HIGH EFFICIENCY LIVE VIDEO STREAMING WITH FRAME DROPPING

Yunlong Li**, Shanshe Wang***, Xinfeng Zhang*‡*, Chao Zhou*†*, Siwei Ma*$^{\star *}$

* National Engineering Laboratory for Video Technology ,Peking University, Beijing, China
† Beijing Kuaishou Technology Co., Ltd, Beijing, China
‡ University of Chinese Academic of Sciences, Beijing, China

ABSTRACT

HTTP based video streaming is widely adopted in video service and its adaptive bitrate algorithms have attracted a lot of studies in recent years. However, most of the algorithms are designed for on-demand video service and are not suitable for live video streaming service, which is sensitive to end-to-end latency. In this paper, we propose a live video streaming framework based on HTTP/2 which enables frame dropping for low latency environment. Firstly, we formulate live video streaming model and QoE model considering frame dropping. Then, an optimization problem is formulated aiming at high quality of video streaming. Furthermore, to solve this problem, we propose a live video streaming adaptation algorithm with frame dropping based on Model Predictive Control (MPC). Finally, extensive experiments are conducted to evaluate the proposed method over realistic traces with general Adaptive Bitrate Algorithms (ABR). Compared with the optimal solution, the proposed method achieves comparable performance with only 8.06% quality loss.

Index Terms— Live video streaming, low latency, MPC, frame dropping, DASH.

1. INTRODUCTION

In recent years, HTTP based adaptive video streaming (HAS) has been widely studied and deployed in video delivery industry [1]. HAS has been proved to be efficient in on-demand (VoD) video service [2, 3]. Furthermore, it has also been extended to live video streaming [4], VR/AR video streaming [5], multi-view video streaming [6] and so on. Live video streaming can be roughly divided into three categories: scalable video coding based streaming [7], HTTP/1.1 chunk based streaming [8] and HTTP/2 push based streaming [4, 9, 10]. By splitting video segments into smaller chunks, the above methods can further reduce video streaming latency. However, high efficiency adaptive bitrate algorithms for live video service have not been well designed in particular.

In practice, there are three major challenges for live video streaming. First, although splitting video segments into smaller chunks can decrease encoding and transmission delay, the resource limitations make the servers can only provide limited bitrate levels for transcoding and storage. Thus the ABR algorithms cannot always find a proper bitrate to transmit for more precise rate control. Second, live video streaming need to provide high quality of experience (QoE) under low latency constraint. The service can provide better QoE for VoD when the player buffer is enough to compensate network fluctuation, but it will significantly increase the latency for live video service. Third, the bandwidth prediction and bitrate decision are more challenging, because network estimation and bitrate decision must be made after an interval when the video chunk is ready.

Considering the low latency constraint and precision control on bitrate, we propose a live video streaming framework with frame dropping strategy based on HTTP/2 to improve QoE for live video service. Firstly, we formulate live video streaming model and QoE model respectively taking frame dropping into account. Then, an optimization problem is formulated aiming at high QoE service for live video. To solve this problem, we propose a live video streaming adaptation algorithm with frame dropping based on Model Predictive Control (MPC) to maximize QoE under the low latency constraint. Finally, extensive experiments are conducted to evaluate the performance of the proposed method over realistic traces. Our contributions can be concluded as follows:

- A HTTP/2-based Live video streaming framework with frame dropping is proposed based on our live video streaming model and relative QoE model.
- A practical algorithm with frame dropping strategy, bandwidth prediction strategy and adaptation strategy is proposed based on MPC.
- Extensive experimental results and analyses are provided to verify the efficiency of our method over realistic traces.

*This work was supported by Key-Area Research and Development Program of Guangdong Province (2019B010133001), National Key Research and Development Project (2019YFF0302703) and High-performance Computing Platform of Peking University, which are gratefully acknowledged.

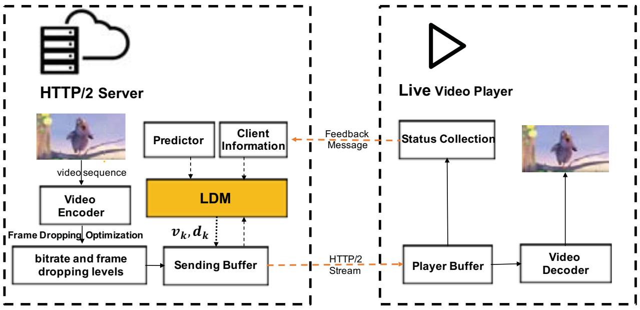


Fig. 1. Live video streaming framework with frame dropping

2. RELATED WORK

In general, the related works on HTTP-based live video streaming can be classified into two categories, which are introduced in the following subsections respectively.

2.1. HTTP/1.1 Chunk Based Streaming

HTTP chunked transfer encoding supported by HTTP/1.1 [11] enables a web server to transmit partial responses in chunks before the complete response is ready. HTTP/1.1 chunk-based approaches split a HAS segment into smaller chunks, encode and transmit chunks before the entire segment is published [8]. These approaches manage to reduce the latency from 1-2 segment durations to 1-2 chunk durations but no efficient chunk scheduling algorithms were proposed to avoid re-buffering.

2.2. HTTP/2 Push Based Streaming

HTTP/2 standard was published as an IETF RFC in February 2015 [12]. HTTP/2 provides two prominent features called *server push* and *stream termination*. Several works leverage the *server push* feature to enable a shorter end-to-end delay and utilize the *stream termination* to discard several frames before video re-buffering occurred [4, 9, 13, 14, 15]. In [9], HTTP/2 is firstly used in live video streaming and a K-Push strategy is proposed to eliminate the request explosion problem introduced by super-short segments.

3. THE FRAMEWORK AND FORMULATION OF THE PROPOSED LIVE VIDEO STREAMING

In this paper, we propose a novel live video streaming framework to improve the QoE of live video service, which is illustrated in Fig.1. Our framework includes a live HAS server and a live HAS client. For the live HAS server, video sequences uploaded from the video captures will be transcoded and packaged into several bitrate presentations and frame dropping levels. Since live video uploading has been optimized based on the method in [16] and our work only optimizes the downlink part performance from the server to the player. Live HAS client is responsible for video receiving, playing and sending player side status back for adaptation decision.

3.1. Live Video Streaming Model

In this section, we describe the evolution model of live video streaming with frame dropping.

Segment Generation: Generally, each video segment is generated in time interval of T for live video streaming service. However, the k^{th} video segment may not be ready at time $k * T$ for encoding and CDN arriving delays. We add a random function $\varepsilon(k)$ to describe the real time generation of live video streaming,

$$t_k^{generation} = k * T + \varepsilon(k), k > 0. \tag{1}$$

Client Buffer Status: The download time of the k^{th} segment is formulated as follows,

$$t_k = \frac{s(v_k, d_k)}{c_k}, \tag{2}$$

where $s(v_k, d_k)$ is the segment size at bitrate level v_k and frame dropping level d_k. c_k is network capacity. Thus, the buffer transfer model can be written as follows,

$$b_k = max(0, (b_{k-1} - t_k - \varepsilon(k) + T)), \tag{3}$$

where b_k is the buffer size after transmitting the k^{th} segment.

Re-buffering: Re-buffering events will occur when the player buffer is drained, the re-buffering interval during downloading the k^{th} segment is,

$$t_k^{rebuffer} = max(0, (t_k + \varepsilon(k) - b_{k-1})). \tag{4}$$

Latency: The time interval between frame display time and its generation time is the latency. The latency will remain unchanged if there is no re-buffering event. The latency is as follows,

$$l_k = l_{k-1} + t_{k-1}^{rebuffer}. \tag{5}$$

When the latency exceeds the max latency limit l_{max}, we will control client play speed to reduce latency.

3.2. QoE Model

We formulate our QoE model following the general QoE model used by MPC [17] as following,

$$QoE = \sum_{k=1}^{K} q(v_k) - \alpha \sum_{k=1}^{K} T_k - \beta \sum_{k=1}^{K} |q(v_k) - q(v_{k-1})|, \tag{6}$$

for a streaming session with K segments in total. Herein, $q(v_k)$ maps bitrate v_k to user-perceived QoE and T_k is the re-buffering time of the k^{th} segment. α and β are parameters reflecting different QoE preference. In this work, we choose VMAF [18] score as the QoE metric,

$$q(v_k) = vmaf(v_k). \tag{7}$$

The detailed introduction for VMAF based frame quality and frame dropping strategy are described in section 4.1. In this work, VMAF score and other QoE metrics are scaled by a certain ratio.

4. THE PROPOSED LIVE VIDEO STREAMING ADAPTATION ALGORITHM

The detailed introduction for the proposed live video streaming adaptation algorithm with frame dropping based on MPC is described, and we name the proposed method as LDM, which includes frame dropping optimization, future information prediction and model predictive control.

4.1. Frame Dropping Optimization

In the proposed LDM, we firstly formulate the frame selection problem as a frame dropping optimization problem based on VMAF score as follows,

$$\begin{aligned} \max \quad & \sum_{i=1}^{I} vmaf(i), \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^{I} s(i) * f_i \leq B_{max} \\ \text{if } f_i = 0, \text{ then } f_j = 0, j \in P_i \end{cases}, \end{aligned} \tag{8}$$

where $s(i)$ is the i^{th} frame size and f_i is frame dropping variable. When f_i is equal to 0, it means that the i^{th} frame is dropped, vice versa. P_i is a frame set in which frames depend on the i^{th} frame and cannot be decoded when it is dropped. Thus, frames in P_i are also dropped when the i^{th} frame is dropped. $vmaf(i)$ is the VMAF score of the i^{th} frame. B_{max} is a predefined upper-bound of segment size. We duplicate the previous frame when current frame is dropped. Thus, $vmaf(i)$ can be calculated as follows,

$$vmaf(i) = \sum_{D=0}^{i-1} (vmaf^D(i) * f_{i-D} * \prod_{d=1}^{D} (1 - f_{i-D+d})), \tag{9}$$

where D is the interval from the nearest dropped frame and $vmaf^D(i)$ is the VMAF score when the i^{th} frame is replaced by the nearest previous frame.

We solve the above frame dropping optimization problem using *IBM Cplex* [19] as a mixed integer optimization problem. We find that $vmaf^D(i) \approx 0$ when $D \geq 2$. Thus, we suppose $vmaf^D(i) = 0, D \geq 2$ to reduce the calculation complexity of the frame dropping optimization problem. By the above assumption, we can get a sub-optimal frame dropping optimization solution in real time. In this work, we prepare several frame dropping levels for selection instead of making frame dropping decision in each step which is a much more complicated problem.

Algorithm 1 LDM Workflow

Initialize
if Start-up stage **then**
  low latency start-up
end if
repeat
  Network Prediction $c_k = \widehat{c}_k, k \in [k+1, M]$
  Segment Information Prediction $info_k = \widehat{info}_k, k \in [k+2, M]$
  $v_{k+1}, d_{k+1} = f_{LDM}(s(k), \widehat{c}_k, \widehat{info}_k)$
  Download chunk $k+1$ with bitrate v_{k+1} and frame dropping level d_{k+1}
until k==K

4.2. Future Information Prediction

4.2.1. Network Capacity Prediction

Network capacity prediction has been fully studied in previous works and can be solved by many algorithms such as Moving Average, Harmonic Mean, LSTM and so on. However, for live video streaming, video segment may not be prepared when client request arrives at the server. For network capacity prediction of next segment, we utilize the harmonic mean by considering t_k^{gap}, which is the gap time until the video segment is ready

$$c_{k+1} = a(t_k^{gap}) * hm(c_{k-m}, ..., c_k), \tag{10}$$

where $a(t_k^{gap})$ is a conserved factor in (0, 1] and calculated as,

$$a(t_k^{gap}) = 1 - \log(1 + t_k^{gap}), t_k^{gap} \in [0, +\infty]. \tag{11}$$

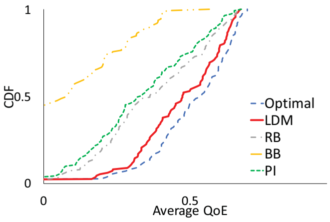
4.2.2. Segment Information Prediction

Video segment information for live video streaming cannot be obtained except that the segment is buffered at the server. In this work, we estimate the future segment size through Moving Average (MA) and the future segment VMAF score through Harmonic Mean (HM).

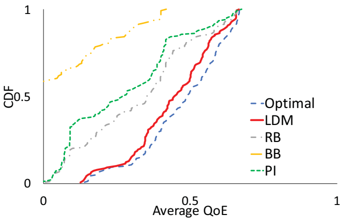
4.3. Model Predictive Control Algorithm

We propose LDM algorithm to make bitrate selection and frame dropping decision. LDM firstly makes network capacity prediction and segment information prediction (if segment is not buffered) for the next M steps and then selects a proper bitrate level and frame dropping level according to the following optimization problem,

$$\begin{aligned} \max \quad & \sum_{i=k}^{k+M} QoE_i \\ \text{s.t.} \quad & S_{i+1} = f(S_i, v_i, d_i, \hat{c}_i, \widehat{info}_i), i \in [k, k+M-1], \end{aligned} \tag{12}$$



(a) HSDPA



(b) FCC

Fig. 2. CDF of QoE Performance

where $S_{i+1} = f(S_i, r_i, d_i, \hat{c}_i, \widehat{info}_i)$ is the live streaming model, $\hat{c}_i$ is the estimated network capacity of the i^{th} segment and $\widehat{info}_i$ is segment information which can be estimated if the i^{th} segment is not ready at the server. v_i and d_i are the bitrate and frame dropping selections for the i^{th} segment. Above all, LDM workflow is shown in Algorithm 1, $f_{LDM}(s(k), \hat{c}_i, \widehat{info}_i)$ is our LDM algorithm, which decides v_{k+1} and d_{k+1} of next segment $k+1$.

5. EXPERIMENTAL RESULTS

In this section, we carry out extensive experiments on real network traces.

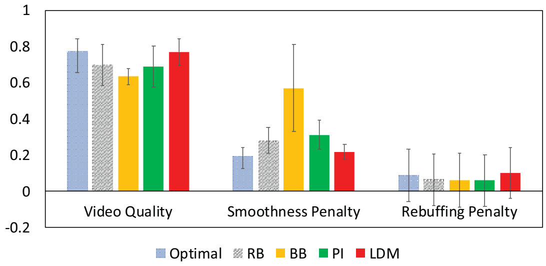
5.1. Setup

Network Traces: We evaluate LDM and several other adaptation algorithms over HSDPA [20] and FCC network trace datasets . HSDPA dataset consists six network conditions including car, train, tram, ferry, metro and bus. FCC dataset is a broadband dataset and relative smooth.

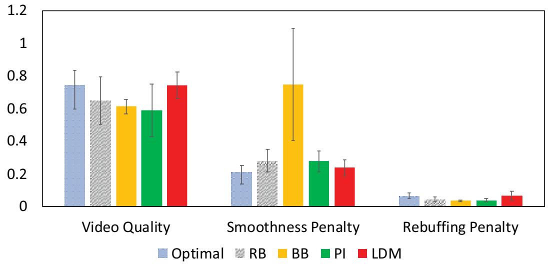
Video Traces: We use "TearsOfSteel" video from [21], where a section with 100s duration is utilized and split into 100 segments with 1s duration. The video is encoded by H.264/AVC codec in the following bitrate levels: 200kbps, 800kbps, 1500kbps, 2400kbps.

Adaptation Algorithms: We evaluate the following algorithms,

- LDM: selects bitrate and makes frame dropping decisions through future information prediction and control algorithm.
- LDM^{optimal} (Optimal for short): future information is assumed to be known and thus optimal decisions can be made.
- BB: selects bitrate and frame dropping levels through buffer occupancy.
- RB: selects bitrate and frame dropping levels through throughput prediction using harmonic mean of past five segments.
- PI: selects bitrate through bandwidth prediction regulated by the difference between the actual buffer and a target buffer length.



(a) HSDPA



(b) FCC

Fig. 3. Comparing LDM with others by analyzing their performance on the individual components in QoE

We evaluate LDM performance in this section. In general, LDM shows 8.77% and 7.49% degradation in QoE comparing with the optimal solution for HSDPA and FCC network traces and outperforms RB, BB and PI methods greatly as shown in Figure 2. BB performance is worst because buffer is too small in low latency live video streaming.

In Figure 3, we split QoE performance in details. Compared with the optimal solution, LDM mainly loses some re-buffering performance and achieves almost the same performance in video quality and smoothness. Comparing performance between HSDPA and FCC, lower re-buffering penalty is achieved in FCC because FCC network dataset is relative smooth.

6. CONCLUSION

In this paper, we proposed a live video streaming architecture with frame dropping based on HTTP/2. We formulated a live streaming model and a QoE model based on VMAF, respectively. Finally, we formulated an optimization problem for live video streaming with frame dropping to improve the QoE of live video service. We also proposed an ABR algorithm denoted as LDM to solve the optimization problem and made extensive evaluation for its efficiency. Over a broad set of network conditions, we find that LDM keeps losses within 8.06% compared with the optimal solution and obviously outperformed the other popular solutions.

7. REFERENCES

[1] "Cisco visual networking index: Forecast and trends, 2017–2022," 2018.

[2] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 2017, pp. 197–210, ACM.

[3] Chao Zhou, Chia-Wen Lin, and Zongming Guo, "mDASH: A Markov decision-based rate adaptation approach for dynamic HTTP streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 738–751, 2016.

[4] Mariem Ben Yahia, Yannick Le Louedec, Gwendal Simon, Loutfi Nuaymi, and Xavier Corbillon, "HTTP/2-based Frame Discarding for Low-Latency Adaptive Video Streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1, pp. 18:1–18:23, Feb. 2019.

[5] Lan Xie, Zhimin Xu, Yixuan Ban, Xinggong Zhang, and Zongming Guo, "360probdash: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming," in *Proceedings of the ACM on Multimedia Conference*, 2017, pp. 315–323.

[6] Xue Zhang, Laura Toni, Pascal Frossard, Yao Zhao, and Chunyu Lin, "Adaptive Streaming in Interactive Multiview Video Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[7] Y. Sanchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Louédec, "Efficient HTTP-based streaming using Scalable Video Coding," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 329–342, Apr. 2012.

[8] V. Swaminathan and S. Wei, "Low latency live video streaming using HTTP chunked encoding," in *IEEE 13th International Workshop on Multimedia Signal Processing*, 2011, pp. 1–6.

[9] Sheng Wei and Viswanathan Swaminathan, "Low Latency Live Video Streaming over HTTP 2.0," in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, Singapore, Singapore, 2013, pp. 37–42.

[10] Mengbai Xiao, Viswanathan Swaminathan, Sheng Wei, and Songqing Chen, "Dash2m: Exploring http/2 for internet streaming to mobile devices," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 22–31.

[11] "Hypertext transfer protocol – http/1.1," June 1999.

[12] "Hypertext transfer protocol version 2," 2015.

[13] H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham, and T. C. Thang, "HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423–2427, Sept. 2018.

[14] Rafael Huysegems, Jeroen van der Hooft, Tom Bostoen, Patrice Rondao Alface, Stefano Petrangeli, Tim Wauters, and Filip De Turck, "HTTP/2-Based Methods to Improve the Live Experience of Adaptive Streaming," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 541–550.

[15] Jeroen van der Hooft, Cedric De Boom, Stefano Petrangeli, Tim Wauters, and Filip De Turck, "An HTTP/2 push-based framework for low-latency adaptive streaming through user profiling," in *NOMS IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–5.

[16] Devdeep Ray, Jack Kosaian, K. V. Rashmi, and Srinivasan Seshan, "Vantage: Optimizing Video Upload for Time-shifted Viewing of Social Live Streams," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 380–393.

[17] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *ACM SIGCOMM Computer Communication Review*, 2015, vol. 45, pp. 325–338.

[18] "VMAF," `https://medium.com/netflix-techblog/vmaf-the-journey-continues`.

[19] "IBM Cplex," `https://www.ibm.com/cn-zh/products/ilog-cplex-optimization-studio`.

[20] Haakon Riiser, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen, "Commute Path Bandwidth Traces from 3g Networks: Analysis and Applications," in *Proceedings of the 4th ACM Multimedia Systems Conference*, 2013, p. 5.

[21] Stefan Lederer, Christopher Müller, and Christian Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012, pp. 89–94.