Buffer State is Enough: Simplifying the Design of QoE-Aware HTTP Adaptive Video Streaming

Weiwei Huang, Yipeng Zhou, Xueyan Xie, Di Wu, *Senior Member, IEEE*, Min Chen, *Senior Member, IEEE*, and Edith Ngai

***Abstract*—Recently, the prevalence of mobile devices together with the outburst of user-generated contents has fueled the tremendous growth of the Internet traffic taken by video streaming. To improve user-perceived quality-of-experience (QoE), dynamic adaptive streaming via HTTP (DASH) has been widely adopted by practical systems to make streaming smooth under limited bandwidth. However, previous DASH approaches mostly performed complicated rate adaptation based on bandwidth estimation, which has been proven to be unreliable over HTTP. In this paper, we simplify the design by only exploiting client-side buffer state information and propose a pure buffer-based DASH scheme to optimize user QoE. Our approach can not only get rid of the drawback caused by inaccurate bandwidth estimation, but also incur very limited overhead. We explicitly define an integrated user QoE model, which takes playback freezing, bitrate switch, and video quality into account, and then formulate the problem into a non-linear stochastic optimal control problem. Next, we utilize control theory to design a dynamic buffer-based controller for DASH, which determines video bitrate of each chunk to be requested and stabilize the buffer level in the meanwhile. Extensive experiments have been conducted to validate the advantages of our approach, and the results show that our approach can achieve the best performance compared with other alternative approaches.**

***Index Terms*—DASH, control theory, buffer state, rate-adaption.**

Manuscript received August 16, 2017; revised November 4, 2017; accepted December 15, 2017. Date of publication January 17, 2018; date of current version June 5, 2018. This work was supported in part by the National Key R&D Program of China under Grant 2016YFB0201900, in part by the National Natural Science Foundation of China under Grant 61572538, in part by the Fundamental Research Funds for the Central Universities under Grant 17LGJC23, in part by the STINT Initiation Grant for International Collaboration under Grant IB2017-6978, and in part by the Australian Research Council under DE180100950. *(Corresponding author: Di Wu.)*

W. Huang, X. Xie, and D. Wu are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China, and also with the Guangdong Province Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou 510006, China (e-mail: huangww26@mail2.sysu.edu.cn; xiexy9@mail2.sysu.edu.cn; wudi27@mail.sysu.edu.cn).

Y. Zhou is with the Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia (e-mail: yipeng.job@gmail.com).

M. Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: minchen@ieee.org)

E. Ngai is with the Department of Information Technology, Uppsala University, SE-751 05 Uppsala, Sweden (e-mail: edith.ngai@it.uu.se).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2018.2789580

I. Introduction

ACCORDING to the Cisco report [1], it is predicted that the global IP traffic will increase threefold between 2013 and 2018, and 80-90 percent of the total traffic will be taken by video streaming by 2018. Such a tremendous growth is credited to the rapid growth of user-generated contents (UGC) through crowdsourcing-based applications (such as YouTube and Twitch), the prevalence of mobile devices (e.g., smart phones, tablets) and business growth of traditional Internet content providers. For instance, Netflix contributes more than 20% of the U.S. Internet traffic [2]. On the other hand, due to expensive bandwidth, varying network conditions and diverse user preferences, it has become a challenging task for a service provider to serve users massive contents with satisfied QoE and sustainable cost, e.g., healthcare cyber-physical systems [3] and vehicular ad hoc network [4], etc.

To tackle the surging video streaming traffic, 3GPP (The 3rd Generation Partnership Projection) and MPEG (The Moving Picture Experts Group) proposed Dynamic Adaptive Streaming over HTTP (DASH), which is a new MPEG standard that defines the interaction and cooperation between consumers and content providers in order to maintain smooth video streaming with limited available bandwidth. Its mechanism is to split the video into chunks with equal playback duration. Each chunk is further encoded into multiple versions with different bitrates. A client-side controller is installed on each user, which selects chunks with appropriate bitrates for transmission according to the latest system state. The underlying principle is that streaming smoothness is more important than streaming bitrate for user experience. Thus, during busy hours with less bandwidth, videos should be delivered at a lower bitrate to avoid streaming interruption and vice verse. Due to its ability to adapt streaming bitrate with network condition, DASH has been widely used by real systems [5], such as Apple HTTP Live Streaming, Microsoft Live Smooth Streaming, Adobe HTTP Dynamic Streaming, QuavStreams Adaptive Streaming over HTTP and so on.

To achieve adaptive video streaming, most of previous DASH solutions performed rate adaptation based on accurate estimation of network bandwidth. Unfortunately, Jiang *et al.* [6] and Tian and Liu [7] pointed out that bandwidth estimation is inherently unreliable and inaccurate over the HTTP layer, which may lead to undesirable bitrate variations and low-quality video streaming. Thus, to avoid the above limitation, researchers start to consider the design of new DASH approaches, in which bandwidth estimation is not

required. The pioneering work [8] showed that it is feasible to perform rate adaptation simply based on the local buffer state. However, their approach is static and has not taken complicated user-perceived QoE into account. The development of pure buffer-based DASH is still in its infant age.

In this paper, we propose a QoE-aware buffer-based DASH scheme to optimize user QoE, which is simple yet effective. It can adaptively choose chunks with appropriate bitrates for downloading based on the latest information of buffer state. To better quantify user QoE, we first propose an integrated user QoE model that considers playback freezing, bitrate switch and video quality jointly. Then we formulate the problem into a non-linear stochastic optimal control problem and formally prove that a static optimal controller doesn't exist. To solve the problem, we exploit control theory to design a novel dynamic buffer-based controller, which can stabilize video buffer level and thus guarantee user QoE. Through extensive simulations, we show that our scheme can not only effectively reduce the frequency of playback interruptions, but also dramatically improve other video quality metrics. Overall, our main contributions in this paper can be summarized as follows:

- We propose an integrated user QoE model for HTTP adaptive video streaming, which considers major factors that affect user QoE, such as playback freezing, bitrate switch, video quality and so on.
- We formulate the QoE-aware HTTP adaptive streaming problem into a non-linear stochastic optimal control problem, and prove that there doesn't exist a static optimal controller to achieve the optimality.
- We further exploit control theory to design a dynamic buffer-based controller for DASH, which determines the bitrate for each chunk to be requested in an online manner. The proposed approach can be easily implemented at the client side.
- We conduct extensive simulations to evaluate our proposed solution. Experimental results indicate that the buffer level can be stabilized effectively and video bitrate switches occur rarely. The freezing events can be almost completely avoided. Our solution outperforms other alternative solutions in most scenarios.

The rest of the paper is organized as follows. Section II reviews related works. In Section III, system model and problem definition are presented. Section IV elaborates the design of our dynamic controller. Experimental results are presented in Section V. Finally, we summarize this paper and suggest some future works in Section VI.

II. Related Work

The field of Adaptive Bitrate (ABR) streaming applications and optimization has been well investigated in the last two decades (e.g., [5] and [9]–[15]). Romero [9] implemented DASH into Google Android systems to provide adaptive video streaming. De Vleeschauwer *et al.* [5] studied the optimization of overall utility in a base station and proposed an algorithm to control the bitrate of each data flow. Joseph *et al.* [10] presented an algorithm to optimize the overall utilities for a group of users under the constraints of bitrate and user dynamics. Ramamurthi and Oyman [11] proposed a framework for wireless resource allocation to reduce the rebuffering probability. Ji *et al.* [16] proposed the first hybrid spatial-temporal metric in adaptive video streaming so as to provide heterogeneous devices with better reliable video. Liang *et al.* [12] proposed an online approach to prefetch and cache data for DASH. Chang and Chiao [13] constructed an streaming scheme for multicast in DASH. Chen *et al.* [14] used the User Adaptive Video (UVA) technique to reduce the bandwidth cost for DASH systems. Krishnapa *et al.* [15] conducted an experimental evaluation of DASH with YouTube video traces and analyzed the advantages and disadvantages of DASH.

For video bitrate selection, the proposed techniques can be classified into two main categories: (1) bandwidth-estimation-based; (2) buffer-based. The bandwidth-estimation-based approach is widely used in [2], [6], and [7]. Most works regard the bandwidth estimation as a reference signal for making choices on selection or adjusting the control systems. De Cicco and Mascolo [2] designed an adaptive video streaming control system consisting of two controllers. The system controls both the client side and the server side. Bandwidth estimation is used for adjustment in that system. Jiang *et al.* [6] proposed a bandwidth-estimation-based algorithm called FESTIVE to achieve three goals of bitrate selection: fairness, efficiency and stability. Tian and Liu [7] presented a buffer controller for bitrate selection based on bandwidth estimation. Yin *et al.* [17] used model predictive control (MPC) theory to solve the bitrate selection problem, which makes optimal bitrate decisions by predicting the expected bandwidth for the next few chunks.

The buffer-based approach is investigated in [8], [18], and [19]. De Cicco *et al.* [18] proposed a client-side controller called ELASTIC. Zhou *et al.* [19] designed a control system for the multi-server DASH system. However, these control systems are multi-input and somewhat complicated. Huang *et al.* [8] proposed a buffer-based system in the client-side for bitrate selection of DASH. The system has only one input: the buffer state. However, it did not consider the problem of improving user QoE and reducing the cost. In addition, the above solutions [18], [19] also need to predict the available bandwidth. There are also some papers [20]–[22] that investigated the QoE measurement and its relation with adaptive bitrate streaming.

Previous studies have shown that control-theoretic approaches are effective for dynamic adaptive streaming [17]–[20]. The advantage to use a controller for bitrate adaption is a cleaner design, which can be not only experimentally tested but also mathematically analyzed [23]. Huang *et al.* [24] developed an effective proportional controller to stabilize the received video quality as well as the bottleneck link queue, for both homogeneous and heterogeneous video systems. The work in [17] analyzed the client-side bitrate adaptation logic based on a control theoretic approach and proposed a novel *model predictive control (MPC)* algorithm that can optimally combine throughput and buffer state information to outperform the state-of-art approaches. In [25], a control-theoretic mechanism is designed to analyze the

TABLE I
MAJOR SYMBOLS USED IN THIS PAPER

Symbol	Meaning
$b(t)$	buffer level at time t
t_k	time instant starting to download the k-th chunk
b_k	buffer level at time t_k
r_k	video bitrate of the k-th chunk
$c(t)$	available bandwidth at time t
c_k	average bandwidth when downloading the k-th chunk
Δt_k	waiting time when the buffer is full
τ	the length of each video chunk
t_f^k	freezing time when downloading the k-th chunk
Q^k	user QoE of the k-th chunk

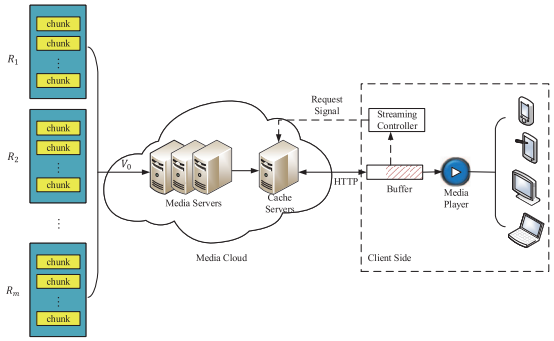


Fig. 1. Architecture of a typical DASH system.

TCP congestion and it is shown that TCP itself is an unstable integral control system. The proportional-integral-derivative (PID) controller is the most common control algorithm used in industry and provides the simplest and most effective solution for many real-world control problems [26]. Zhou *et al.* [19] proposed a novel proportional-derivative (PD) controller to adapt video bitrate with high responsiveness and stability. The study in [27] analyzed the video-based pricing and transmission schemes with consideration of heterogeneous services requirements, and proposed a polymatroid-based marginal control to optimize the video streaming with differentiation QoE provision. These successful efforts clearly demonstrate the strength of the control-theoretic approaches.

III. System Model and Problem Definition

In this section, we first describe a typical DASH system architecture, and then develop its system model, which includes network model, buffer model, video request model and user QoE model. To facilitate our presentation, the notations are summarized in Table I.

A. System Architecture

Based on the 3GPP-DASH specification [28], a typical architecture of a DASH system can be illustrated in Fig 1. Generally, a DASH system is comprised of three components: (1) media servers, which generate video chunks encoded in different bitrates; (2) cache servers (or streaming servers), which deliver video chunks over HTTP; (3) clients, which receive video chunks using various end devices.

In terms of media servers, their responsibility is to convert video chunks into different versions with different bitrates and store encoded chunks. The playback duration of each chunk is almost the same. Video chunks will be delivered by cache servers to clients. In the client side, there is a streaming controller installed on each user device, which determines the bitrate of the next chunk to be requested.

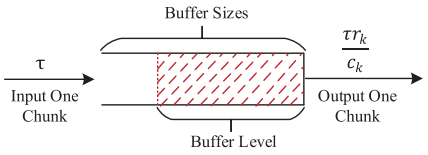


Fig. 2. Playback buffer model.

B. Network Model

In the network model, we model the dynamic process of network bandwidth. Let $c(t)$ be network throughput at time t, and c_k be the average download speed of the k-th video chunk. Assume that all the chunks are downloaded sequentially according to the playback sequence. We adopt a stochastic model to represent network conditions as $c(t)$ could change significantly with time and environment in the real world, especially in the environment with mobile devices. A stochastic model is appropriate in this situation. More specifically, the network capacity $c(t)$ at time t follows a specific distribution, and distributions at different moments are independent and identically distributed (i.i.d). The distribution of network bandwidth can be changed according to different environments (e.g., wired/wireless networks).

C. Buffer Model

As network condition is unpredictable, the arrival of chunks is also a stochastic process. Because chunk size could be different for DASH, we focus on the change of playable chunks cached in the buffer, defined as buffer level. The playback buffer can be modeled as a FIFO queue, as commonly adopted in the previous literatures to analyze HTTP video streaming [8], [18], [29]. Fig. 2 illustrates a simple buffer model.

In Fig. 2, the player plays video content chunk-by-chunk from the buffer. Let r_k be the bitrate of the k-th chunk and c_k be the average bandwidth during downloading the k-th chunk. Assume that each chunk can be played for τ seconds. The player drains $\frac{\tau r_k}{c_k}$ seconds of video content when it finishes downloading the k-th chunk. Note that, an incomplete video chunk will not be played. We further define $b(t)$ as the buffer level at time t and b_k as the buffer level at the beginning of the download process of video chunk k. The relationship between $b(t)$ and b_k can be described as:

$$b_k = b(t_k), \tag{1}$$

where t_k is the time starting to download the k-th video chunk.

Note that once the playback buffer is full, the download will be suspended to avoid overflowing the playback buffer [29]. Let Δt_k denote the time that a user needs to wait before he

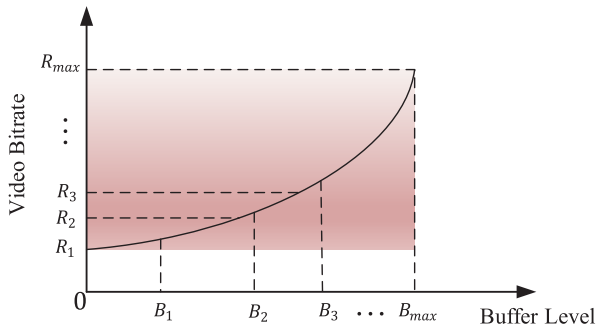


Fig. 3. Example of mapping function.

can request for the next chunk. Then, Δt_k can be derived as follows:

$$\Delta t_k = \max\left\{b_k + \tau - \frac{\tau r_k}{c_k} - B_{max}, 0\right\}, \tag{2}$$

where τ is the length of each video chunk and B_{max} is the bound of the buffer level. With such a mechanism, we can guarantee that the buffer will never be overflowed. In reality, the buffer bound is sufficiently large such that the chance is very rare for b_k to reach B_{max}.

The evolution of the buffer level can be derived as follows:

$$b_{k+1} = \max\left\{b_k + \tau - \frac{\tau r_k}{c_k} - \Delta t_k, \tau\right\}. \tag{3}$$

In the above equation, the term $\tau - \frac{\tau r_k}{c_k}$ represents the incremental amount of the buffer level. If a freezing event occurs, b_{k+1} will be τ. Otherwise, if the buffer is full, the user will wait Δt_k for the next chunk.

D. Video Request Model

In the buffer-based DASH system, the bitrate selection is determined by a mapping function, which chooses the bitrate of the next chunk based on the current buffer state. The client requests the chunk with the chosen bitrate from the remote servers.

Formally, let $\Psi(\cdot)$ denote the mapping function, then we have

$$R_k = \Psi(B_k), \tag{4}$$

where B_k is the buffer level and R_k is the bitrate for the next requested chunk. Note that the feasible bitrate of video chunks can only be selected from the complete bitrate set $\mathbb{R} = \{R_1, R_2, \ldots, R_{max}\}$, and $B_k \in [0, B_{max}]$. It is obvious that there are an infinite number of valid mapping functions.

For better understanding, we use an example to explain how the mechanism works. In Fig. 3, the mapping function is drawn in the solid line. The dash area is the feasible region of the mapping function. R_1 is the lowest bitrate, while R_{max} is the highest. Since the video bitrate is discrete, the mapping function is also discrete. For example, when a buffer level falls into an area, say B_1 and B_2, the client will request a video chunk with the corresponding bitrate (e.g., R_2 in this case).

E. User QoE Model

In the integrated user QoE model, we consider three major factors directly related to user QoE, namely, playback interruption, bitrate switch and video quality. These three major factors reflect the users' QoE from different aspects and thus are necessary to be considered [6], [30]–[32]. We use a QoE score to represent the value of a chunk for user QoE, which includes three sub-scores: freezing score, bitrate switch score and video bitrate score.

The QoE score of the k-th video chunk downloading is defined as:

$$Q^k = Q_v^k + \eta Q_f^k + \lambda Q_s^k, \tag{5}$$

where η and λ are positive weight coefficients. These scores are specified as follows.

1) Freezing Score: When the buffer is drained out, a freezing event will occur and thus deteriorate the user QoE. In [30], the acceptability probability of a video is proposed with logistic regression. The probability that a user does not accept the video due to rebuffered events is modeled as follows:

$$p_{na} = \frac{\exp(-\alpha + \beta T_f)}{1 + \exp(-\alpha + \beta T_f)}, \tag{6}$$

where α and β are constants, T_f is the total freezing time, and p_{na} is in range (0, 1].

In the process of playing the (k-1)-th video chunk, the freezing time t_f^k can be estimated from:

$$t_f^k = \max\left\{\frac{\tau r_k}{c_k} - b_k, 0\right\}. \tag{7}$$

It means that if the buffer was drained out during the (k-1)-th period, the freezing time will be $\frac{\tau r_k}{c_k} - b_k$. Otherwise the freezing time will be 0.

Based on the above model, the QoE score during the downloading of the *k-th* video chunk due to freezing events can be further defined as follows:

$$Q_f^k = -\frac{\exp\left(-\alpha + \beta t_f^k\right)}{1 + \exp\left(-\alpha + \beta t_f^k\right)}. \tag{8}$$

It clearly shows that a longer rebuffered time results in lower QoE.

2) Bitrate Switch Score: From the user perspective, frequent video bitrate switch is also undesirable. Switching from high-quality to low-quality is annoying and less tolerant by users. Such degradation of QoE can be reflected in the degradation of video quality. Based on [6], we define the QoE score of bitrate switching between two adjacent video chunks as follows:

$$Q_s^k = \frac{\mu |r_k - r_{k-1}|}{r_k}, \tag{9}$$

where r_k is the bitrate of the k-th video chunk and μ is a negative scaling factor.

3) Video Bitrate Score: Generally, a higher video bitrate leads to a higher QoE score. As in the previous work, we assume that the QoE function is concave and follows the diminishing return law. Typically, the QoE function is modeled in the logarithmic form [31], [32]. Based on these previous works, we specify the QoE score in terms of video bitrate as follows:

$$Q_v^k = \ln r_k, \tag{10}$$

where r_k is the video bitrate of *k-th* video chunk.

F. Problem Formulation

In this work, our objective is to look for the optimal controller that can maximize user QoE. Since the chunk arrival process is stochastic, so we use the expected value of QoE score. Our problem can be formally defined as:

$$\begin{aligned} &\max_{\Psi} \ \sum_{k=0}^{N-1} \left\{ \mathbb{E}(Q^k) \right\}, \\ &\text{s.t.} \ \ (3)\ (4), \end{aligned} \tag{11}$$

where N is the total number of video chunks that a user requests.

The QoE maximization problem can be translated into an optimal stochastic closed-loop controller design problem. We use the optimal stochastic control theory [33] to analyze this problem and obtain the following result:

Lemma 1: The optimal static mapping function that can optimize the user QoE score does not exist.

The detailed proof is in the Appendix. The result points out the complication of our problem, which means there is no solution for the optimal problem (11) so that we can optimize the QoE score easily. As the optimal static controller does not exist, we choose to design a dynamic controller that can provide a decent QoE performance.

IV. Dynamic QoE-Aware Buffer-Based Controller for DASH

In this section, we exploit control theory to design a dynamic QoE-aware buffer-based controller for DASH systems. Intuitively, a proportional controller calculates the difference between the current buffer level and the target buffer level to avoid draining out the buffer. A derivative controller monitoring the change rate of buffer level is used to reduce bitrate switches. An integration controller estimating the cumulative difference between buffer level and target buffer level can maximize the video bitrate such that the requested bitrate oscillates around the long term average bitrate. The above three controllers can work jointly to make the bitrate adjustment decision. Compared with predictive methods, the control-theoretic approach avoids the need of bandwidth prediction. Our controller makes decisions simply based on the buffer states at the client side.

A. Negative Feedback Control Model

A negative feedback model is suitable to describe a stable system. In DASH systems, a natural idea is that the video bitrate has a positive relationship with the buffer level. Therefore, we propose a proportional controller to model the original control system. It means that, if the buffer level is too high (or too low), we need to request chunks with a higher (or lower) video bitrate to ensure the stability of the buffer.

From the controller point of view, there is an essential conflict between smoothing video bitrate and stabilizing buffer level, due to the ineluctable network bandwidth fluctuations. Nevertheless, from the end user point of view, video bitrate fluctuations are much more perceivable than buffer level oscillations [7]. Using a buffer-based approach to maintain a stable buffer level has a positive relationship with maintaining a stable video bitrate. Moreover, by maintaining the buffer level stability, freezing events can also be avoided. In this case, user QoE can be guaranteed. A threshold b_f is used as the stable buffer level to relieve the buffer level oscillations on video bitrate adaption. When the buffer level deviates from b_f, b_f is used as the operating point to select appropriate video bitrates to avoid buffer level oscillations.

The original control system uses the boundary value b_f to select the video bitrates as follows:

$$r_k = K_{p_1}(b_k - b_f) + C, \tag{12}$$

where K_{p_1} and C are two parameters.

Theorem 1: If K_{p_1} and C satisfy:

$$\begin{cases} 0 < K_{p_1} \le \frac{c_k}{\tau}, \\ 0 < C \le c_k, \end{cases} \tag{13}$$

for all k, then the buffer will not be depleted, which means that the freezing events will be avoided.

Proof: Assume that the conditions (13) hold, then we have

$$\begin{aligned} &\frac{c_k}{\tau}\big((b_k - b_f)\big) + c_k \ge K_{p_1}\big(b_k - b_f\big) + C \\ \Rightarrow\ &\frac{c_k}{\tau}\big((b_k - b_f)\big) + c_k \ge r_k \\ \Rightarrow\ &b_k - b_f + \tau - \frac{\tau r_k}{c_k} \ge 0 \\ \Rightarrow\ &b_k + \tau - \frac{\tau r_k}{c_k} \ge b_f \end{aligned}$$

If the buffer is not overflow, then according to (3), we have $b_{k+1} = b_k + \tau - \frac{\tau r_k}{c_k} \ge b_f > 0$. Thus the buffer will not be drained out. ■

Since it is difficult to obtain the accurate changing rate of the video buffered time, we use the fluid approximation, which evenly distributes the added video time τ over the download interval for the whole video chunk [19]. Then we have:

$$b'(t) = \frac{db(t)}{dt} \approx \frac{b_k - b_{k-1}}{t_k - t_{k-1}} = \frac{c_k}{r_k} - 1. \tag{14}$$

Thus, the control system model can be described by Eq. (12) and (14).

The block diagram of the control model is shown in Fig. 4, where $Q(\cdot)$ denotes the quantization operation because the available video bitrate is discrete. To facilitate the system analysis and design, we propose to linearize the model described in Fig. 4 around the operating point of b_f satisfying $b'(t) = 0$.

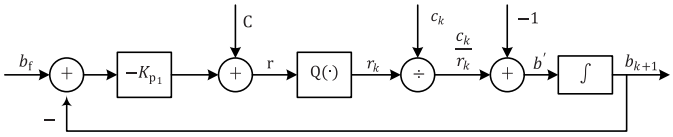


Fig. 4. Block diagram of the original control model described by Eq. (12) and (14).

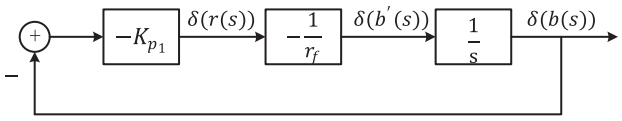


Fig. 5. Block diagram of the linearized control model described by Eq. (16).

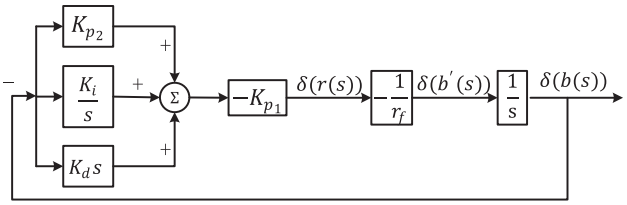


Fig. 6. Block diagram of the PID control system.

Let b_f be the corresponding video bitrate and c_f be the corresponding average bandwidth, then we have

$$b'(t) = 0 \Rightarrow r_f = c_f. \tag{15}$$

The following linearized state and output equations are obtained for the feedback system:

$$\begin{cases} \delta(r_k) = K_{p_1}\delta(b(t)), \\ \delta(b'(t)) = -\frac{1}{r_f}\delta(r_k), \end{cases} \tag{16}$$

where $\delta(r_k) = r_k - r_f$, $\delta(b(t)) = b(t) - b_f$ and $\delta(b'(t)) = b'(t) - b'_f$. After applying the Laplace transform to the differential equations, we derive the linearized system block as shown in Fig. 5. The transfer function of the linearized control system is:

$$H_p(s) = \frac{K_{p_1}}{r_f s + K_{p_1}}. \tag{17}$$

Theorem 2: The linearized control system is stable if $K_{p_1} > 0$.

Proof: The pole of the first-order system is $s_p = -\frac{K_{p_1}}{r_f}$. It is clear that $r_f > 0$, so the pole $s_p < 0$ is located in the left phase plane. Thus the system is stable. ■

B. PID Controller for the Original System

We redesign the PID controller to achieve dynamic control of buffer level. Due to its stable performance in a variety of operating conditions and easy operability, the PID controller is widely used in the industrial control systems and a variety of other applications requiring continuously modulated control. Traditionally, PID controller consists of three parts, including a proportional controller, an integral controller and a derivative controller. PID controller calculates an error value as the difference between a measured process variable and a desired set point. The PID controller algorithm includes three separate parameters: the proportional factor K_p, the integral factor K_i and the derivative factor K_d. The proportional controller calculates the current error message e and generates a control signal $K_p e$. The integral controller accumulates the error message $\int_0^t e\,dt$ and generates a control signal $K_i \int_0^t e\,dt$. The derivative controller is used to adjust the control vector based on the variation trend of the process variable: $\frac{de}{dt}$, and the adjustment will be $K_d \frac{de}{dt}$.

Compared with P controller (a special case of the PID controller in which the integral and derivative of the errors are not used.), PD controller and PI controller, PID controller performs better since it has all the advantages of the three controllers. A pure P controller would cause oscillations and may not converge. In order to improve performance, we design a novel buffer-based PID controller for the original control system.

The transfer function of a classic PID controller is:

$$G(s) = K_{p_2} + \frac{K_i}{s} + K_d s. \tag{18}$$

Applying the PID controller to the original control system, we have the transfer function of the new control system as follows:

$$\begin{aligned} H(s) &= \frac{H_p(s)G(s)}{1 + H_p(s)G(s)} \\ &= \frac{K_{p_1}K_d s^2 + K_{p_1}K_{p_2}s + K_{p_1}K_i}{(K_{p_1}K_d + r_f)s^2 + K_{p_1}(K_{p_2}+1)s + K_{p_1}K_i}. \end{aligned} \tag{19}$$

The block diagram of the PID control system is shown in Fig. 6.

The PID controller uses the difference between the current buffer level and operating point b_f, the changing rate of the buffer level, and the cumulative difference to compute the adjustment of video bitrate. In the time domain, the adjustment of video bitrate can be obtained in the following way:

$$\begin{aligned} \delta(r_k) &= K_{p_1}K_{p_2}(b_k - b_f) \\ &\quad + K_{p_1}K_d\frac{b_k - b_{k-1}}{t_k - t_k - 1} + K_{p_1}K_i\int_0^{t_k}(b(t) - b_f)dt. \end{aligned} \tag{20}$$

Then, the video bitrate of the k-th chunk will be:

$$r_k = Q(\delta(r_k) + r_f). \tag{21}$$

Now, we turn to analyze the parameters K_{p_1}, K_{p_2}, K_d and K_i to stabilize the PID control system described in Fig. 6.

Theorem 3: The control system described in Fig. 6 is stabilized by the PID controller, if the parameters satisfy the following conditions:

$$\begin{cases} (K_{p_2}+1)(K_{p_1}K_d + r_f) > 0 \\ K_i(K_{p_1}K_d + r_f) > 0 \\ K_{p_1} > 0. \end{cases} \tag{22}$$

Proof: The poles of the second-order control system are the roots of the following quadratic equation:

$$(K_{p_1}K_d + r_f)s^2 + K_{p_1}(K_{p_2}+1)s + K_{p_1}K_i = 0.$$

The discriminant is:

$$\Delta = K_{p_1}\Big[K_{p_1}(K_{p_2}+1)^2 - 4K_i(K_{p_1}K_d + r_f)\Big].$$

Algorithm 1 Buffer-Based PID Controller

```
initialize b_0, b_f, r_0, K_p1, K_p2, K_d, K_i
let t ← 0, b(0) ← b_0, r ← r_0
for i ← 1 to N do
    download i − 1 chunk with bitrate r_{i−1};
    update buffer level b_i according to Eq. (3);
    compute δ(r_i) according to Eq. (20);
    r_i = Q(δ(r_i) + r_{i−1});
end for
```

If $\Delta < 0$, then there are two conjugate complex roots $s_{1,2} = -\frac{K_{p_1}(K_{p_2}+1)}{2(K_{p_1}K_d+r_f)} \pm i\frac{\sqrt{-\Delta}}{2(K_{p_1}K_d+r_f)}$. Since $(K_{p_2}+1)(K_{p_1}K_d+r_f) > 0$ and $K_{p_1} > 0$, so $-\frac{K_{p_1}(K_{p_2}+1)}{2(K_{p_1}K_d+r_f)} < 0$. Thus the roots are located in the left phase plane.

If $\Delta = 0$, then there is exactly one root $s = -\frac{K_{p_1}(K_{p_2}+1)}{2(K_{p_1}K_d+r_f)}$. Since $(K_{p_2}+1)(K_{p_1}K_d+r_f) > 0$ and $K_{p_1} > 0$, so $s < 0$ is located in the left phase plane.

If $\Delta > 0$, then there are two distinct real roots s_1 and s_2. According to the conditions (22), we have $s_1 + s_2 = -\frac{K_{p_1}(K_{p_2}+1)}{(K_{p_1}K_d+r_f)} < 0$ and $s_1 s_2 = \frac{K_{p_1}K_i}{K_{p_1}K_d+r_f} > 0$, so $s_1 < 0$ and $s_2 < 0$. Thus the roots are also located in the left phase plane.

In summary, the poles of the control system are located in the left phase plane. Thus the system is stable. ■

C. Design of Dynamic QoE-Aware Buffer-Based Controller

If the buffer level is stable, the selected video bitrate for the next chunk will be c_f. However, as discussed, we cannot have the knowledge of the future bandwidth c_k. Even the estimation of c_k is unreliable. Nevertheless, the PID controller can adjust the video bitrate based on the current buffer level. Since the deviation between the current buffer level and the stable buffer level is caused by the improper choice of r_{k-1}, we set

$$r_f = r_{k-1}, \tag{23}$$

for all $k = 1, 2, \ldots, N$, where N is the number of chunks need to be requested.

The details of buffer-based controller are given in Algorithm 1. To ensure the stability of the buffer level, we should initialize K_{p_1}, K_{p_2}, K_d and K_i according to the conditions (22). We use the total bandwidth capacity to download the $(i-1)$-th chunk. Then we update the buffer level b_i and the bitrate adjustment $\delta(r_i)$ according to Eq. (3) and Eq. (20) respectively.

V. Experimental Evaluation

In this section, we provide experimental results of different mechanisms on DASH. In order to evaluate the effectiveness of our proposed controller, we developed a discrete-event simulator to simulate the behavior of a video playback buffer of a mobile device under different DASH requesting strategies and network conditions. First of all, we introduce the experiment setup and specify the parameters.

In the simulation environment, video contents are stored in a streaming server and will be delivered to the client through access networks. The access network can be wired or wireless.

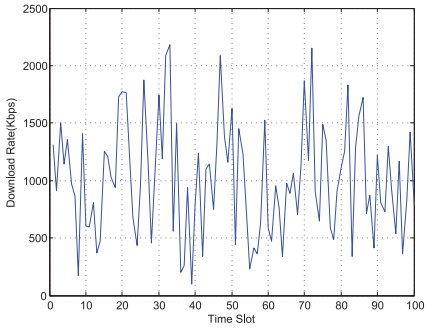


Fig. 7. Download rates of mobile devices (Rayleigh distribution with mean 1000Kbps).

As discussed in Section III, the network condition can be modeled as a stochastic process. Considering the prevalence of mobile devices, we adopt a Rayleigh distribution, which is widely used to simulate the wireless conditions, to approximate the stochastic changes of download rates as [34]. Fig. 7 is an example of download rate fluctuation of wireless network.

A. Parameter Setting

1) System Parameter Setting: In the following experiments, there are 10 feasible video bitrates in the set R (Kbps): $R = \{235, 375, 560, 750, 1050, 1400, 1750, 2350, 3600, 4500\}$. Each video chunk lasts for 4 seconds and the viewing process lasts for 1500 seconds. The buffer length is set as long as 12.5 video chunks, namely, $B_{max} = 50$ seconds. The average available bandwidth is $1050Kbps$.

2) Impact of b_f: We first analyze the impact of b_f on the controller performance. For each different values of b_f, we run 100 simulations under the same system configuration and analyze their total freezing times, average buffer level, average bitrate per chunk and average bitrate switch. In order to facilitate the observation of the impact of b_f, we set $B_{max} = 100$. The result are shown in Fig. 8.

From Fig. 8(a), if b_f is less than 10 seconds, the controller suffers from long freezing time. If b_f is greater than 10 seconds, the total freezing time is negligible. Fig. 8(b) shows that the average buffer level is nearly the same with b_f. Fig. 8(c) shows that if b_f is greater than 10 seconds, the average bitrate per chunk is quite stable. The same pattern for average video bitrate switch appears in Fig. 8(d). A small b_f may cause the system to request a higher video bitrate, thus leads to a higher freezing time and higher average video quality. Generally, a larger b_f means the larger design space for K_{p_1} and K_{p_2}, because the range of the error term $b_k - b_f$ is larger. Nevertheless, it may also result in the smaller design space for K_i since the range of the integral part is larger. For the trade-off between the total freezing time and the average buffer level, we choose $b_f = 20$ in the following simulations.

B. Performance Evaluation

For performance comparison, we evaluate the following algorithms, including buffer-based method and prediction-based method:

- *Prediction method:* The method in comparison is proposed by [35]. It uses historical download rates to

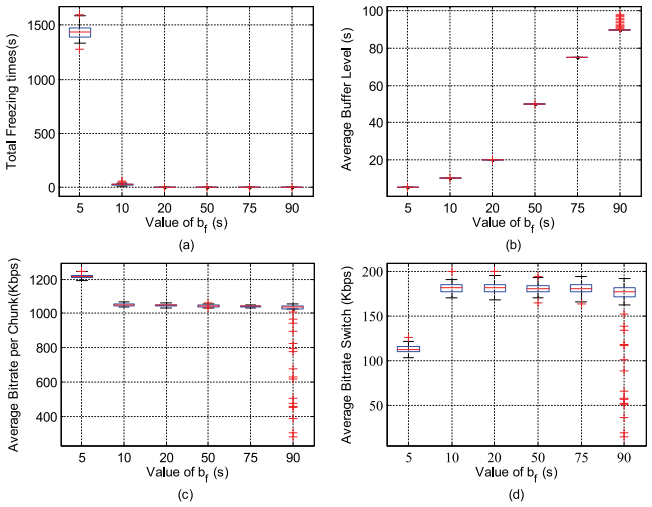


Fig. 8. Performance under different b_f. (a)Total freezing time. (b)Average buffer level. (c)Average bitrate of per chunk. (d)Average video bitrate switch.

predict the future download rate with a smooth factor. The controller will request a video bitrate closest to the predicted result. We use the average download speed of the last 10 chunks for TCP throughput estimation and the dynamic margin is calculated as a function of SI with 0.05 and 0.3. We choose the video bitrate of chunk k closest to the result.

- *Buffer-based approach:* We adopt the algorithm in [8] as the buffer-based approach. We use a linear mapping function, in which the buffer is divided into ten isometric regions. From the front end to the tail end of the buffer, the client will request video bitrate from the lowest to the highest successively.
- *FESTIVE:* We implement the algorithm described in [6]. Since FESTIVE needs to estimate the network capacity, we use the average throughput of the last 10 chunks as the estimation. We set the trade-off factor $\alpha = 12$. And we let n be the number of video bitrate switches over the last 20 seconds. The video bitrate is chosen to minimize the stability score plus α times efficiency score. We do not use a randomized scheduler here.

We set $B_{max} = 50$ and run 100 simulations for each algorithm. The detailed results are shown in Fig. 11.

Fig. 11(a) shows the average bitrate switch of each chunk. The average bitrate switch for PID controller is less than 50. The value for the prediction method is sightly higher than the PID controller. The mean value of the data for the buffer-based approach is around 75. The FESTIVE method has the largest average bitrate switch value.

Fig. 11(b) shows the average buffer level. As we have seen in Fig. 8, the PID controller can maintain a steady average buffer level around b_f. The buffer-based method also achieves a steady buffer level. The average buffer level of prediction method ranges from 15 to 35. FESTIVE leads to a highest average buffer level since it selects much lower bitrates than the other methods.

Fig. 11(c) shows the CDF of the average video bitrate. Except for FESTIVE, the other algorithms achieve nearly the same performance in average video bitrate. Fig. 11(d) shows the CDF of total freezing time for different algorithms. We can see that the freezing time of the buffer-based approach and the PID controller is negligible while FESTIVE and prediction method have a mean value of total freezing time around 30 seconds. Thus, we can see that, an inaccurate prediction may ruin the decision of bitrate.

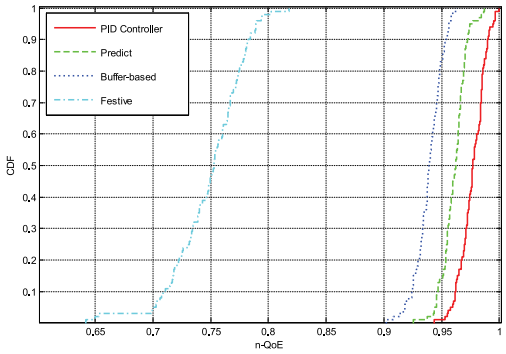


Fig. 9. Normalized QoE for different algorithms.

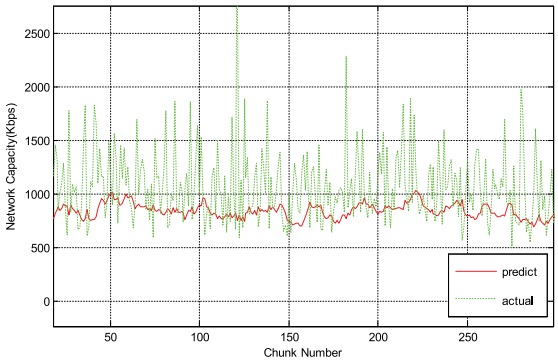


Fig. 10. Predicted network capacity of FESTIVE and the actual network capacity.

C. QoE Metrics and Performance

For QoE analysis, we set $\alpha = 1$ and $\beta = 1$ in Eq. (8). We set $\eta = 8$, which means that one second freezing time equals to a degradation of $4500Kbps$. We set $\mu = 5$, $\lambda = 1$. We find the maximum QoE of the four algorithms and normalize their QoE: n-QoE = $\frac{QoE}{QoE_{max}}$. The result is shown in Fig. 9.

We can see that the PID controller algorithm outperforms the other algorithms. Note that FESTIVE has a poor performance in the simulations, which is because that: (1) The predicted network capacity is inaccurate. The predicted network capacity and the actual network capacity are shown in Fig. 10. (2) FESTIVE puts a higher weight on stability which leads to a lower average video bitrate. It increases the video bitrate slowly but decreases the video bitrate immediately, which causes the switch of bitrates.

D. Sensitivity Analysis

In the following part, we study the sensitivity of the approaches to the key factors, such as: (1) QoE preference, (2) buffer size and (3) startup delay. For each factor, we run

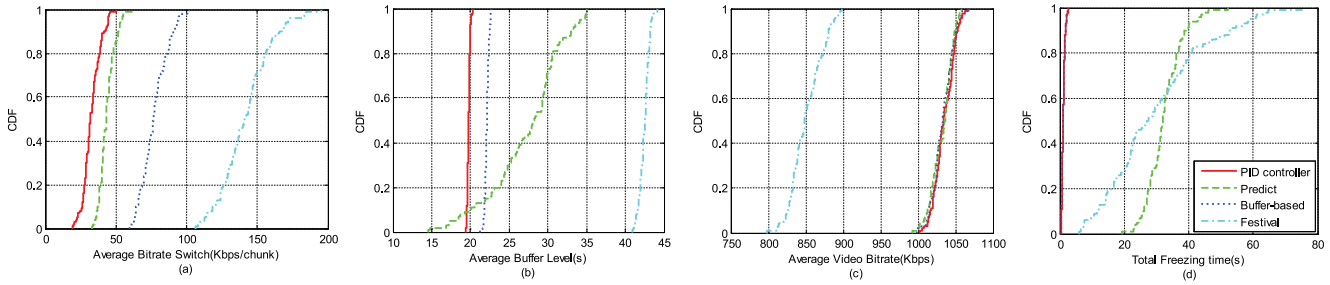


Fig. 11. Detailed performance for different approaches.

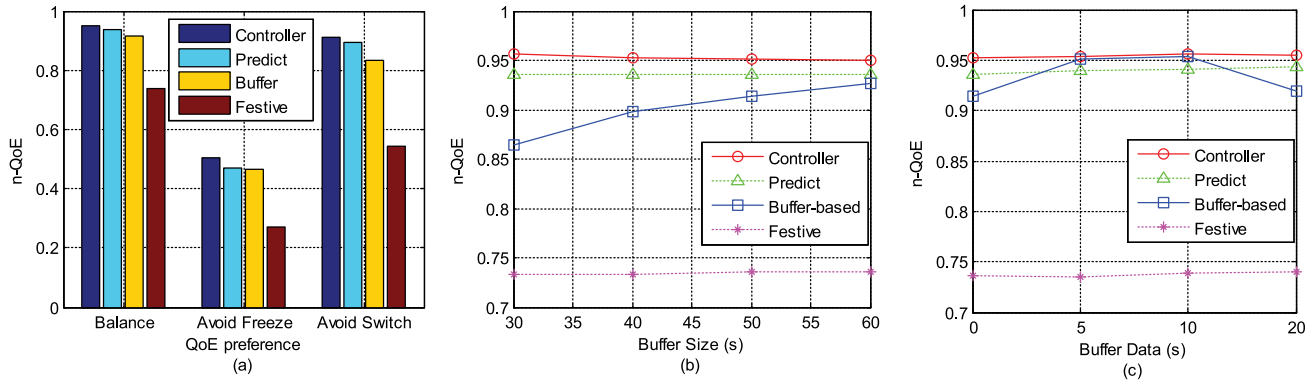


Fig. 12. Sensitivity analysis.

100 simulations and compare the normalized QoE. The results are shown in Fig. 12.

1) User QoE Preferences: We compare the performances of the algorithms under different QoE weights: "Balanced"($\eta = 8, \mu = 5$), "Avoid Freezing Events"($\eta = 16, \mu = 5$) and "Avoid Bitrate Switching"($\eta = 8, \mu = 10$). From Fig. 12(a), we can see that, with a higher weight on freezing events, the QoE of all algorithms are degraded. It is because the freezing score is negative and reduces the QoE. FESTIVE suffers even greater degradation since it has more freezing events. If we put more weight on video bitrate switching, the buffer-based method and the FESTIVE method fall faster, since they have higher bitrate switch values.

2) Buffer Size: We set the buffer size with four different values: $B_{max} = \{30, 40, 50, 60\}$. The result is shown in Fig. 12(b). As we can see, the buffer-based approach will lead to higher QoE as the buffer size increases. It is because with a larger buffer size, the buffer-based approach will be able to absorb more network fluctuations. Other algorithms are almost unaffected.

3) Startup Delay: When the system is initializing, the buffer is empty, which results in the system suffering from startup delay. If the startup delay is long enough, the buffer will be filled with enough video chunks. We compare the performances of the algorithms with different initial buffer states: (1) The buffer is empty, (2) The buffer has 5 seconds video content, (3) The buffer has 10 seconds video content and (4) The buffer has 20 seconds video content. The result is shown in Fig. 12(c). As the initial length of the buffered video content increase from 0 to 5 seconds, the performances of all algorithms increase slightly. But a higher initial buffer state has less impact on the QoE of the algorithms.

Overall, the sensitivity analysis shows the PID controller has a stable performance.

VI. Conclusion

In this paper, we investigated the buffer-based bitrate selection problem for DASH. We formulated the problem as a stochastic optimization problem with an objective to maximize the user QoE. Moreover, we transformed the optimization problem into an optimal stochastic controller design problem. Based on the optimal stochastic control theory, we proved that the optimal static controller cannot be achieved. We then analyzed the structure of the system and proposed a PID controller. We verified the algorithm via extensive simulations. Compared with other strategies, our controller can effectively control the average buffer level, video freezing time and video bitrate switch times. In addition, our controller can achieve a decent QoE level. In the future, we plan to take more QoE metrics into account and implement our algorithm the real systems.

Appendix
Proof of Lemma 1

According to Eq. (5), we can express the QoE score of the k-th chunk period as:

$$Q^k = -q(b_k, c_k, r_k, r_{k-1}, k). \tag{24}$$

We define

$$\hat{F}_0 = \sum_{k=0}^{N-1} \mathbb{E}\Big[-Q^k\Big] = \sum_{k=0}^{N-1} \mathbb{E}\big[q(b_k, c_k, r_k, r_{k-1}, k)\big]. \quad (25)$$

Then the optimization problem (11) can be reformulated as:

$$\min_{\Psi} \ \hat{F}_0, \quad \text{s.t. (3) (4).} \quad (26)$$

Let's consider the (N-1)-th period first. We define

$$\begin{aligned}\hat{F}_{N-1} &= \mathbb{E}\big[q(b_{N-1}, c_{N-1}, r_{N-1}, r_{N-2}, N-1)\big] \\ &= q^*\big(\bar{b}_{N-1}, \bar{c}_{N-1}, r_{N-1}, r_{N-2}, N-1\big), \end{aligned} \quad (27)$$

where $q^*(\cdot)$ is the expectation function of $q(\cdot)$. $\bar{b}_{N-1}$ and $\bar{c}_{N-1}$ are the expectation of b_{N-1} and c_{N-1} respectively. Since $c(t)$ follows the same distribution across different time slots, $\bar{c}_k$ is a constant.

If the optimal bitrate r_{N-1} in (N-1)-th period is found, then the minimal value of $\hat{F}_{N-1}$ will be a function of $\bar{b}_{N-1}$. Define Bellman function $S_{N-1}(\bar{b}_{N-1})$ as the minimal value of $\hat{F}_{N-1}$ as follows:

$$S_{N-1}\big(\bar{b}_{N-1}\big) = \min_{r_{N-1}\in\mathbb{R}} \hat{F}_{N-1}\big(\bar{b}_{N-1}, r_{N-1}, (N-1)\big). \quad (28)$$

Note that in (N-1)-th period, r_{N-2} should be known.

Now consider both (N-1)-th and (N-2)-th period. The cost function of these two periods can be expressed as:

$$\begin{aligned}\hat{F}_{N-2} = {} & q^*\big(\bar{b}_{N-2}, \bar{c}_{N-2}, r_{N-2}, r_{N-3}, N-2\big) \\ & + q^*\big(\bar{b}_{N-1}, \bar{c}_{N-1}, r_{N-1}, r_{N-2}, N-1\big). \end{aligned} \quad (29)$$

According to Eq. (3), the buffer update function can be expressed as:

$$b_{k+1} = f(b_k, c_k, r_k, k). \quad (30)$$

Then we have

$$b_{N-1} = f(b_{N-2}, c_{N-2}, r_{N-2}, N-2), \quad (31)$$

and the expectation of b_{N-1} is:

$$\bar{b}_{N-1} = f^*\big(\bar{b}_{N-2}, \bar{c}_{N-2}, r_{N-2}, N-2\big), \quad (32)$$

where $f^*(\cdot)$ is the expectation function of $f(\cdot)$.

Replacing the terms in (29) with (32), we can see that if the optimal bitrate r_{N-1} and r_{N-2} can be found, the minimal value of $\hat{F}_{N-2}$ is determined by $\bar{b}_{N-2}$. Similar to function S_{N-1}, we can define Bellman function $S_{N-2}(\cdot)$ as follows:

$$\begin{aligned} S_{N-2}\big(\bar{b}_{N-2}\big) &= \min_{\substack{r_{N-1}\in\mathbb{R} \\ r_{N-2}\in\mathbb{R}}} \hat{F}_{N-2} \\ &= \min_{r_{N-2}\in\mathbb{R}} \big\{q^*\big(\bar{b}_{N-2}, \bar{c}_{N-2}, r_{N-2}, r_{N-3}, N-2\big) \\ &\qquad + S_{N-1}\big(f^*\big(\bar{b}_{N-2}, \bar{c}_{N-2}, r_{N-2}, N-2\big)\big)\big\}. \end{aligned} \quad (33)$$

If the value of $\bar{b}_{N-2}$ is known, then we can find the optimal bitrate r_{N-2}. Then the optimal bitrate r_{N-1} can be found according to (28).

Generally, the Bellman function of two adjacent periods can be represented as follows:

$$\begin{aligned} S_k\big(\bar{b}_k\big) = \min_{r_k\in\mathbb{R}} \big\{ & q^*\big(\bar{b}_k, \bar{c}_k, r_k, r_{k-1}, k\big) \\ & + S_{k+1}\big(f^*\big(\bar{b}_k, \bar{c}_k, r_k, k\big)\big)\big\}. \end{aligned} \quad (34)$$

If the value of $\bar{b}_k$ is known before hand, we can obtain the optimal bitrate r_k.

So, if the optimal mapping function exists, then we can have a linear relationship of r_k and b_k like this:

$$r_k = l_k b_k, \quad (35)$$

where l_k is a function of k. But according to (34), r_k is a function of $\bar{b}_k$ and $q^*(\bar{b}_k, \bar{c}_k, r_k, r_{k-1}, k)$, both of them are nonlinear functions on b_k. That is to say, we can not obtain a simple linear relationship of r_k and b_k. Thus, the optimal mapping function dose not exists. ■

REFERENCES

[1] "Cisco visual networking index: Forecast and methodology, 2013–2018," San Jose, CA, USA, Cisco, White Paper, pp. 2013–2018, 2014.

[2] L. De Cicco and S. Mascolo, "An adaptive video streaming control system: Modeling, validation, and performance evaluation," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 526–539, Apr. 2014.

[3] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[4] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. M. Leung, "SOVCAN: Safety-oriented vehicular controller area network," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 94–99, Aug. 2017.

[5] D. De Vleeschauwer *et al.*, "Optimization of HTTP adaptive streaming over mobile cellular networks," in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 898–997.

[6] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol.*, Nice, France, 2012, pp. 97–108.

[7] G. Tian and Y. Liu, "Towards Agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol.*, Nice, France, 2012, pp. 109–120.

[8] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proc. ACM Conf. SIGCOMM*, Chicago, IL, USA, 2014, pp. 187–198.

[9] L. R. Romero, "A dynamic adaptive HTTP streaming video service for Google Android," M.S. thesis, Roy. Inst. Technol., Stockholm, Sweden, 2011.

[10] V. Joseph, S. Borst, and M. I. Reiman, "Optimal rate allocation for adaptive wireless video streaming in networks with user dynamics," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 406–414.

[11] V. Ramamurthi and O. Oyman, "Video-QoE aware radio resource allocation for HTTP adaptive streaming," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, 2014, pp. 1076–1081.

[12] K. Liang, J. Hao, R. Zimmermann, and D. K. Y. Yau, "Integrated prefetching and caching for adaptive video streaming over HTTP: An online approach," in *Proc. 6th ACM Multimedia Syst. Conf.*, Portland, OR, USA, 2015, pp. 142–152.

[13] S.-Y. Chang and H.-T. Chiao, "Adaptive streaming schemes for MPEG-DASH over WiFi multicast," in *Proc. 15th IEEE Int. Conf. Commun. Technol. (ICCT)*, 2013, pp. 168–173.

[14] W. Chen, L. Ma, G. Sternberg, Y. A. Reznik, and C.-C. Shen, "User-aware DASH over Wi-Fi," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Garden Grove, CA, USA, 2015, pp. 749–753.

[15] D. K. Krishnappa, D. Bhat, and M. Zink, "Dashing YouTube: An analysis of using DASH in YouTube video service," in *Proc. IEEE 38th Conf. Local Comput. Netw. (LCN)*, Sydney, NSW, Australia, 2013, pp. 407–415.

[16] W. Ji, Z. Li, and Y. Chen, "Joint source-channel coding and optimization for layered video broadcasting to heterogeneous devices," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 443–455, Apr. 2012.

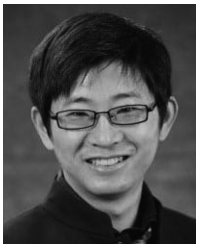
[17] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proc. ACM Conf. Special Interest Group Data Commun.*, London, U.K., 2015, pp. 325–338.
[18] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "Elastic: A client-side controller for dynamic adaptive streaming over HTTP (DASH)," in *Proc. 20th Int. Packet Video Workshop (PV)*, San Jose, CA, USA, 2013, pp. 1–8.
[19] C. Zhou, C.-W. Lin, X. Zhang, and Z. 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.
[20] G. Cofano, L. De Cicco, and S. Mascolo, "A control architecture for massive adaptive video streaming delivery," in *Proc. Workshop Design Qual. Deployment Adapt. Video Streaming*, Sydney, NSW, Australia, 2014, pp. 7–12.
[21] N. Bouten *et al.*, "QoE optimization through in-network quality adaptation for HTTP adaptive streaming," in *Proc. 8th Int. Conf. Workshop Syst. Virtualiztion Manag. (SVM) Netw. Service Manag. (CNSM)*, Las Vegas, NV, USA, 2012, pp. 336–342.
[22] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 20–27, Apr. 2012.
[23] L. De Cicco, S. Mascolo, and V. Palmisano, "Feedback control for adaptive live video streaming," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, San Jose, CA, USA, 2011, pp. 145–156.
[24] Y. Huang, S. Mao, and S. F. Midkiff, "A control-theoretic approach to rate control for streaming videos," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1072–1081, Oct. 2009.
[25] S. Sanadhya and R. Sivakumar, "Adaptive flow control for TCP on mobile phones," in *Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 2912–2920.
[26] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.
[27] W. Ji, B.-W. Chen, Y. Chen, and S.-Y. Kung, "Profit improvement in wireless video broadcasting system: A marginal principle approach," *IEEE Trans. Mobile Comput.*, vol. 14, no. 8, pp. 1659–1671, Aug. 2015.
[28] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, San Jose, CA, USA, 2011, pp. 133–144.
[29] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: Picking a video streaming rate is hard," in *Proc. ACM Conf. Internet Meas. Conf.*, Boston, MA, USA, 2012, pp. 225–238.
[30] T. De Pessemier, K. De Moor, W. Joseph, L. De Marez, and L. Martens, "Quantifying the influence of rebuffering interruptions on the user's quality of experience during mobile video watching," *IEEE Trans. Broadcast.*, vol. 59, no. 1, pp. 47–61, Mar. 2013.
[31] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: Where microeconomics meets psychophysics and quality of experience," *Telecommun. Syst.*, vol. 52, no. 2, pp. 587–600, 2013.
[32] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, Miami, FL, USA, 2013, pp. 389–400.
[33] Y.-W. Fang, *Optimal Control for Stochastic Systems*, Tsinghua Univ., Beijing, China, 2005.
[34] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[35] G. Tian and Y. Liu, "On adaptive HTTP streaming to mobile devices," in *Proc. 20th Int. Packet Video Workshop*, San Jose, CA, USA, 2013, pp. 1–8.

Weiwei Huang received the bachelor's degree in applied mathematics from the South China University of Technology, Guangzhou, China, in 2015. He is currently pursuing the master's degree with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, under the supervision of Prof. D. Wu. His research interests mainly include multimedia networking, data center networking, and online social networks.

Yipeng Zhou received the B.S. degree from the Department of Computer Science, University of Science and Technology of China, the M.Phil. and Ph.D. degrees from the Department of Information Engineering, Chinese University of Hong Kong (CUHK) respectively. He is an Australia Research Council DECRA Researcher and a Research Fellow with the Institute for Telecommunications Research, University of South Australia. He was an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, from 2013 to 2016. From 2012 to 2013, he was a Post-Doctoral Research Fellow with the Institute of Network Coding, CUHK. His main research interests lie in user behavior analysis, video recommendation, information diffusion, reputation systems, and edge computing.

Xueyan Xie received the B.S. degree in microelectronic from the Fudan University, Shanghai, China, in 2013, and the M.S. degree in computer science from the Sun Yat-sen University, Guangzhou, China, in 2016. His research interests include multimedia communication, content distribution networks, and could computing.

Di Wu (M'06–SM'17) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2000, the M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2007. He was a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA, from 2007 to 2009, advised by Prof. K. W. Ross. He is currently a Professor and the Assistant Dean of the School of Data and Computer Science with Sun Yat-sen University, Guangzhou, China. His research interests include cloud computing, multimedia communication, Internet measurement, and network security. He was a co-recipient of the IEEE INFOCOM 2009 Best Paper Award. He has served as an Editor of the *Journal of Telecommunication Systems* (Springer), the *Journal of Communications and Networks*, *Peer-to-Peer Networking and Applications* (Springer), *Security and Communication Networks* (Wiley), and the *KSII Transactions on Internet and Information Systems*, and a Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He has also served as the MSIG Chair of the Multimedia Communications Technical Committee in the IEEE Communications Society from 2014 to 2016. He served as the TPC Co-Chair of the IEEE Global Communications Conference—Cloud Computing Systems, and Networks, and Applications in 2014, the Chair of the CCF Young Computer Scientists and Engineers Forum, Guangzhou from 2014 to 2015, and a member of the Council of China Computer Federation.



Min Chen (M'08–SM'09) has been a Full Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), Wuhan, China, since 2012, where he is the Director of Embedded and Pervasive Computing Lab. He was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University (SNU), where he was a Post-Doctoral Fellow for one and a half years. He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of British Columbia (UBC) for three years. He has authored or co-authored over 300 paper publications, including over 200 SCI papers, over 80 IEEE TRANSACTIONS/Journal papers, 16 ISI highly cited papers, and eight hot papers. He has published four books entitled *OPNET IoT Simulation* (HUST Press, 2015), *Big Data Inspiration* (HUST Press, 2015), *5G Software Defined Networks* (HUST Press, 2016), and *Introduction to Cognitive Computing* (HUST Press, 2017), a book on big data entitled *Big Data Related Technologies* (2014, Springer) and a book on 5G entitled *Cloud Based 5G Wireless Networks* (2016, Springer). His latest book (co-authored with Prof. Kai Hwang), is *Big Data Analytics for Cloud/IoT and Cognitive Computing* (Wiley, 2017). His research interests include cyber-physical systems, IoT Sensing, 5G networks, mobile cloud computing, SDN, healthcare big data, medical cloud privacy and security, body area networks, emotion communications, and robotics. He was a recipient of the Best Paper Award from QShine 2008, IEEE ICC 2012, ICST Industrial IoT 2016, and IEEE IWCMC 2016, and the IEEE Communications Society Fred W. Ellersick Prize in 2017. He is the Chair of IEEE Computer Society Special Technical Communities on Big Data. He is as an Editor or an Associate Editor of *Information Sciences, Information Fusion*, and the IEEE ACCESS. He is a Guest Editor of the IEEE NETWORK, the IEEE WIRELESS COMMUNICATIONS, and the IEEE TRANSACTION SERVICE COMPUTING. He is the Co-Chair of IEEE ICC 2012—Communications Theory Symposium, and the Co-Chair of the IEEE ICC 2013—Wireless Networks Symposium. He is the General Co-Chair of the IEEE CIT-2012, Tridentcom 2014, Mobimedia 2015, and Tridentcom 2017. He is a Keynote Speaker of CyberC 2012, Mobiquitous 2012, Cloudcomp 2015, IndustrialIoT 2016, and the 7th Brainstorming Workshop on 5G Wireless. His Google Scholars Citations reached over 10 500 with an h-index of 50. His top paper was cited over 1000 times.

Edith Ngai received the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2007. She is currently an Associate Professor with the Department of Information Technology, Uppsala University, Uppsala, Sweden. She was a Post-Doctoral Researcher with Imperial College London, London, U.K., from 2007 to 2008. Since 2015, she has been a Visiting Professor with Ericsson Research Sweden. Her research interests include wireless sensor and mobile networks, Internet of Things, network security and privacy, smart city, and e-health applications. She has served as a TPC Member in leading networking conferences, including IEEE ICDCS, IEEE Infocom, IEEE ICC, IEEE Globecom, IEEE/ACM IWQoS, and IEEE CloudCom. She was the TPC Co-Chair of the Swedish National Computer Networking Workshop and QShine14. She is a Program Chair of ACM womENcourage 2015, the TPC Co-Chair of IEEE SmartCity 2015 and IEEE ISSNIP 2015. She has served as a Guest Editor for a special issue of the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, Springer *Mobile Networks and Applications* (MONET), and the *EURASIP Journal on Wireless Communications and Networking*. She was a recipient of the VINNMER Fellow Award by VINNOVA, Sweden, in 2009, and the Best Paper Runner-up Awards of IEEE IWQoS 2010 and ACM/IEEE IPSN 2013 for her co-authored papers. She is a member of the ACM.