Optimal Scheduling of QoE-Aware HTTP Adaptive Streaming

Ray-I Chang[§], Yu-Chi Liu[§], Jan-Ming Ho[*†], Yu-Hsien Chu [§], Wei-Chun Chung[*†‡], Chi-Jen Wu[†]
[§]Department of Engineering Science and Ocean Engineering, National Taiwan University, Taiwan
[*]Institute of Information Science, Academia Sinica, Taiwan
[†]Research Center for Information Technology Innovation, Academia Sinica, Taiwan
[‡]Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
rayichang@ntu.edu.tw,r03525057@ntu.edu.tw,d00525003@ntu.edu.tw,{hoho}@iis.sinica.edu.tw,{wcchung, cjwu}@iis.sinica.edu.tw

***Abstract*—Recently HTTP adaptive streaming (HAS) has been leading the trend in the delivery of video content over Internet. The scope of this technology describes that a video is segmented into small intervals and encoded in different qualities for adapting the variance of network bandwidth. This method allows the client adjusting the quality of the requested stream dynamically. To the best of our knowledge, most heuristic algorithms proposed for HAS run a risk of freezing in video playback and thus induce a poor Quality of Experience (QoE). Therefore, how to maintain a good user experience becomes a more challenging task for the service provider. In this paper, we propose a optimal scheduling of QoE-aware HAS method which achieves the following QoE requirements. (1) Avoiding video streaming freezing (i.e. the client buffer underflow). (2) Minimizing the initial delay before the video streaming starts playing. Testing by the video *Aladdin*, experimental results show that our method can find the best QoE service for HAS under various QoE requirements and resource constraints.**

Keywords*—HTTP Adaptive Streaming, Rate adaptation algorithm, Quality of Experience (QoE)

I. Introduction

Mobile data traffic is dominated by video streaming, which delivered over the top (OTT) [1]. Video content providers (e.g. Netflix or YouTube) adopt HTTP Adaptive Streaming (HAS) technology to provide uninterrupted video streaming service for users with the dynamic network conditions and device capabilities. The HAS technology is also standardized in MPEG-DASH [2, 3]. The concept of the MPEG-DASH system is shown in Fig. 1. A video content on a server is encoded into different bitrates, and it is split into several segments of short duration (e.g. tens of seconds). Besides, a media presentation description (MPD) is defined for MPEG-DASH to communicate between an HTTP server and a DASH client.

A lot of research [5]-[10] in MPEG-DASH area tries to find the best adaptation or scheduling strategy to maximize a user's Quality of Experience (QoE). The conventional rate adaptation algorithms only heuristically determine the presentation of media stream by bandwidth-based or buffer-based method [5]. However, most of them exist a potential risk of stalling video playback, result in decreasing the QoE. In this paper, we proposed optimal scheduling of QoE-aware HAS method to avoid video freezing. Firstly, upper-bound algorithm found the upper bound of playback schedule, and calculated the minimum initial delay before a client application starts playing a video. The second is QoE playback schedule algorithm, which decides what segment presentation should be scheduled for download next. Specifically, the following contributions are achieved in this paper:

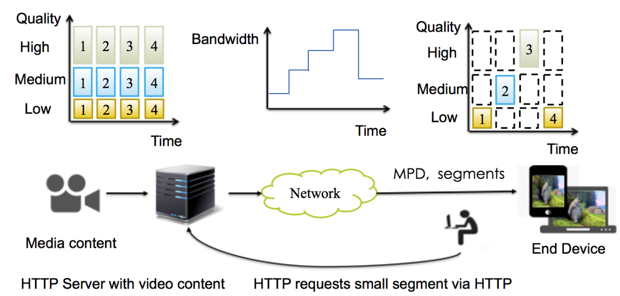

Fig. 1: The general architecture of MPEG-DASH

1) Propose a media stream model and two algorithms, upper-bound schedule algorithm and QoE playback schedule algorithm, to fulfill the QoE criteria as well as avoiding video stalling and minimizing initial delay. To be more precise, we achieve the optimal requirement where the obtained playback schedule will not to be re-buffering in the future.

2) In the simulation, the buffer size and the initial delay are considered at different network bandwidths. The results show that there is existing a best QoE service with the bandwidth 50 Kbps.

The remainder of this paper is organized as follows. Section II revisits related work. Upper-bound schedule algorithm, minimum initial delay/buffer and QoE playback schedule algorithm are defined in Section III. Section IV presents the experimental results for our approach. Finally, Section V concludes the article, and gives future work.

II. Related work

For QoE optimisation of video traffic, Qadir *et al.* [4] categories the approach of literature proposed based on these

978-1-4799-8641-5/15/$31.00 2015 IEEE

functions, including rate adaptation, cross-layer mechanisms, scheduling, content and resource management. Nevertheless, the type of rate adaptation is almost heuristic algorithm to fulfil QoE criteria. Such as, buffer-based method [5, 6, 7, 8] detects the buffer size in the client side, then sets some thresholds to swap video bitrates. Furthermore, bandwidth-based method [9, 10] takes bandwidth into account to make the switch-up or switch-down decision. But, it is difficult to estimate throughput accurately under an uncertain complex network.

III. METHODOLOGY

First of all, the media stream is modeled by the notation and variables that are summarized in Table I. There are two algorithms are introduced. Upper-bound schedule algorithm found the upper-bound of the video playback schedule, and calculated the minimum initial delay and buffer size. Secondly, QoE playback schedule algorithm will schedule the feasible playback scheme based on our methods.

TABLE I: Notation used in our methods

Notation	Definition and description
N	The number of segments divided by a video.
F	The number of frames in the video. In addition, each segment i includes f frames.
R	The number of different bitrates. In addition, $k = 1$ is used to denote the highest bitrate and so on.
$V(S_{ij}, k)$	The video with different bitrates k chopped into multiple segments i at the j-th frame, $1 \leq k \leq R, 1 \leq i \leq N, 1 \leq j \leq F$.
U	The set of the upper-bound we determine.
ΔT	The segment interval.

A media stream is available in k={1,2,...,R} representations and divides into N segments, each segment S_{ij} includes f frames at index j and has a constant duration ΔT. Therefore, the Eq. (1) denotes the video with N segments, F frames and R bitrates.

$$\{V(S_{ij}, k) \colon k = 1, 2, ..., R\} \quad 1 \leq i \leq N, 1 \leq j \leq F \tag{1}$$

A. Our Upper-Bound Schedule Algorithm

To generate a playback schedule without stalling, transmitting data is supposed to be lower than the data size transmitted. Thus, the backward induction is used in our algorithm, and takes the N^{th} segment and the $(N-1)^{th}$ segment in a video for example. As shown in Fig. 2, the first step is to determine the video stream shifting up or shifting down in each segment. If the cumulative consumption in any frame of the segment is higher than the transmitting data, the segment will be shifted down to hit the bandwidth at one point (i.e. all frames in the segment under the bandwidth). Otherwise, shifting up the segment hits the bandwidth at one point (i.e. all frames in the segment over the bandwidth).

After our algorithm shifts the segment, the middle between them would not connect to each other on the same point. Therefore, each segment $N-1$ instead of the N^{th} segment should be checked the $i * f$ frame (i.e. the end point of the segment) whether it is higher than the connection point (i.e. the $(i-1) * f$ frame in the segment $i+1$). If it is higher than the connection point, the segment will be shifted down to match the connection point. Otherwise, not do anything to the segment. At last, the upper-bound in the segment i is determined by selecting the maximum cumulative frame size in all video bitrates R to form the upper-bound set U, and fitted to the following equation:

$$U = max\{V(S_{ij}, k)\} \quad {}_{1 \leq i \leq N, 1 \leq k \leq R, (i-1)*f \leq j \leq (i*f)} \tag{2}$$

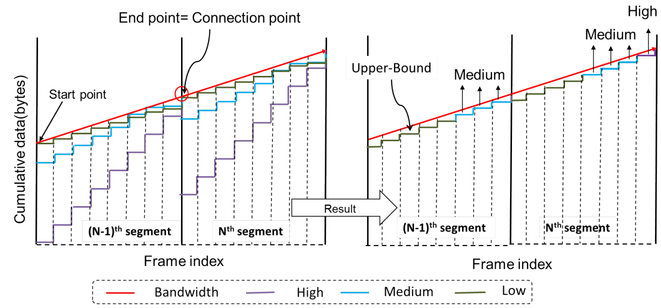

Fig. 2: An illustration of the upper-bound process

B. Calculating Minimum Initial Delay and Buffer

As shown in Fig. 3, taking two segments for example. After upper-bound schedule algorithm is completed, we extend the bandwidth (the red line) to hit the highest position of bitrate k (the green line; low bitrate) at a new horizontal line. In our example, they hits at the point A. Given the slope of the constant bandwidth is M and the difference between the bandwidth and the highest position of the resolution k is H_k. Hence, the minimum initial delay can be calculated by $\frac{H_k}{M}$. Besides, The virtual buffer constraint line (the dotted red line) can be drawn, which is parallel with the bandwidth, and is tangent to the lowest position of the upper-bound at a point. The minimum buffer size is the difference between the bandwidth and the lowest point of upper-bound.

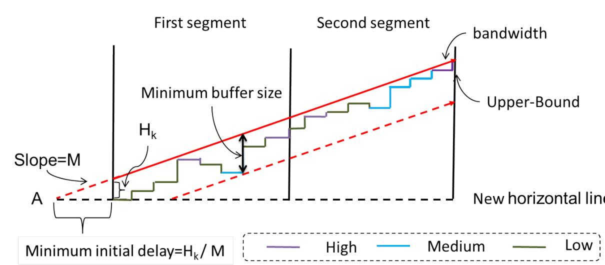

Fig. 3: An illustration of the minimum initial delay and buffer size

C. Our QoE Playback Schedule Algorithm

A schedule is feasible if and only if transmitted data is lower than the bandwidth during playback. More specifically, each segment we schedule is not over the bandwidth at each frame. Thus, the algorithm iterates through bitrates to find a segment with appropriate bitrate for playback, and every segment we determine will form the playback schedule.

IV. Experimental Results

In this paper, a media trace from the website [11] is examined to evaluate the effecticeness of the proposed algorithm. The video *Aladdin* is selected as a demo-stream with 3 different bitrates (high: 435 Kbps, medium: 155 Kbps and low: 100 Kbps). The statistics of it is listed in Table II, which includes frame numbers, the frame rates (number of frames played per second), the frame sizes (maximum, and average), and the group-of-picture sizes (maximum, and average). The cumulative playback data of the first 143 frames of *Aladdin* is shown in Fig. 4.

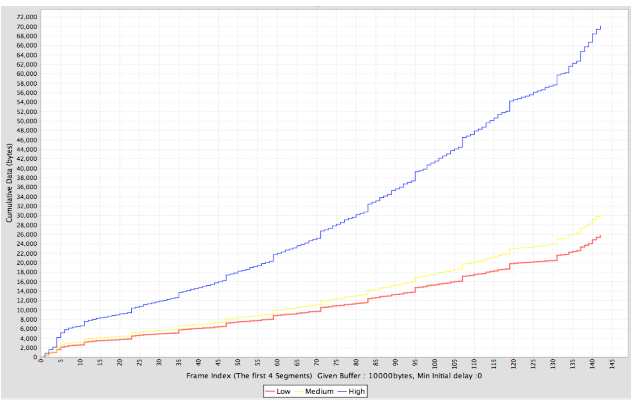

Fig. 4: The cumulative playback data of the first 4 segments (i.e. 143 frames) with 3 difference bitrates

The simulation first executes the algorithm our proposed. Assumed that the limited of available network bandwidth is 50 Kbps, and a segment is three times the number of a GoP's (i.e. a GoP = 12 frames; a segment = 36 frames). As shown in Fig. 5, the process and result for finding the upper-bound are illustrated, and take the first 4 segments for example. The algorithm shifts all segments under the network bandwidth, and the highest position of bitrates is selected as a upper-bound. Then, the minimum initial delay and buffer size are calculated, which are 4 frames (i.e. 4*25 ms = 1 sec) and 2342 bytes at the 135-th frame (see Fig. 6) respectively. At last, Fig. 7 shows the playback schedule with our QoE playback schedule algorithm (after delay 1 sec), which selects the best bitrate if the network bandwidth can transmit it.

Furthermore, the different constant bandwidths (30 Kbps, 40 Kbps, 50 Kbps, 60 Kbps, 70 Kbps, 80 Kbps, 90 Kbps) are examined to observe the change of the initial delay and the buffer size. The intolerable limit for initial delay [12] is set to the range as [500 ms, 100 ms] (the dotted gray line) to analyze the proper network bandwidth and buffer size. Our experiments show that too much or too little bandwidth would increase the required client buffer (see Fig. 8). As the result shown in the Fig. 8 and Fig. 9, our algorithm can calculate and decide the feasible and suitable bandwidth 50 Kbps which has the best QoE in serving the video. Owing to it is the point of minimum client buffer and the initial delay in the intolerable range compared with other bandwidths.

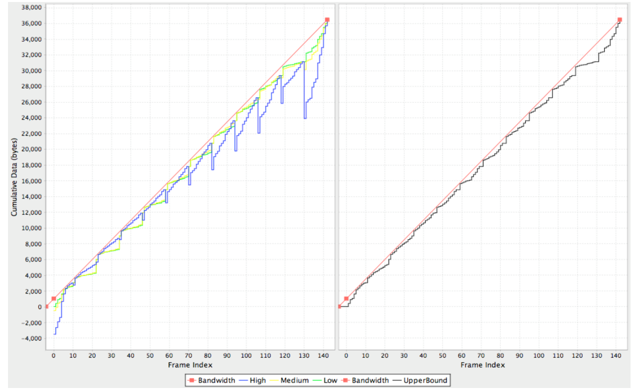

Fig. 5: The process for finding the upper-bound with 3 different bitrates

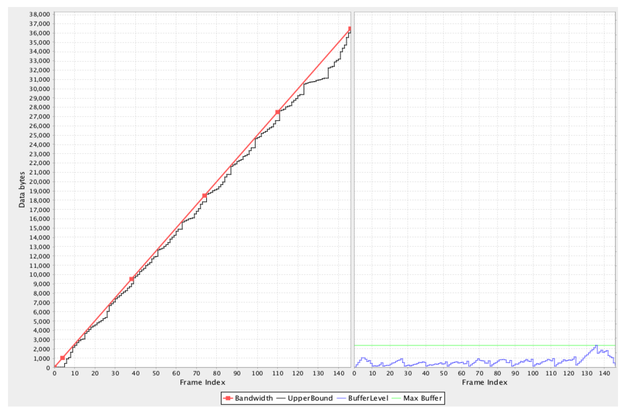

Fig. 6: The minimum buffer size

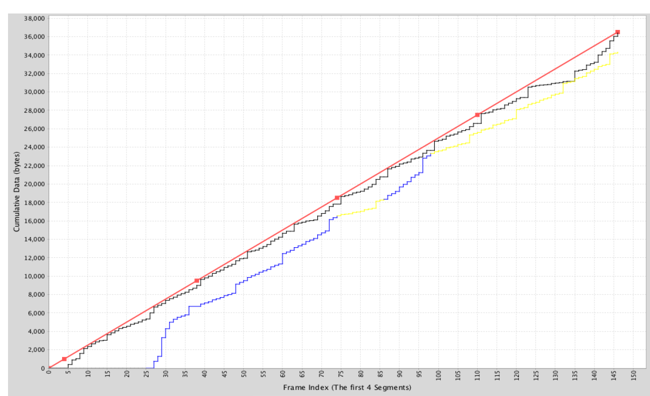

Fig. 7: The playback schedule with upper-bound constraint

TABLE II: Statistics a real video *Aladdin* with 3 different bitrates of the media streams used in our experiments

Stream name	Frame numbers	Frames rate (fps)	Quality	Bitrate (Kbps)	Frame size (KB)		GoP size (KB)	
					MAX	AVG	MAX	AVG
			Low	100	6241	503	29471	6039
Aladdin	89998	25	Medium	155	6735	774	48025	9297
			High	435	15385	2176	116034	26122

fps: frame-per-second; GoP: group-of-picture.

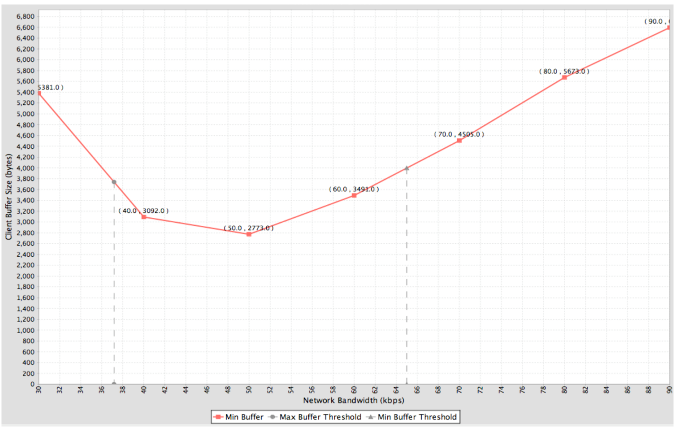

Fig. 8: The minimum buffer size under different bandwidths by testing a real video *Aladdin* with 3 different bitrates in QoE

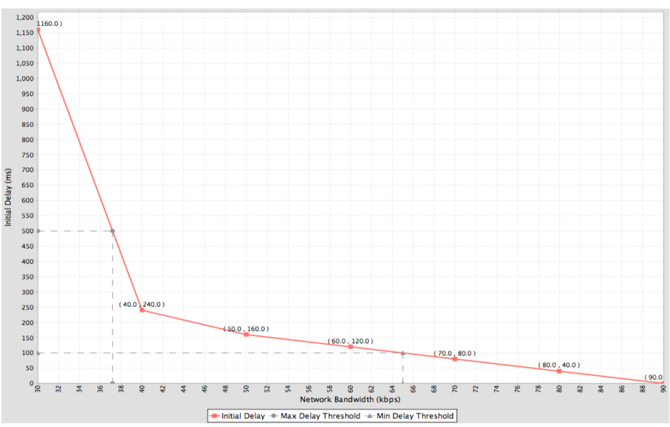

Fig. 9: The minimum initial delay under different bandwidths by testing a real video *Aladdin* with 3 different bitrates in QoE

V. Conclusion and Future work

We have proposed two optimal scheduling of QoE-aware HTTP adaptive streaming algorithms. The one is upper-bound schedule algorithm, which determines the bound that makes the playback schedule not be stagnation, and calculates the minimum waiting time for the client to play the video. The other is QoE playback schedule algorithm, which is the greedy schedule in transmitting the video data, and is characterized by avoiding frequent interruptions. At last, our experiments show that the best condition of QoE and the proper bandwidth for the sample using our proposed algorithm are found. Interesting future work is involved in a variable bandwidth wireless environment to determine a minimal resource allocation for each user.

Acknowledgment

The authors wish to thank the anonymous reviewers for their help, suggestions, and valuable comments. This research is partially supported by Ministry of Science and Technology of Taiwan under grant NSC 102-2221-E-001-013-MY3, 102-2410-H-002-170-MY3 and 104-2627-E-002 -005, and System Management and Content Retrieval Technologies for Supporting Cloud-based Digital Archive Systems and Services of Academia Sinica.

References

[1] I. Cisco, " Cisco visual networking index: Forecast and methodology, 2011—2016," CISCO White paper, 2011-2016.

[2] I. Sodogar, "The MPEG-DASH Standard for Multimedia Streaming over the Internet", IEEE Multimedia, vol. 18, no. 4, pp. 62-67, Oct.-Dec. 2011.

[3] T. Stockhammer, " Dynamic adaptive streaming over HTTP: standards and design principles," Proceedings of the second annual ACM conference on Multimedia systems, 2011, pp. 133-144.

[4] Q. M. Qadir, A. Kist, and Z. Zhang, "Mechanisms for QoE optimisation of Video Traffic: A review paper," Australasian Journal of Information, Communication Technology and Applications, vol. 1, pp. 1-18, 2015.

[5] C. Yuming, Y. Xiaoquan, W. Jia, and S. Li, " A QoE friendly rate adaptation method for DASH," Proceedings of 2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2014, pp. 1-6.

[6] L. De Cicco, S. Mascolo, and V. Palmisano, " Feedback control for adaptive live video streaming," Proceedings of the second annual ACM conference on Multimedia systems, 2011, pp. 145-156.

[7] C. Mller, S. Lederer, and C. Timmerer, " An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," Proceedings of the 4th Workshop on Mobile Video, 2012, pp. 37-42.

[8] C. Zhou, X. Zhang, L. Huo, and Z. Guo, "A Control-Theoretic Approach to Rate Adaptation for Dynamic HTTP Streaming," Proceedings of Visual Communications and Image Processing (VCIP), 2012, pp. 1-6.

[9] C. Liu, I. Bouazizi, and M. Gabbouj, " Rate adaptation for adaptive HTTP streaming," Proceedings of the second annual ACM conference on Multimedia systems, 2011, pp. 169-174.

[10] J.-M. Jeong and J.-D. Kim, " Effective bandwidth measurement for Dynamic Adaptive Streaming over HTTP," Proceedings of 2015 International Conference on Information Networking, 2015, pp. 375-378.

[11] M. Reisslein and F. H. Fitzek, "Trace files." http://www-tkn.ee.tu-berlin.de/research/trace/ltvt.html, 2015. [Online; accessed 02-February-2015].

[12] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler, "Waiting times inquality of experience for web based services," Proceedings of 2012 Fourth International Workshop on Quality of MultimediaExperience (QoMEX), 2012, pp. 86-96.