SDN paradigm?". To answer this, we designed, implemented, and tested a low-complexity SDN-based solution, which aims offering a smooth adaptive playback for DASH video streaming. Our paper has the following organization: after this introduction; Section II discusses the background work, Section III describes our proposal, including its implementation and evaluation; and finally, Section IV concludes the manuscript.

The authors acknowledge the support of Fundação para a Ciência e Tecnologia, under the project UID/EEA/50008/2019.

II. Background Work

A. DASH Protocol

The DASH protocol was designed to provide adaptive streaming. In this way, the video playout is better adjusted to eventual network constraints. The video adjustment is made in two dimensions, time and spatial. The time adjustment is achieved by rate control. The spatial adjustment is obtained by change of video resolution. To enable these video adaptive functions, each video is encoded in several versions, e.g., with different bitrates. Then each video version is divided into several chunks. All chunks of the same bitrate are part of the same representation.

When the clients use DASH to receive video streaming, they can select among several video versions. The information about these video versions is stored within a Media Presentation Description (MPD) file. The MPD is an XML document that instructs the DASH client in how to retrieve each video version. In this way, the information available in the MPD file includes video versions, metadata, codecs and a list of servers identified by Uniform Resource Locators. Consequently, each DASH client can choose a server from which that client can retrieve video chunks associated to any available video version.

After receiving the MPD from the server, the DASH client decides which representation it wants to visualize and requests the according chunks through ordinary HTTP GET Requests. The DASH protocol does not control the video transmission rate in a very precise way. To achieve that, the video transmission rate should be regulated by the TCP layer [4].

After downloading the initial chunks, the DASH client keeps on periodically downloading new chunks in a chronological order throughout the session and typically buffers some chunks to facilitate the playback and to avoid/reduce rebuffering. During the session, the client playout can switch among video versions, aiming to maximize the visualized video quality. In this way, the DASH client senses some video parameters, choosing the video version more suitable for the aimed QoE. Therefore, the DASH protocol has the following advantages:

- Playback can be adjusted according to a target QoE;

- Video version switching without stopping playback;
- Chunks are retrieved from different servers, giving more flexibility to use available network resources.

However, when the network resources are constrained, DASH cannot directly reconfigure the network resources to obtain a satisfactory video throughput. Therefore, DASH needs some external contribution to solve these limitations at the signal processing level [10][11], or at the network management level. The following sub-section discusses how SDN can successfully manage DASH video streaming in usage scenarios with constrained network resources.

TABLE I. LIST OF RELATED WORK

Related Proposal	Main Characteristic(s)
SDN for Segment based Flow Routing of DASH [12]	Implements an SDN structure that provides path selection to increase the throughput of DASH video.
Design and Experimental Evaluation of Network-assisted Strategies for HTTP Adaptive Streaming [13]	Introduces a Video Control Plane which enforces Video Quality Fairness among concurrent video flows generated by heterogeneous client devices.
An SDN Architecture for Privacy-Friendly Network-Assisted DASH [14]	Implements a Service Manager that has two different adaptation mechanisms that bring more stability and a higher bitrate.
Towards Network-wide QoE Fairness Using OpenFlow-assisted Adaptive Video Streaming [8]	Proposes an OpenFlow-assisted QoE Fairness Framework that aims to fairly maximise the QoE of multiple competing clients in a shared network environment.
A Scalable User Fairness Model for Adaptive Video Streaming Over SDN-Assisted Future Networks [15]	Develops a model based upon three metrics: video quality, switching impacts and cost efficiency.
SDN Based QoE Optimization for HTTP-Based Adaptive Video Streaming [16]	Uses an SDN infrastructure to build a more efficient and intelligent video delivery network that evaluates both the video content complexity and the playout buffer status.
SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking [17]	Provides an SDN architecture that manages and allocates the network resources dynamically for each client based on its expected QoE.

B. Enhancing DASH Video Streaming using SDN

Using the SDN paradigm, it is possible to describe the system upon three planes (Fig. 1): i) control plane where the controller is located; ii) data plane, where the forwarding devices are located (these devices communicate with the controller through the southbound API); and iii) management plane where the network applications are located (these communicate with the controller via the northbound API).

Adopting this three-layered approach, more flexible networks are deployed, offering the capability of self-adjustment to significant topology changes. The SDN adjusts flow data paths to the unexpected challenges imposed by previous changes. Therefore, the final goal is to guarantee that the network resources are efficiently used. It is also possible to program intelligent algorithms that avoid the continuous surveillance of network managers.

C. Related Work

Table I summarizes some relevant contributions proposing SDN-based solutions to enhance DASH video streaming. From these, [14] is the proposal more aligned with our current contribution. Nevertheless, our solution is different from [14] since our solution does not require a dedicated communication channel between each client and the Service Manager. In addition, according to our best knowledge, the literature lacks a comprehensive study that highlights the individual positive contribution of either SDN or DASH to enhance video streaming.

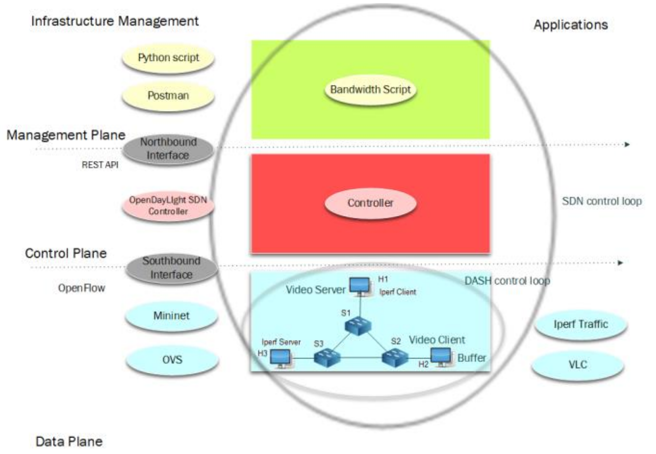

Fig. 1. Proposed System Architecture

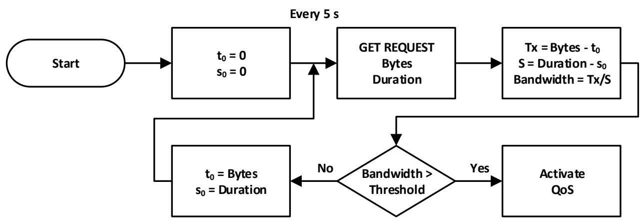

Fig. 2. Flowchart of Bandwidth Control Loop

III. PROPOSED SOLUTION IMPLEMENTATION AND EVALUATION

This section describes the proposed system, as depicted in Fig. 1. There are two control loops. The first one is managed by DASH. The second one is controlled by the SDN system. The DASH player evaluates the level of client buffering and, based on this, it requests the more adequate video version. Nevertheless, this control loop only takes the final decision based on the local client information, ignoring how the network resources are used. Our proposal uses the SDN control loop to mitigate that last limitation. This SDN control loop (Fig. 2), it is implemented via a python script. It runs over northbound API and it monitors every five seconds the capacity of each available datapah link. If that capacity drops below a threshold then the script reacts to it by installing, via the SDN controller, some flow QoS rules in the switch. These flow rules classify and shape the switch outbound traffic, according to the aimed QoS of each traffic type. For example, if the UDP traffic is discarded then the video quality of the received video can be enhanced.

A. System Testbed Implementation

The whole testing environment was developed on a Debian Virtual Machine over the VirtualBox hypervisor.

The video used in our experiments was "Big Buck Bunny". The original file was encoded into several versions, each corresponding to a different bitrate. Additionally, an appropriate MPD file was generated that describes a manifest of the available content, its various alternatives, its URL address, and other useful data for proper video playback. All video versions share the same coding parameters with the exception of the bitrate. Files were encoded using the following bitrates: 500 kbps, 1000 kbps, 2000 kbps and 4000 kbps. In addition, each video version was segmented into chunks of two seconds.

We have deployed a testbed based on SDN and DASH. In the studied scenario, we give higher priority to video stream on a network when the video suffers a very aggressive competition from UDP traffic. The strategy chosen in this implementation to protect the video is based on the traffic shaping, using switch queues, as shown in Table II. Our tests have used two types of traffic: i) video streaming, and ii) iperf to simulate non-video (i.e., UDP) traffic. We have used the network topology with three interconnected switches along the vertexes of a triangle. The links connecting switches are restricted to 20Mbps. These links are traversed by UDP traffic with 18Mbps. This implies the video streaming is restricted to a maximum value of 2 Mbps. To protect the video traffic, the UDP traffic needs to be constrained. For this, we map each type traffic into a specific queue, which guarantees a maximum rate for that traffic type.

TABLE II. CONFIGURED QUEUES

Queue	Switch	Purpose	Rate
Q1	S1	Video for H2	20 Mbps
Q2	S1	Iperf	400 kbps
Q1	S3	Video for H2	20 Mbps

The video server was implemented with the python HTTP simpleServer that is included by default in Mininet (network emulator). To activate the server, we simply have to open a terminal in H1, go to the desired path through the Command Line Interface (CLI) and then activate the server in port 80 using the command "python3 -m http.server 80&". Then, the clients could access the server simply by opening a terminal in Mininet and accessing it through the desired application. In this case, the VideoLan Client (VLC) player was used by each client to retrieve the video from the server and reproduce it.

We have run tests over three streaming scenarios. All scenarios were tested upon similar conditions: a video stream was reproduced by H2, then H1 would inject UDP iperf traffic to H3 bringing competition on the link between S1 and S3 (where the video stream to H2 also goes by). As UDP iperf tends to use all available bandwidth within a link, the different scenarios were made to study how the video stream reacts to the concurrence of the iperf traffic. In addition, we aim to investigate how DASH alone and DASH & SDN together could mitigate the unfair UDP concurrence.

B. Evaluation

This sub-section discusses the test results for the three video streaming scenarios: i) without any control mechanism; ii) with DASH only; iii) with both DASH and SDN.

1) *Scenario 1:* *no control mechanism*

This scenario has the simplest implementation as no adaption mechanism is used. Since there is no mechanism to protect video traffic from the interference of the iperf traffic, a poor video playback performance is likely. Fig. 3 shows the bandwidth of both video and iperf for the first scenario.

As expected, the results of Fig. 3 show that when the iperf traffic is injected, it starts to consume all available bandwidth and the video rate drops indicating a poor video reproduction.

2) *Scenario 2:* *only DASH control is active*

This scenario is likely to have distinct results (Fig. 4) from the initial scenario. Since the iperf traffic would still leave some available bandwidth, it is expected that the client could reproduce the video, even tough, it won't have sufficient resources to reproduce the highest quality version of the video content. The results showed that the player was trying to adapt in the best possible way its playback to the available resources (Fig. 4).

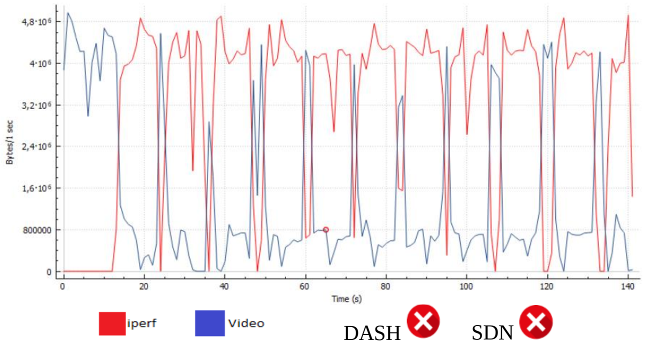

Fig. 3. Results for Scenario 1

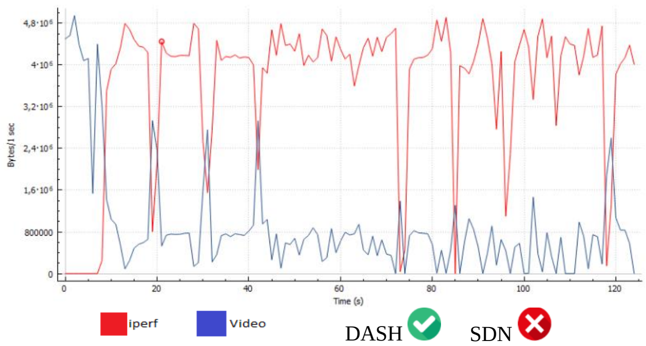

Fig. 4. Results for Scenario 2

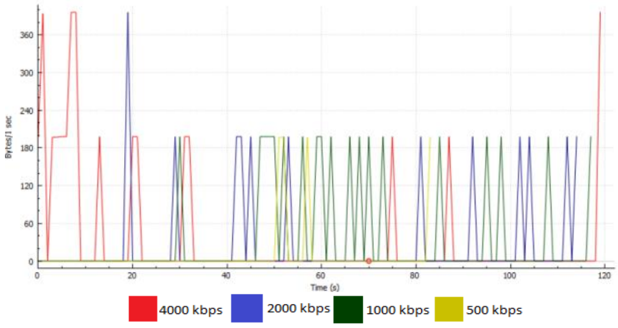

Fig. 5. Downloaded Chunks for Scenario 2

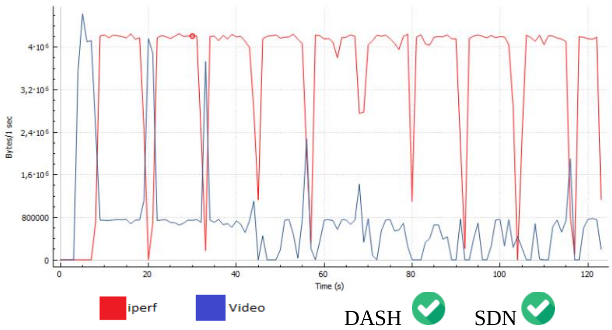

Fig. 6. Results for Scenario 3

The video versions for each downloaded chunk are shown in Fig. 5. Here, one can note a strong dominance of the middle video versions (2000 kbps & 1000 kbps). Due to the DASH protocol, the video reproduction never stopped. Nevertheless, it is possible to observe the playout of the lowest version of the video (500 kbps) in several instants along the video streaming session.

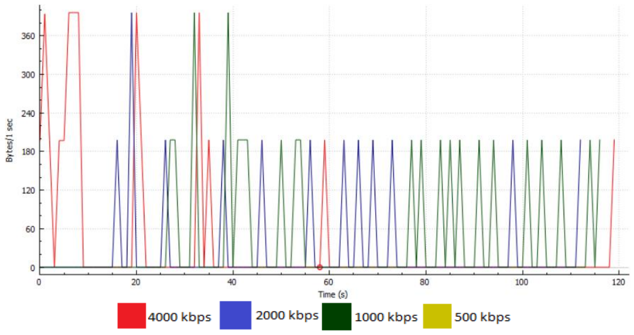

Fig. 7. Downloaded Chunks for Scenario 3

TABLE III. MAJOR RESULTS FROM OUR EVALUATION SCENARIOS

Scen.	Major Result
1	The DASH player client does not have enough resources to reproduce the video and the playout stops every time the concurrent iperf traffic was on.
2	Facing the competition of the iperf traffic the player was able to switch to lower video rate versions, including a few times to the lowest rate one, and go on with the video reproduction without any interruption on the video playout.
3	The results evidence that the SDN part of the proposed solution in cooperation with the DASH control loop can increase the stability of the video reproduction at the DASH player client by diminishing the number of times that the DASH control loop changes among the video versions, when comparing the current results with the ones of Scenario 2. In addition, and by opposition to what was seen from the results of Scenario 2, analyzing the current results, there is no evidence that the DASH client played anytime the lowest video rate (i.e. 500 kbps)

3) *Scenario 3: both control mechanisms are active*

This scenario is the one where the relevance of the DASH protocol in combination with the SDN paradigm is more evident. This scenario has a close control on the network resources thanks to the SDN controller and, it offers a more powerful management of the available network resources, in a complete distinct way from the previous scenario, where the control was made only by local client status information. Consequently, the current scenario achieves better results (Fig. 6), when compared with similar results from Scenario 2.

Analyzing the results of Fig.7, we can conclude that faced to a strong competition, the DASH player tries to readapt and achieve the best playout again. Furthermore, if we compare these results with the ones from the Scenario 2, it is possible to see that the SDN controller brings more robust control over the usage of network resources and that, even though the rate assigned to the iperf traffic is the same as in Scenario 2, the DASH player reaches more playback stability and stabilizes for a certain video version during larger periods. Table III summarizes the major results of our evaluation.

IV. CONCLUSIONS AND FUTURE WORK

Our work proposed a bandwidth management control loop programmed into a new service that is operating over a SDN controller. The evaluation results confirm the decrease on the number of times each DASH client switches among video versions, enhancing the quality of received video in that client.

We expect evolve our SDN-based video adaptive proposal in a similar way to a very recent work [18]. This is related with differentiated QoS and our future work will be about QoE.

REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2015-2020," Forecast Methodol., p. 22, 2017.

[2] J. Joskowicz and J. C. L. Ardao, "A parametric model for perceptual video quality estimation," 2012 IEEE Int. Symp. Broadband Multimed. Syst. Broadcast., vol. 49, pp. 49–62, 2012.

[3] R. M. Abuteir, A. Fladenmuller, O. Fourmaux, and S. Universites, "Variable-Threshold Buffer Based Adaptation for DASH Mobile Video Streaming," pp. 1–7, 2017.

[4] J. Kua, G. Armitage, and P. Branch, "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP," IEEE Commun. Surv. Tutorials, vol. 19, no. 3, pp. 1842–1866, 2017.

[5] M. Abdullah, "A Novel CDN Testbed for Fast Deploying HTTP Adaptive Video Streaming.," Proc. 9th EAI Int. Conf. Mob. Multimed. Commun., pp. 65–71, 2016.

[6] J. M. Jimenez, O. Romero, A. Rego, A. Dilendra, and J. Lloret, "Study of Multimedia Delivery over Software Defined Networks," Netw. Protoc. Algorithms, vol. 7, no. 4, p. 37, 2016.

[7] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," Proc. Second Annu. ACM Conf. Multimed. Syst. - MMSys'11, p. 157, 2011.

[8] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," Proc. 2013 ACM SIGCOMM Work. Futur. human-centric Multimed. Netw. - FhMN '13, pp. 15-20, 2013.

[9] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," Proc. IEEE, vol. 103, no. 1, pp. 14–76, 2015.

[10] P. Nunes, L. D. Soares, and F. Pereira, "Error resilient macroblock rate control for H.264/AVC video coding," in 2008 15th IEEE International Conference on Image Processing, 2008, pp. 2132–2135.

[11] P. Nunes, L. D. Soares, and F. Pereira, "Automatic and adaptive network-aware macroblock intra refresh for error-resilient H.264/AVC video coding," in 2009 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 3073–3076.

[12] C. Cetinkaya, E. Karayer, M. Sayit, and C. Hellge, "SDN for segment based flow routing of DASH," IEEE Int. Conf. Consum. Electron. - Berlin, ICCE-Berlin, vol. 2015–Febru, no. February, pp. 74–77, 2015.

[13] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, "Design and Performance Evaluation of Network- assisted Control Strategies for HTTP Adaptive Streaming," ACM Trans. Multimed. Comput. Commun. Appl., vol. 13, no. 3s, pp. 1–24, 2017.

[14] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "An SDN Architecture for Privacy-Friendly Network-Assisted DASH," ACM Trans. Multimed. Comput. Commun. Appl., vol. 13, no. 3s, pp. 1–22, 2017.

[15] M. Mu, M. Broadbent, A. Farshad, N. Hart, D. Hutchison, Q. Ni, and N. Race, "A scalable user fairness model for adaptive video streaming over SDN-assisted future networks," IEEE J. Sel. Areas Commun., vol. 34, no. 8, pp. 2168–2184, 2016.

[16] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, "SDN Based QoE Optimization for HTTP-Based Adaptive Video Streaming," Proc. - 2015 IEEE Int. Symp. Multimedia, ISM 2015, pp. 120–123, 2015.

[17] Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking," Proc. 2016 ACM Multimed. Conf. - MM '16, pp. 1296–1305, 2016.

[18] Sayit, M., Cetinkaya, C., Yildiz, H. U., Tavli, B., DASH-QoS: A scalable network layer service differentiation architecture for DASH over SDN, Computer Networks, vol. 154, pp. 12-25, 2019.