International Journal of Internet, Broadcasting and Communication Vol.12 No.4 166-172 (2020)
http://dx.doi.org/10.7236/IJIBC.2020.12.4.166

IJIBC 20-4-20

A Novel Bit Rate Adaptation using Buffer Size Optimization for Video Streaming

Young-myoung Kang

Staff Researcher, Network Business, Samsung Electronics, Suwon, Korea
kang.youngmyoung@gmail.com

Abstract

Video streaming application such as YouTube is one of the most popular mobile applications. To adjust the quality of video for available network bandwidth, a streaming server provides multiple representations of video of which bit rate has different bandwidth requirements. A streaming client utilizes an adaptive bit rate scheme to select a proper video representation that the network can support. The download behavior of video streaming client player is governed by several parameters such as maximum buffer size. Especially, the size of the maximum playback buffer in the client player can greatly affect the user experience. To tackle this problem, in this paper, we propose the maximum buffer size optimization according to available network bandwidth and buffer status. Our simulation study shows that our proposed buffer size optimization scheme successfully mitigates playback stalls while preserving the similar quality of streaming video compared to existing ABR schemes.

Keywords: *Dynamic Adaptive Streaming, Video Playback Buffer Optimization, Network Dynamics*

1. Introduction

Video streaming is currently one of the most popular Internet applications and it accounts for more than 65% of world-wide mobile downstream traffic [1]. DASH (Dynamic Adaptive Streaming over HTTP)[2], a standard proposed to cope with the Internet environment in which the amount of traffic and bandwidth is dynamic, enables video content to be serviced at various bit rates. To support a video streaming with an appropriate bit rate according to an available bandwidth, a DASH server provides multiple instances of small sized (*ex.*, several seconds long) video chunks encoded at different bit rates. On the other hand, DASH clients decide the bit rate of the video chunks based on the measured available bandwidth, and request it to a DASH server. Various ABR (Adaptive Bit Rate) methods have been proposed to select an appropriate bit rate for the network environment[3-9].

A DASH client player works based on the following 3 stages; (1) A player fills its playback buffer to the maximum level (B_{max}), (2) Starting to play the video when the buffer is filled with the minimum amount of video and continues to retrieve media chunks until the initial buffering completes, (3) Video chunks

Manuscript Received: September. 11, 2020 / Revised: September. 15, 2020 / Accepted: September. 19, 2020
Corresponding Author:kang.youngmyoung@gmail.com
Tel: +82-70-7652-7855
Staff Researcher, Network Business, Samsung Electronics, Suwon, Korea

downloading is continued until the buffer level falls below the maximum buffer level by playback. These operations of DASH clients lead to an ON/OFF traffic pattern[9].

The size of the maximum playback buffer size set in the DASH Client player can greatly affect the user experience. If a buffer of too small size is used, a playback stall may occur in a sudden bad network situation. If it is too large, traffic is wasted unnecessarily if the user stops watching in the middle (especially if you are using a mobile network, *e.g.*, LTE/5G). This can also adversely affect the user experience). However, this maximum buffer size uses a fixed value in most DASH client implementations. In this paper, we propose a method of adjusting the maximum buffer size to improve the streaming user's experience according to the predicted network conditions.

The remainder of this paper is organized as follows. Related work is reviewed in Section 2, and our approach is introduced in Section 3. Section 4 evaluates the performance of our proposed scheme with trace-driven simulations. We conclude in Section 5.

2. Related Work

A plethora of research efforts have been focused on addressing issues in video streaming under dynamic dynamics. [6] introduce a feedback controller which utilizes the buffer status to determine the bit rate, however the final decision strongly depends on bandwidth prediction. [10] extends the approach in [6] to smooth the effect of network dynamics by choosing lower bit rates than can support when larger throughput variability is observed. [11] propose a Proportional-Integral (PI) controller harnessing the playback buffer level as a feedback signal to the controller. In [3], authors introduce an ABR scheme based only on playback buffer status. It increases the buffer length if a requested bit rate is lower than the measured bandwidth, and decreases otherwise. [5] focus on undesirable bit rate fluctuations when contending clients struggle to obtain more bandwidths which results in degrading the user experiences. To remedy these issues, authors adopt a proactive provisioning mechanism estimating the target average bit rate. [7] tackles to define a streaming QoE optimization problem. Their approach however requires pre-computed information for a specific video setting which results in higher overhead.

3. Proposed Scheme

We extend the ABR selection scheme (LQE) [14] to optimize the maximum size of the playback buffer according to network status. Based on control theory using a discrete time linear quadratic regulator, LQE uses the playback buffer level as a signal, in which the controller targets the buffer level to maintain the reference buffer level q_0. Note that the maximum buffer size (B_{max}) of LQE is fixed but larger than q_0. Thus, the playback behavior of LQE is also governed by B_{max}, resulting in ON-OFF cycle as shown in [9]. Our approach aims to adjust B_{max} to enhance video streaming quality while preserving a feasible amount of buffered video chunk. The rationale behind our approach is as follows:

• If the playback buffer is filled with more amount of video chunk than q_0, ABR is likely to increase the bit rate of the next video chunks to achieve better streaming quality.

• If the expected consumption of the buffer for downloading the bit rate selected for the next chunk still maintains the larger buffer size than q_0, it means that the network bandwidth is enough to support better video quality, *i.e.*, higher bit rate.

• If the client player tries to buffer more amount of video chunks in this situation, the quality of stored video

chunks will be ones of high bit rates, resulting in overall high-quality video streaming. Also, more buffering enables the client player to avoid playback stalls due to a buffer depletion caused by future network failures.

Based on these insights, we propose the following algorithm described in Figure 1 to decide B_{max} according to the current buffer status and expected network bandwidth. Suppose that B_k is the amount of buffered video when the player starts the kth video chunk. Given the previous bandwidth history and the reference buffer level q_0, the LQE algorithm selects the bit rate of the next chunk R_{k+1}. Let $\hat{C}$ be the estimated bandwidth based on the previous bandwidth history. If the two inequalities $B_k >= q_0$ and $B_k - R_{k+1} \times L/\hat{C} > q_0$ are satisfied, it can mean that the current network bandwidth is sufficiently larger and the buffer quickly fills to the maximum level and the client continuously initiates ON-OFF patterns for chunk downloads. Thus, our algorithm increases B_{max} by the chunk length L up to B_{MAX} in order to force the client to download more, where B_{MAX} is a value to limit the increase of B_{max} since buffering large amounts can result in traffic waste if a user abandons the video without playing it back fully.

R_{k+1} = LQE_SELECT(B_k, previous bandwidth history, q_0)
IF $B_k >= q_0$ and $B_k - R_{k+1} \times L/\hat{C} > q_0$
THEN
    B_{max} = min($B_{max} + L$, B_{max})
ELSE
    If $B_k < q_0$
    Then
        $B_{max} = B_{\mathrm{MAX}}$

Figure 1. Proposed Buffer Optimization Algorithm

4. Trace-Driven Simulations

We compare the performance of LQE with buffer adjustment with following buffer-based ABRs:

• Tian: [6] drives the playback buffer level to a set-point by scaling predicted throughputs as a function of the buffer level and its trend. We use the control parameter K_p =0.1.

• BBA: [3] selects bit rate R_k based on a function of the playback buffer level. The reservoir and cushion parameters are set to 20s and 70s, respectively.

• LQE: [14] utilizes Linear Quadratic Regulator for a rate adaptation.

We conduct trace-driven simulations using a set of 85 public available traces of cellular bandwidth [13]. Assuming that a streaming device obtains sum of entries in two different traces as a bandwidth, we use combinations of the throughput traces to input bandwidth values for each interface. Note that if a trace is shorter than the simulation running time, we continue to repeat the trace from the beginning. This results in 3570 scenarios with which to investigate the performance of the streaming schemes, which is intended to reflect the bandwidth variability in real networks. In the simulation, we assume that the streaming server provides six representations of the video, of which length is 1300 seconds, with resolutions varying from 144p to 1080p, of which bit rates are from 0.27 Mbps to 8.9 Mbps. Given a bandwidth scenario, our simulator calculates streaming bit rate, playback buffer, and rebuffering statuses based on the streaming behavior model

[7].

We use following performance metrics to compare ABR schemes:

- Average bit rate: This is the average of the bit rate of all downloaded chunks, which is defined as $\frac{\sum_{k=1}^{K} R_k}{K}$ where K is the total number of chunks and R_k is the bit rate of the kth chunk.
- Average rebuffering duration: This is defined as the time spent in the rebuffering phase divided by the number of rebuffering occurrences during the entire playback.
- Number of bit rate changes: We count the number of times the bit rate increases or decreases during the entire playback.

Figures 2, 3, and 4 present Whisker plots showing the minimum, first quartile (Q1), median, third quartile (Q3), and maximum of average bit rate and average rebuffering duration and CDF of number of bit rate changes. As shown in Figure 3, LQE and LQE-buf yields shorter average rebuffering duration than Tian and BBA: in particular, the maximum average rebuffering duration of LQE and LQE-buf is less than six seconds while those of Tian and BBA are longer than 27 seconds. As with the results of [14], LQE exhibits lower average bit rates (2.18 Mbps in terms of median) than BBA that shows the best average bit rate (2.26 Mbps in terms of median) in Figure 3. However, LQE-buf achieves the average bit rate (2.21 Mbps in terms of median), which is close to the average bit rate of BBA that experiences longer rebufferings. This is because LQE-buf tries to buffer video chunks with higher bit rates more aggressively by increasing the maximum buffer size when the network bandwidth is sufficiently large. In Figure 4, we observe that LQE-buf yields more bit rate changes than Tian and BBA as LQE does since LQE-buf is based on the LQE algorithm.

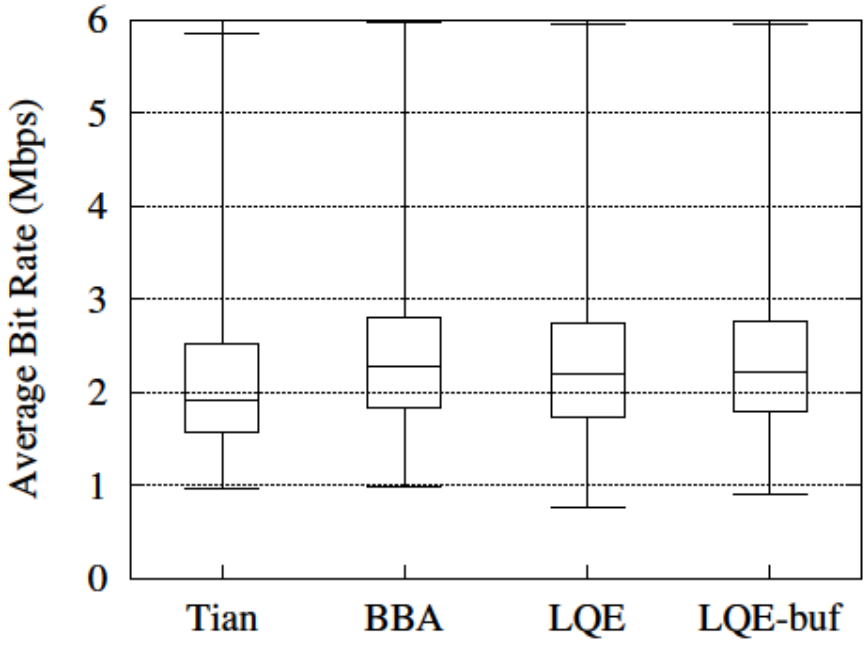

Figure 2. Whisker plot of Average Bit Rate

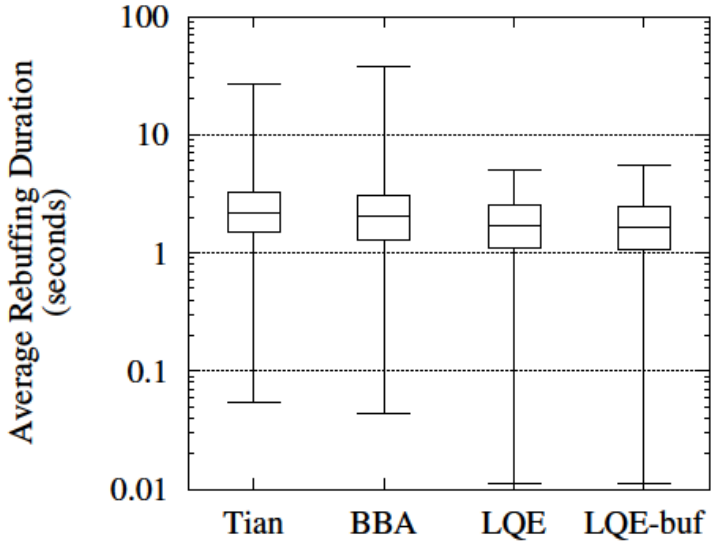

Figure 3. Whisker plot of Average Rebuffering Duration

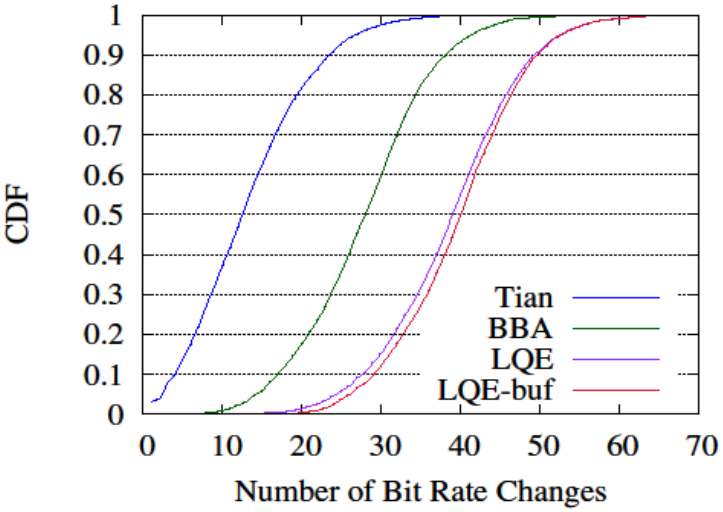

Figure 4. CDF of Number of Bit Rate Changes

Figures 5 and 6 exhibit example traces showing how LQE and LQE-buf select bit rates according to available throughput in a selected scenario. In the figure, we see that LQE-buf chooses higher bit rate than LQE, which is more proper for the network bandwidth. Figures 7 and 8 present the change of buffer status of LQE and LQE-buf in the same scenario. As shown in figure 7 and 8, LQE experiences rebufferings at around time 900 secs while LQE-buf does fewer since it prepared more buffered video at around time 600 secs. We observe that those rebufferings affect the selected bit rates even after 1000 secs: LQE could not increase bit rates properly while LQE-buf aggressively increases bit rates as shown in Figures 5 and 6.

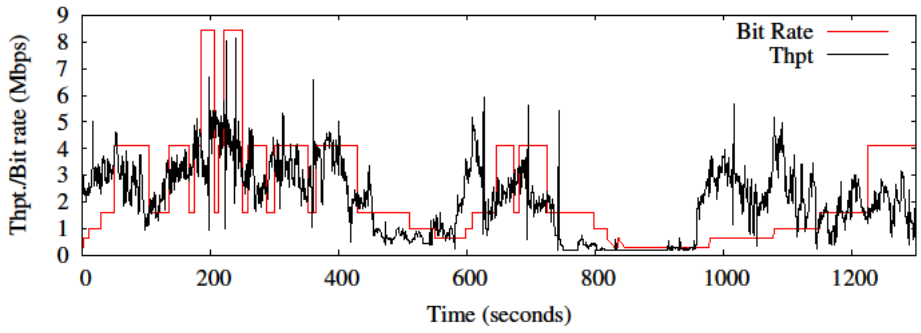

Figure 5. Example Bit Rate Trace of LQE

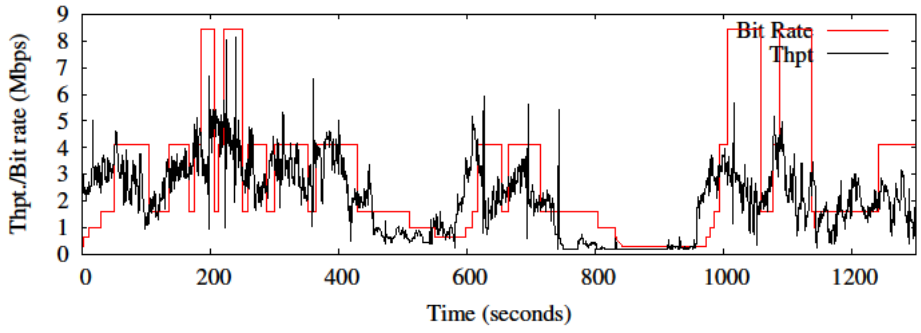

Figure 6. Example Bit Rate Trace of LQE with buffer size optimizaiton

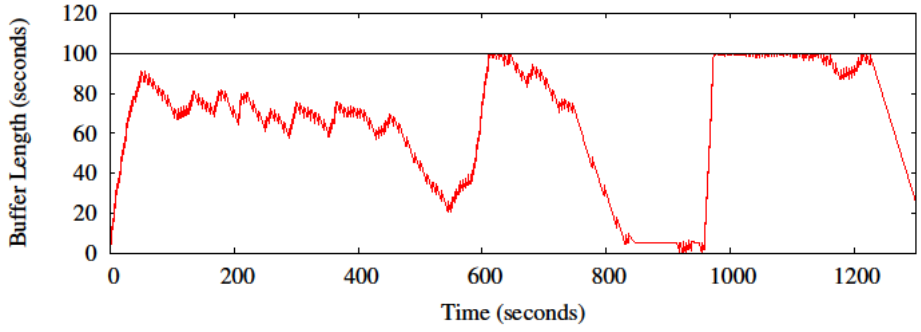

Figure 7. Example Buffer Length Trace of LQE

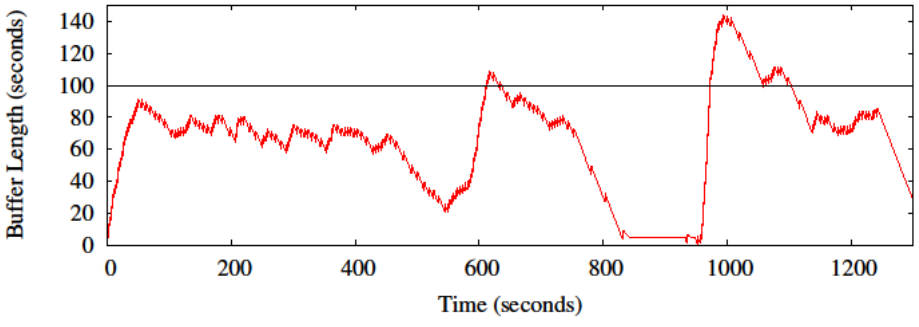

Figure 8. Example Buffer Length Trace of LQE with buffer size optimization

5. Conclusion

The size of the maximum playback buffer size set in the conventional DASH Client player can greatly affect the user experience. If a buffer of too small size is used, a playback stall may occur in a sudden bad network situation. In contrasts, if it is too large, traffic is wasted unnecessarily if the user stops watching in the middle. As a result, adversely affecting the user experiences. This paper proposes a novel maximum buffer size optimization according to available network bandwidth and buffer status. Based on control theory using a discrete time linear quadratic regulator, LQE uses the playback buffer level as a signal, in which the controller targets the buffer level to maintain the reference buffer level. To evaluate the performance of our proposed scheme, we conduct the trace-based simulations in terms of average bit rate, average rebuffering duration, and number of bit rate changes. Our extensive simulation study shows that our proposed buffer size optimization scheme successfully mitigates playback stalls while preserving the similar quality of streaming video compared to existing ABR schemes.

References

[1] Sandvine. The Mobile Internet Phenomena Report, 1H 2020. https://www.sandvine.com/download-report-mobile-internet-phenomena-report-2020-sandvine.

[2] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP – Standards and Design Principles", in Proc. of ACM MMSys, pp. 133–144, Feb 2011, DOI: https://doi.org/10.1145/1943552.1943572

[3] T.Y. Huang, R. Johari, N. McKeown, and M. Trunnell, and M. Watson, "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service", in Proc. of ACM SIGCOMM, pp. 187–198, Aug 2014. DOI: https://doi.org/10.1145/2740070.2626296

[4] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive". IEEE/ACM Transactions on Networking, vol 22, Issue: 1, pp 326–340. Feb 2014. DOI: https://doi.org/10.1145/2413176.2413189

[5] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. Begen, and D. Oran, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale", IEEE Journal on Selected Areas in Communications, vol 32, Issue 4, pp. 719–733, April 2014. DOI: https://doi.org/10.1109/JSAC.2014.140405

[6] G. Tian, and Y. Liu, "Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming", in Proc. of ACM CoNEXT, pp. 109–120, Dec 2012. DOI: https://doi.org/10.1145/2413176.2413190

[7] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP", in Proc. of ACM SIGCOMM, pp. 325–338, Aug 2015. DOI: https://doi.org/10.1145/2785956.2787486

[8] T. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard". in Proc. of ACM IMC, pp. 225–238, Nov 2012, DOI: https://doi.org/10.1145/2398776.2398800

[9] A. Rao, Y. Lim, C. Barakat, A. Legout, D. Towsley, and W. Dabbous, "Network Characteristics of Video Streaming Traffic", in Proc. of ACM CoNEXT, 2011, pp. 25:1–25:12. Dec 2012, DOI: https://doi.org/10.1145/2079296.2079321

[10] G. Tian, and Y. Liu, "On Adaptive HTTP Streaming to Mobile Devices", in Proc. Packet Video Workshop (PV) 20th International, pp. 1–8, Dec 2013. DOI: https://doi.org/10.1109/PV.2013.6691450

[11] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP DASH".in Proc. Packet VideoWorkshop (PV), 2013 20th International, pp. 1–8. Dec 2013. DOI: https://doi.org/10.1109/PV.2013.6691442

[12] G.C. Goodwin, S.F. Graebe, and M.E. Salgado Control System Design; Prentice Hall PTR: Upper Saddle River, NJ, USA, 2000.

[13] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications", in Proc. of ACM MMSys, pp. 114–118, Feb 2013. DOI: https://doi.org/10.1145/2483977.2483991

[14] Y. Kang, et al. "Bit Rate Adaptation using Linear Quadratic Optimization for Mobile Video Streaming", TR-2020.