BUFFER-BASED CONTROL THEORETIC APPROACH FOR DYNAMICALLY HTTP STREAMING

Zhimin Xu[1,2], *Chao Zhou*[1], *Li Liu*[1], *Xinggong Zhang*[1,3,*], *Zongming Guo*[1,3]

[1]Institute Computer Science & Technology, Peking University, Beijing, P.R. China
[2]Beijing University of Posts & Telecommunications
[3]Cooperative Medianet Innovation Center, Shanghai, China

ABSTRACT

Dynamic adaptive streaming over HTTP (DASH) has recently been widely deployed in the Internet. It, however, does not impose any adaptation logic for selecting the quality of video fragments requested by clients. In this challenge, we have designed a novel rate adaptation scheme for Dynamic HTTP Streaming, by which, low start-up time, continuous and smooth video playback, and high bandwidth utilization are obtained. The algorithm is mainly based on our previous work [1] [2] that a PD controller are adopted to guide the rate adaptation. Moreover, we further improve the performance of start-up delay by fast-start approach and dynamic buffer size adjustment. The numerous experiment results have demonstrated the good performance of our designed rate adaptation compared with the Bitdash [3].

1. INTRODUCTION

In recent years, dynamic adaptive streaming over HTTP (DASH) has been widely adopted for providing uninterrupted video streaming service under various network conditions and heterogeneous devices[4]. Dynamic rate adaptation is one of the most important features of DASH. However, there are still many open issues for client-side rate adaptation[5].

In this challenge, we have designed a novel control-theoretic approach to switch video segments in dynamic adaptive HTTP streaming, by which, low start-up delay, continuous and smooth video playback, and high bandwidth utilization are obtained [1] [2] [6]. First, we have designed a dual-threshold buffer model, by which the effect of short-term bandwidth fluctuation on rate decision is subsided, and smooth video playback is guaranteed. Then, to prevent buffer underflow (playback freeze), a PD controller is designed to make the rate selection decision. At last, to quick start video playback, the minimal buffered video time threshold (the video playback is allowed start only when the buffered video time is no less than this threshold) is dynamically adjusted according to fragment length. Besides, a fast-start technology is also adopted. Compared with Bitdash, our proposed rate adaptation algorithm has obtained much smoother video bit-rate and higher bandwidth utilization. Moreover, continuous video playback is also guaranteed that seldom video playback freezing happens during all the experiments.

It is also noticed that the start-up delay in our proposed method is a little higher than that of Bitdash. It is because Bitdash uses a minimal playback buffer(1 segment), while it is no less than 2 in our method. Minimal buffering is able to reduce delay, however, it also easily leads to playback freezing and bit-rate fluctuation under bandwidth variation. We consider using a dynamic buffer size adjustment and fast-start approach to reduce start-up delay, instead of minimizing the buffer size. Even under this unfairness comparison, our performance of start-up delay is still very close to that of Bitdash.

All in all, extensive experiments are carried under controlled test-bed and real Internet trace. This work is mainly based on our previous works[1] [2]. The novelty and performance can be summarized into three-folds:

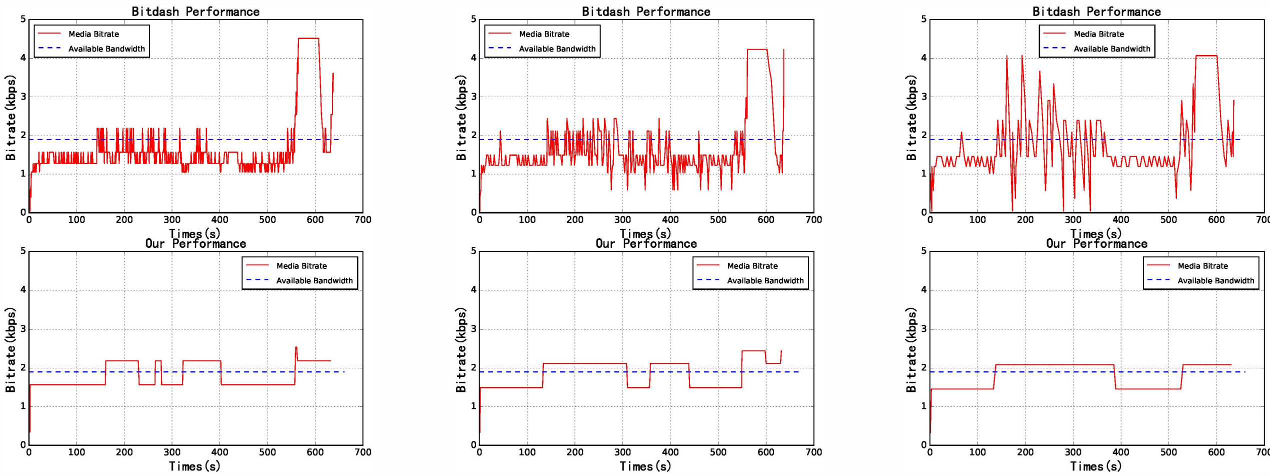
- A dual-threshold-based video buffering model to smooth video bit-rate reduces switching frequency more than 60% compared with Bitdash.
- A rate adaptation scheme with PD controller is designed to prevent video playback freezing, while the bandwidth utilization is improved more than 30%.
- Combined with fast-start technology and our designed dynamic buffering, we decrease start up delay and guarantee smoothness of video bit-rate simultaneously.

2. SYSTEM SETUP

In our implementations, the server with Apache server is used as the media server which hosts media fragments and *.mpd* file. For the client, we modify the open source available MPEG-DASH player *dash.js* [7] and implement our proposed rate adaptation algorithms. All the tested dataset are downloaded from the ICME 2016 Grand Challenge site. The test network consists of four nodes, i.e., Client, Router, Network

*Corresponding author. E-mail:zhangxg@pku.edu.cn

This work was supported by National Natural Science Foundation of China under contract No. 61471009.



(a) Fragment length 1s

(b) Fragment length 2s

(c) Fragment length 4s

Fig. 1. Performance evaluation with available bandwidth unchanged

Emulator, and HTTP Server, where Network Emulator is used to control the available bandwidth between server and client.

The system consists of the following components:

- Bandwidth estimation. A simple but effective exponential averaging method is used to estimate bandwidth from historical download rate.

- Smoothing bit-rate switching. A dual threshold based model is designed to subside the effect of bandwidth variation. More details can refer to our previous works[1] [2].

- Playback freezing avoidance. To guarantee continuous video playback and high bandwidth utilization, we have designed a PD controller. It switches rate by comprehensively considering bandwidth, buffer occupancy and the changing rate as well. More details can refer to our previous works [1] [2].

- Fast start-up. Combined with the fast-start technology and designed dynamic buffer adjustment, we decrease the start up delay and guarantee the smoothness of video playback simultaneous.

Platform	Average Video Bitrate (Kbps)	Switching Frequency (Times)
Bitdash	1119.38	155
Our	1703.66	18

Table 1. Performance statistics under real Internet trace

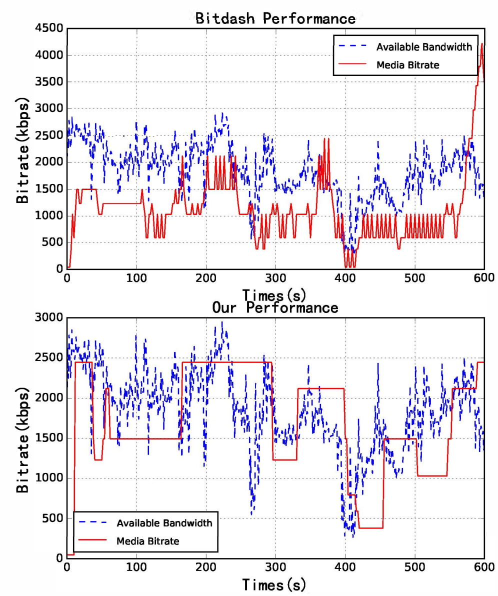
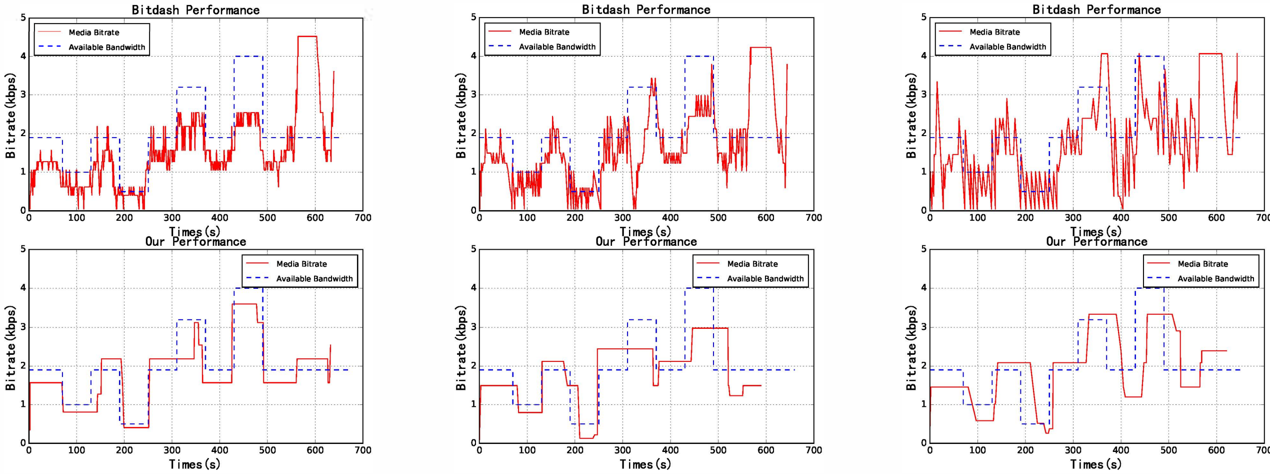


Fig. 4. Performance under real Internet trace

3. EXPERIMENT RESULTS AND ANALYSIS

We conduct extensive experiments and compare it with Bitdash under the various experimental environments. In a controlled network, both regular network conditions, such as fixed and square bandwidth, and irregular conditions by replying Internet real traces, are carried out. To eliminate the side effects of fragment length, we test three different fragment lengths ($1s$, $2s$ and $4s$) under all network conditions.



(a) Fragment length 1s

(b) Fragment length 2s

(c) Fragment length 4s

Fig. 2. Performance evaluation with long term bandwidth variations

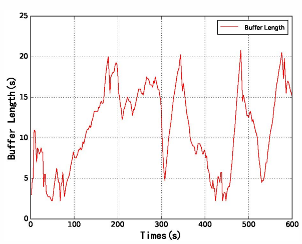


Fig. 5. Buffer evolution of our system under real Internet trace

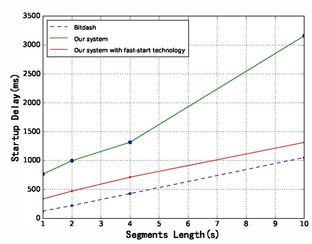


Fig. 6. Startup delay under 500kbps bandwidth

3.1. Experiments with Regular Bandwidth

We first test the performance of Bitdash and our system under the case of fixed bandwidth with 1900kbps. The results are shown in Fig. 1(a) - Fig. 1(c). In the results we notice that the video bit-rate in Bitdah is switched frequently for all the fragment length. Moreover, the gap of the bit-rate between the adjacent fragments can be very big in Bitdash, and it even switches between the highest and lowest video bit-rate. This is annoying to user experience. While in our proposed method, we can find that much more smoother video bit-rate is obtained. And as the fragment length increases, the performance of smoothness performs better.

The impact of long-term bandwidth variations are shown Fig. 2(a) - Fig. 2(c). The results show that both Bitdash and our system are responsiveness to bandwidth variations. But the oscillation frequency and amplitude are still high for Bitdash. Also there are some case in Bidash the rate oscillation results into re-buffering. As the contrast, the video rate in our system is more smoothing and continuous.

At last, we summarize the results of experiments where the available bandwidth goes through positive or negative spikes that last for a few seconds in Fig. 3. Generally, short-term variations are common in practice and we think that a good rate adaptation scheme should be able to compensate for such spikes using its buffered video, without causing short-term rate switching. In Bitdash, similar to the case of fixed bandwidth and long term bandwidth variations, the video bit-rate always switches frequently and the switching amplitude is also very big. While in our system, most of the bandwidth spikes are ignored and smooth video bit-rate is provided, and it mainly thanks to our proposed dual threshold buffer model.

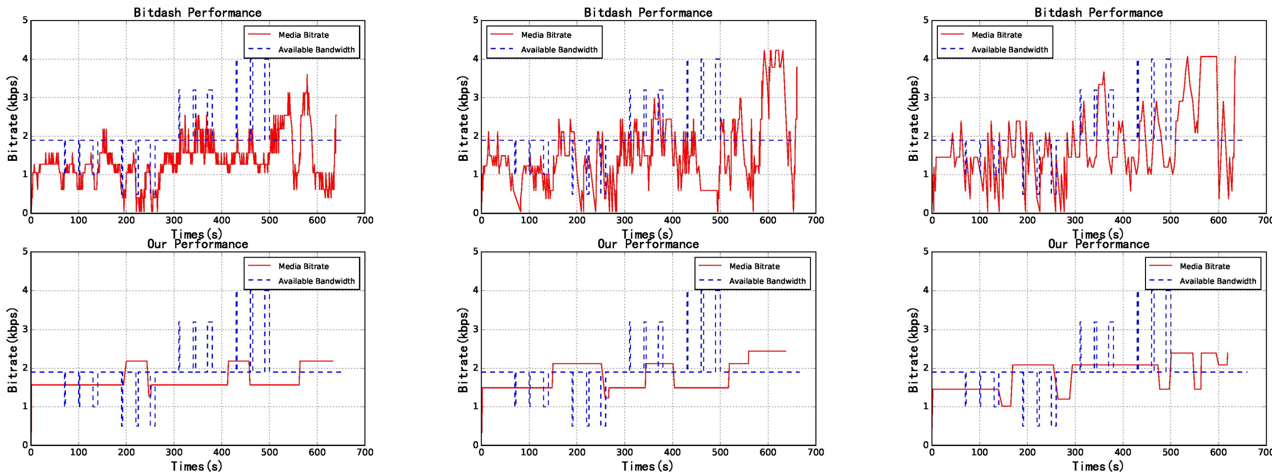
3.2. Experiments with Real Internet Trace

Secondly, we test our system with real Internet bandwidth trace. In all experiments, we do observe long-term shift and short-term fluctuations of bandwidth along the bandwidth trace. The video bit-rate results in Fig. 4 demonstrate that our video bit-rate adaptation algorithms can adapt well to the network conditions, which is consistent with our previous results. Our system is able to compensate for shot-term spikes using its buffered video, without causing short-term rate switchings. While for long-term bandwidth variations, it switches to a suitable video bit-rate without causing buffer overflow or playback interruptions as Fig. 5 shows.

Last, the results shown in Table 1 clearly show that our proposed scheme achieves the highest bandwidth utilization and average video bit-rate. Besides, the video bit-rate switch times is less than of Bitdash, and much smoothing video bit-rate is provided.

3.3. Start Up Delay Performance

The above results shows that compared with Bitdash, our proposed approach can obtain smoother video bit-rate and higher



(a) Fragment length 1s

(b) Fragment length 2s

(c) Fragment length 4s

Fig. 3. Performance evaluation with short bandwidth variations

bandwidth utilization. However, for the start-up delay, Bitdash gets better performance as Fig. 6, this is mainly in Bitdash, only one fragment need to be buffered before beginning to start the playback. While in our proposed scheme, the video fragments needed to be pre-buffered is dynamically determined based on the fragment length and it is generally more than one fragment. However, we think this is sensible that only sacrifice a little performance of start-up delay, we can effectively avoid playback freeze, and it also give us more space for smooth video bit-rate control. Besides, we also adopt the fast-start technology to reduce the start-up delay without affecting the other performance, such as the smooth, the bandwidth utilization, and so on.

4. CONCLUSION

In this work, we have designed a novel rate adaptation scheme for Dynamic HTTP Streaming and implemented it over *dash.js*, by which, low start-up delay, continuous and smooth video playback, and high bandwidth utilization are obtained. Client implementation and several experimental log files are also uploaded as supplement materials.

5. REFERENCES

[1] C. Zhou, X. Zhang, and Z. Guo, “A control-theoretic approach to rate adaptation for dynamic HTTP streaming,” *Visual Communications and Image Processing (VICP), IEEE*, 2012.

[2] Chao Zhou, Chia-Wen Lin, Xinggong Zhang, and Zongming Guo, “A Control-Theoretic Approach to Rate Adaption for DASH Over Multiple Content Distribution Servers,” *IEEE Trans. Circuits. Syst. Video Technol.*, vol. 24, no. 4, pp. 681–694, Apr. 2014.

[3] bitmovin, “bitdash.min.js,” *available online: http://bitmovin.com.*

[4] C. Mller, D. Renzi, S. Lederer, S. Battista, and C. Timmerer, “Using Scalable Video Coding for Dynamic Adaptive Streaming over HTTP in mobile environments,” *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, IEEE*, 2012.

[5] TY Huang, N Handigol, B Heller, N Mckeown, and R Johari, “Confused, timid, and unstable: picking a video streaming rate is hard,” *Acm Conference on Internet Measurement Conference*, 2012.

[6] N. Bouzakaria, C. Concolato, and J. Le Feuvre, “Fast DASH bootstrap,” *Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on*, 2015.

[7] DASH Industry Forum, “dash.js,” *available online: http://github.com/DASH-Industry-Forum/dash.js.*