*m*DASH: A Markov Decision-Based Rate Adaptation Approach for Dynamic HTTP Streaming

Chao Zhou, Chia-Wen Lin, *Senior Member, IEEE*, and Zongming Guo

***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 paper, we propose a novel Markov decision-based rate adaptation scheme for DASH aiming to maximize the quality of user experience under time-varying channel conditions. To this end, our proposed method takes into account those key factors that make a critical impact on visual quality, including video playback quality, video rate switching frequency and amplitude, buffer overflow/underflow, and buffer occupancy. Besides, to reduce computational complexity, we propose a low-complexity sub-optimal greedy algorithm which is suitable for real-time video streaming. Our experiments in network test-bed and real-world Internet all demonstrate the good performance of the proposed method in both objective and subjective visual quality.**

***Index Terms*—Dynamic adaptive streaming over HTTP (DASH), Markov decision, quality of experience, rate adaptation.**

I. Introduction

DYNAMIC adaptive streaming over HTTP (DASH) has been recently widely adopted for providing uninterrupted video streaming services to users with dynamic network conditions and heterogeneous devices [1], [2]. In contrast to the past RTP/UDP, the use of HTTP over TCP is easy to configure and, in particular, can greatly simplify the traversal of firewalls and network address translators. Besides, the deployment cost of DASH is relatively low since it employs standard HTTP servers and, therefore, can easily be deployed within content delivery networks. In DASH, a video clip is encoded into multiple versions at different bitrates, each being further divided into small video fragments containing seconds or tens of seconds worth of video. At the client side, a DASH client continuously requests and receives video fragments from the DASH servers that own the fragments. To adapt the video bitrate to a varying network bandwidth, DASH allows clients to request video fragments from different versions of a video, each of which being coded with a specific bitrate. This is known as dynamic rate adaptation, which is one of the most important features of DASH since it can automatically throttle the visual quality to match the available bandwidth so that a user receives the requested video at the maximum quality possible.

Since network conditions can be highly dynamic, it is very challenging to provide satisfactory user experience during an entire video session. Without an effective rate adaption algorithm, a DASH client may suffer from frequent interruptions and significant visual quality degradation. For example, a video bitrate higher than the available bandwidth would cause network congestion and, on the other hand, when the video bitrate is lower than the available bandwidth, the visual quality cannot reach the maximum allowed by the available bandwidth.

There are several rate adaptation schemes proposed for DASH, such as bandwidth-based schemes[1] [3]–[4] and buffer-based schemes [5]–[7]. The bandwidth-based schemes mainly aim to dynamically adapt the video bitrate to an available bandwidth, which usually leads to a low bandwidth utilization and cannot reach the maximum quality allowed by the available bandwidth. This is because, in such schemes, the video bitrates higher than the available bandwidth are never allowed to be selected to avoid playback interruptions. On the other hand, buffer-based schemes are focused on stabilizing the buffer occupancy within a certain range to ensure continuous video playback. However, such kind of schemes generally lead to frequent bitrate switching which could be visually very annoying [8]. This is mainly because there is a trade-off between the stability of buffer occupancy and the smoothness of video bitrate due to the time-varying bandwidth.

To consider the influence of a rate selection decision on its future fragments to be downloaded, Markov decision-based approaches have been proposed to model the dynamics of a video streaming system under time-varying network conditions [9]–[12]. However, these existing schemes do not comprehensively take into account the factors that that make critical influence on visual quality. Moreover, the existing schemes usually resort to a single reward mechanism for a whole streaming session, making it not well adapted to the dynamic network conditions, thereby degrading quality of experience (QoE). This motivates us to propose a more effective Markov-decision-based rate adaptation scheme which has the advantages of both the bandwidth-based and buffer-based schemes, while getting rid of their disadvantages.

In this paper, by extending our previous works in [13] and [14], we formulate rate adaption for DASH as a Markov

Manuscript received February 18, 2015; revised July 10, 2015 and November 5, 2015; accepted January 11, 2016. Date of publication January 27, 2016; date of current version March 15, 2016. This work was supported in part by the Ministry of Science and Technology, Taiwan under Grant MOST 100-2628-E-007-026-MY3 and Grant MOST 103-2221-E-007-046-MY3. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Christian Timmerer. *(Corresponding author: Chia-Wen Lin.)*

C. Zhou and Z. Guo are with the Institute of Computer Science & Technology, Peking University, Beijing 100871, China (e-mail: zhouchaoyf@gmail.com; guozongming@pku.edu.cn).

C.-W. Lin is with the Department of Electrical Engineering and the Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan 30013, R.O.C. (e-mail: cwlin@ee.nthu.edu.tw).

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/TMM.2016.2522650

[1] "Open source media framework," [Online]. Available: http://www.osmf.org/

decision-based optimization problem. In our method, a state vector is defined to describe the current system situation, including the current buffer occupancy and its changing rate, the video bitrates for previously downloaded fragments, the bitrate consistency function, and the bandwidth conditions. Accordingly, based on the probabilistic characteristics of system states, the transition probability matrix comprising the transfer probabilities between every two consecutive states is derived. Considering those factors affecting QoE for a video streaming session, including video playback quality, video rate switching frequency and amplitude, buffer overflow/underflow, and buffer occupancy, we propose a reward function for three different scenarios of buffer occupancy to measure the effectiveness of each rate-switching decision. As a result, the optimal streaming policy, i.e., the best video bitrates for all fragments, can be found by maximizing the long-term reward.

The main contribution of this paper is threefold.

1) We formulate rate adaption for DASH as a Markov decision-based optimization problem. To improve the system performance, we systematically characterize the behavior of a DASH video streaming system in terms of the buffer occupancy transfer model, video rate switching, and channel condition transfer model.
2) We propose a reward function to take into account key factors that affect visual quality, including video playback quality, video rate switching frequency and amplitude, buffer overflow/underflow, and buffer occupancy. Accordingly, we propose to dynamically adapt the reward function to three different scenarios.
3) To reduce computation, we propose an efficient suboptimal greedy rate-adaptation algorithm without significantly sacrificing visual quality.

II. Related Work

Although DASH is a relatively new application, due to its popularity, it has attracted much research effort recently. Watson systematically introduced the DASH framework of Netflix [15], which has been the largest DASH stream provider in the world.

As mentioned above, dynamic rate adaptation is one of the most important features of DASH since it can automatically throttle the visual quality to match the available bandwidth so that each user receives the video with the maximum quality possible. Akhshabi *et al.* [16] compared the rate adaption schemes used for three popular DASH clients: Netflix client [15], Microsoft Smooth Streaming [17], and Adobe OSMF. It was reported in [16] that none of the DASH client-based rate adaptation is good enough, as they are either too aggressive or too conservative. Some clients even just switch between the highest and lowest bitrates. Also, all of them lead to relatively long response time under the shift of network congestion level. Existing rate adaptation schemes for DASH, such as bandwidth-based schemes [3]–[4] and buffer-based schemes [5]–[7], aim to either achieve a high bandwidth utilization efficiency by dynamically adapting the video bitrate to an available bandwidth, or maintain continuous video playback by smoothing the video bitrate to avoid buffer overflow/underflow. Nevertheless, due to unavoidable bandwidth variations, existing schemes usually cannot achieve a good tradeoff between video bitrate smoothness and bandwidth utilization. In our previous work [13], a dual-threshold-based buffer occupancy model was proposed to smooth out the short-term bandwidth variations so as to maintain the smoothness of video rate. To avoid buffer overflow and playback interruptions, the video rate is dynamically regulated by a PD controller which has proven to effectively mitigate buffer overflow/underflow. The main objective of the rate adaption method in [13] is, however, to avoid buffer overflow and underflow without considering other factors, such as switching frequency and amplitude, which also can affect the perceived visual quality [18]. Furthermore, the method does not consider the influence of the current rate selection decision on the future fragments to be downloaded, making it unable to achieve the optimal performance possible.

QoE is defined in [19] as the overall acceptability of an application or service, as perceived subjectively by an end-user, which typically involves several factors such as network, client, and terminal. Under time-varying network conditions, the objective of rate adaptation for DASH is to maximize the QoE, which is influenced by several factors, such as video playback quality [12], [20], [21], video rate switching frequency and amplitude [12], [18], [22], buffer overflow/underflow [6], [7], and buffer occupancy [23]. In [18], Mok *et al.* presented a QoE-aware DASH system that estimates the bandwidth by probing with the video data and keeping the video rate as smooth as possible. It was shown that users generally prefer a gradual quality change to an abrupt switching. The work in [12] addresses the issues of DASH with scalable video coding, in which both the visual quality and bitrate smoothness are considered when making rate-switching decision. There are also some other works addressing issues in QoE for DASH [11], [22]–[24]. Nevertheless, the existing methods only take into account part of the critical factors of DASH, which usually may not achieve the best performance.

Fig. 1 illustrates the rate switching process for DASH prior to downloading each fragment, where the segment's bitrate is determined based on the current system state and network condition. The bitrates of all fragments form a streaming policy and the aim is to find the best streaming policy that maximizes the QoE. Markov decision process (MDP) has been employed in rate adaptation for DASH in the literature as it can be used to effectively model the dynamics of a video streaming system under time-varying network conditions. For example, in [25], rate selection is performed offline by an MDP assuming that the available bandwidth can be estimated using a transition matrix. Applying the model on-line, however, may result in inaccurate estimations due to unpredictable characteristics of network conditions. This work is further extended in [9], [10]. However, no subjective validation was conducted to convincingly justify the proposed rate adaption scheme and visual quality model. In [11], [12], a stochastic dynamic programming (SDP) technique was proposed for rate adaption for DASH, where the system rate is determined based on client buffer occupancy and bandwidth condition. While selecting the rate adaptation strategy, a cost function involving the video rate level, video rate switching and video freezes was proposed.

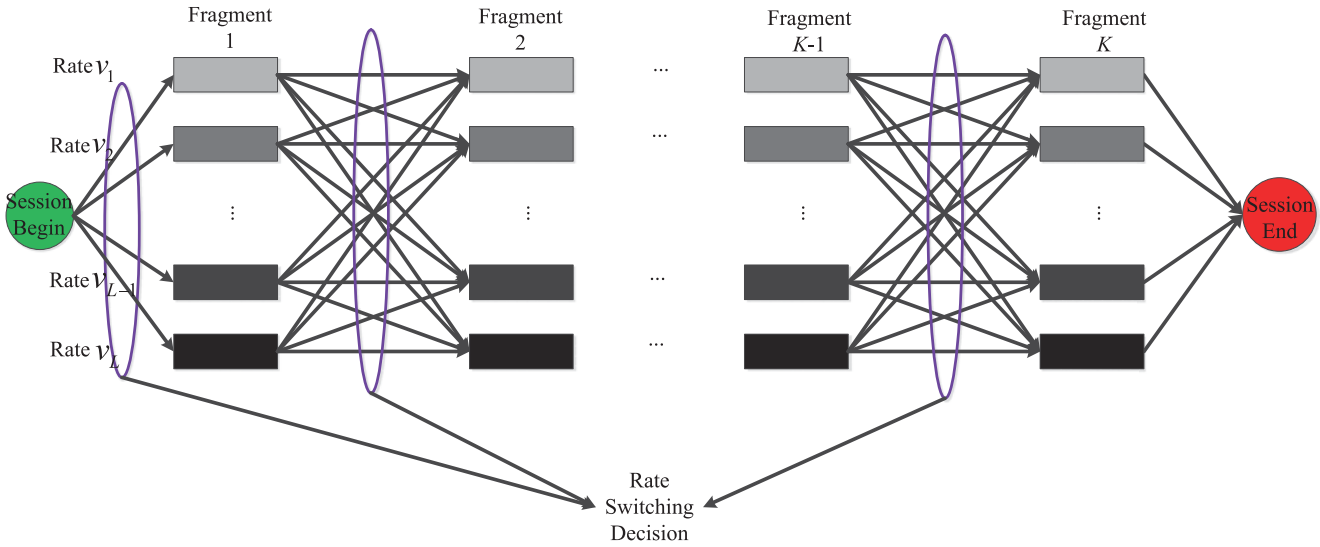


Fig. 1. Rate switching process for a whole DASH streaming session.

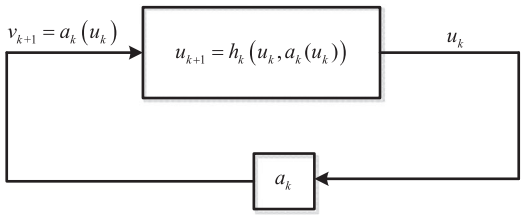


Fig. 2. System state evolution model. u_k denotes the system state at stage k, according to u_k, the next requested video rate is derived under the action a_k, and the state moves to u_{k+1} through system function h_k.

III. Formulation of Rate Adaption

Without loss of generality, in our proposed DASH system, the video bitrate can be adapted only when the last fragment has been downloaded completely. Besides, since the time-varying network bandwidth can greatly affect the decision of the video bitrate switching, the rate adaption process can be considered as a discrete-time stochastic system that evolves along time in stages. Considering the Markov property of the system states, an MDP can be adopted in DASH for bitrate adaption. We therefore propose an MDP-based DASH (or *m*DASH for short) in this paper.

In order to apply MPD techniques, the state transition model of *m*DASH needs to be devised. In *m*DASH, a rate decision is made at stage k for fragment $k+1$, so the total number of stages equals the number of fragments K. For stage k, we denote the state as u_k, which contains all the information gathered from the network once fragment k has been completely downloaded, and the controller applies control action a_k to determine the video rate for fragment $k+1$ based on the information in state u_k. In this work, the output of a control action is the video bitrate to be requested, i.e., $v_{k+1} = a_k(u_k)$. Then, the current state u_k, along with action a_k, determines the evolution of the system state towards state u_{k+1}, through system function h_k as Fig. 2 shows.

Now, we will give the details of the considered system parameters that make significant influence on the states and actions. Our previous work [13] shows that the buffered video time and its changing rate play important roles in rate switching decision. Also, the estimated bandwidth is indispensable in rate adaptation schemes. Thus, similar to what we did in [13], these three system parameters are considered in this work as well. Besides, it has been shown that video bitrate switching frequency and amplitude also have great effect on the QoE [18], [22]. Therefore, we consider two additional system parameters: the historic video bitrate vector and the video bitrate consistency function. As a result, in *m*DASH, each state vector u_k involves five system parameters as follows:

$$u_k = (q_k, q'_k, \mathbf{v}_k, \lambda_k, b_k) \tag{1}$$

where q_k is the buffered video time (buffer occupancy, in this work, we use buffered video time and buffer occupancy interchangeably) once fragment k has been completely downloaded, q'_k is the average changing rate of the buffered video time when downloading fragment k, $\mathbf{v}_k = [v_{k-N+1}, v_{k-N+2}, \cdots v_{k-1}, v_k]$ is a $1 \times N$ video rate vector, where v_k is the video rate assigned to fragment k, b_k is the average bandwidth during the period of downloading fragment k which is also used as the estimated available bandwidth for downloading fragment $k+1$, and λ_k is the video bitrate consistency function with $\lambda_k = 1$ indicating that all the latest N fragments have the same video rate; otherwise, $\lambda_k = 0$, i.e.

$$\lambda_k = \begin{cases} 1, & \text{if } v_{k-N+1} = v_{k-N+2} = \cdots = v_k \\ 0, & \text{else.} \end{cases} \tag{2}$$

According to the Markov property, the state at any time instance only depends on its immediately previous state. Given any state u_k and actions a_k, the transition probability of the MPD can be given as

$$\mathcal{P}(u_{k+1}|u_k, a_k) = \Pr(u_{k+1}|u_k, a_k(u_k)). \tag{3}$$

In order to evaluate the effectiveness of an action, we define a reward value r_k associated with action a_k at stage k as a function of state u_k, i.e., $r_k = R(u_k)$. We then define ψ as the streaming policy as a mapping of the action taken at each stage. Then, the long-term reward $\mathcal{R}^{\psi}(u_k)$ under policy ψ can be computed by

$$\mathcal{R}^{\psi}(u_k) = \sum_{u_{k+1}} \mathcal{P}(u_{k+1}|u_k, a_k(u_k))(r_k + \gamma \mathcal{R}(u_{k+1})) \tag{4}$$

where $\gamma \in [0,1]$ is a discount parameter reflecting the present value of future reward where a small γ leads to "myopic" evaluation, where as a large γ leads to "far-sighted" evaluation.

Our goal is to find the optimal strategy policy ψ^* that maximizes the reward during streaming. To this end, the video rate adaptation process can be formulated as the following optimization problem:

$$\psi^* = \arg\max \mathcal{R}^{\psi}(u_k). \tag{5}$$

In order to find the optimal streaming policy ψ^* in (5), we need to derive the state transition probability $\mathcal{P}(u_{k+1}|u_k, a_k)$ and devise the reward function r_k for each state k associated with action a_k as will be elaborated below.

IV. Transition Probability Evolution

In this section, we derive the transition probability evolution defined in (3) by considering all the five system parameters.

A. Buffered Video Time Model

To maintain continuous playback, a video streaming client normally contains a video buffer to absorb temporary mismatch between the video downloading rate and video playback rate. In conventional single-version video streaming, the buffered video playback time can be easily measured by dividing the buffered video size by the average video playback rate. In DASH, however, different video versions have different video playback rates. Since a video buffer contains fragments from different versions, there is no longer a direct mapping between the buffered video size and the buffered video time. To tackle the problem, we use the buffered video time to measure the length of video playback buffer.

The buffered video time process, represented as $q(t)$, can be modeled as a queue with a constant service rate of unity, i.e., in each second, a piece of video with playback length of one second is dequeued from the buffer and then played. The enqueue process is driven by the video download rate and the downloaded video version. Specifically, we assume a video clip is encoded into L different versions, with different playback rates $V_1 < V_2 < \ldots < V_L$. All versions of the video are partitioned into equal-length fragments, each of which consuming the same playback time of T. We adapt the video bitrate when a fragment has been downloaded completely. Without loss of generality, suppose a client starts downloading fragment k at time instant t_k^s and the fragment is downloaded completely at t_k^e. Then, we have

$$t_{k+1}^e - t_{k+1}^s = \frac{v_{k+1}}{b_k} T = \frac{a_k(u_k)}{b_k} T. \tag{6}$$

And the buffered video time evolution becomes

$$q(t_{k+1}^e) = q(t_{k+1}^s) + T - \frac{a_k(u_k)}{b_k} T \tag{7}$$

where the second term of (7) is the added video time upon the completion of the downloading of fragment $k+1$, and the third term reflects the fact that the buffered video time is consumed linearly at a rate of unity during the downloading process. Therefore, when the fragment are requested continuously, we have $t_{k+1}^s = t_k^e$ and

$$q_{k+1} = q(t_{k+1}^e) = q_k + T - \frac{a_k(u_k)}{b_k} T. \tag{8}$$

On the other hand, if the bandwidth is too high, a sleep mechanism is used to postpone the the fragment request so as to avoid buffer overflow [13]. Assuming the sleeping time is τ_s, we have $t_{k+1}^s = t_k^e + \tau_s$ and

$$q_{k+1} = q(t_{k+1}^e) = q_k + T - \frac{a_k(u_k)}{b_k} T - \tau_s. \tag{9}$$

For the changing rate of buffer occupancy, since it is hard to obtain the accurate expression for any time instance, we propose to use the fluid approximation [26], which evenly distributes the added video time over the download interval for the whole fragment, then we have

$$\begin{aligned} q'_{k+1}(t)\, l = & \frac{dq(t)}{dt} \\ \approx & \frac{q(t_{k+1}^e) - q(t_{k+1}^s)}{t_{k+1}^e - t_{k+1}^s} \\ = & \frac{b_k}{a_k(u_k)} - 1, t \in (t_{k+1}^s, t_{k+1}^e]. \end{aligned} \tag{10}$$

B. Video Rate Switching Model

According to the above-mentioned system evolution and state definition, the video rate of fragment $k+1$ is determined by the associated state and action as follows:

$$v_{k+1} = a_k(u_k). \tag{11}$$

Thus, given state u_k and action a_k, the video rate vector and video rate consistency function are updated as

$$\mathbf{v}_{k+1} = [v_{k-N+2}, v_{k-N+3}, \cdots, v_k, a_k(u_k)] \tag{12}$$

$$\lambda_{k+1} = \begin{cases} 1, & \text{if } v_{k-N+2} = v_{k-N+3} = \cdots = v_k = a_k(u_k) \\ 0, & \text{else} \end{cases} \tag{13}$$

where $v_{k-N+2}, v_{k-N+3}, \cdots, v_k$ are given in state u_k.

C. Markov Channel Model

In this work, the smoothed throughput is used for bandwidth estimation. However, instead of directly using the smoothed throughput, a heterogeneous and time-varying Markov model

is used to estimate the future bandwidth [27], [28]. To introduce the channel model based on Markov theory, we first divide the bandwidth into several regions, each presenting a state of the Markov channel model, where the total number of states equals the number of regions. Assuming that there are C states, i.e., the bandwidth is divided into C regions, the following $C \times C$ transition matrix is used to characterize the state transitions of a Markov channel at stage k:

$$\mathbf{P}^k = \begin{pmatrix} p_{11}^k & \cdots & p_{1C}^k \\ \vdots & \ddots & \vdots \\ p_{C1}^k & \cdots & p_{CC}^k \end{pmatrix} \tag{14}$$

where element p_{ij}^k denotes the transition probability from state i to state j.

The matrix is initialized with $p_{ij}^0 = \frac{1}{C}, \forall i \le C, j \le C$. Assuming that the smoothed throughput of downloading fragments k and $k+1$ falls in region i' and j' respectively, the matrix is then updated after successfully downloading fragment $k+1$ as follows:

$$p_{ij}^{k+1} = \begin{cases} \dfrac{C * p_{ij}^k + 1}{C+1}, & \text{if } \ i = i' \& j = j' \\ \dfrac{C * p_{ij}^k}{C+1}, & \text{if } \ i = i' \& j \neq j' \\ p_{ij}^k, & \text{else.} \end{cases} \tag{15}$$

Note, the number of states C can be adjusted considering that a large C will generate smaller quantification intervals and hence provide a more accurate bandwidth predication, while increasing the number of state-variables and computational complexity.

D. Transition Probability

The set of transition probabilities between consecutive system states is used to characterize the system evolution. For a selected action a_k, the transition probability from state u_k to u_{k+1} is given by (3), which can be rewritten as

$$\begin{aligned} &\mathcal{P}\left(u_{k+1} | u_k, a_k\right) \\ &\quad = \Pr\left(q_{k+1}, q'_{k+1}, \mathbf{v}_{k+1}, \lambda_{k+1}, b_{k+1} \,|q_k, q'_k, \mathbf{v}_k, \lambda_k, b_k, a_k\left(u_k\right)\right) \end{aligned} \tag{16}$$

Considering that the system parameters are independent of each other, from (8)–(9), we can find that *1*) buffer occupancy q_{k+1} is a function of q_k, b_k, and $a_k\left(u_k\right)$; *2*) the changing rate of buffer occupancy q'_{k+1} is a function of q_k and $a_k\left(u_k\right)$ as shown in (10); *3*) (11)–(13) show that video rate vector $\mathbf{v}_{k+1}$ and video rate consistency function λ_{k+1} are only dependent on action $a_k\left(u_k\right)$; and *4*) the bandwidth model is independently formulated by a Markov channel. Besides, for a given state and action at stage k, q_k, b_k, and $a_k\left(u_k\right)$ are constant. Therefore, (16) can be further rewritten as

$$\begin{aligned} &\mathcal{P}\left(u_{k+1} | u_k, a_k\right) \\ &\quad = \Pr\left(q_{k+1} \,|q_k, b_k, a_k\left(u_k\right)\right) * \Pr\left(q'_{k+1} \,|b_k, a_k\left(u_k\right)\right) \\ &\qquad * \Pr\left(\mathbf{v}_{k+1} \,|\mathbf{v}_k, a_k\left(u_k\right)\right) * \Pr\left(\lambda_{k+1} \,|\lambda_k, a_k\left(u_k\right)\right) \\ &\qquad * \Pr\left(b_{k+1} \,|b_k\right) \end{aligned} \tag{17}$$

where at the right hand side of (17), the first and second terms about buffer occupancy can be obtained by (8)–(10), the third and fourth terms about video rates can be obtained by (11)–(13), and the last term is derived from the Markov channel model in Section IV-C.

V. Reward for Rate Switching

In this section, we propose a reward function to measure the effectiveness of an action. The reward function is then used to find the optimal video rate that maximizes the QoE. Thus, when we determine the actions, i.e., the bitrates of video segments to be requested, we need to consider the factors that affect the QoE. In this work, we take into account the effects of video playback quality, video rate switching frequency and amplitude, buffer overflow/underflow, and buffer occupancy on user experience.

For the video playback quality and video rate switching frequency and amplitude, the mean video rate and temporal variance of video rate associated with the latest N segments are used to measure the rewards. Therefore, for action a_k, the mean video rate m_k and temporal variance σ_k are calculated by

$$m_k = \frac{1}{N-1}\left(\sum_{i=1}^{N-1} v_{k+i-N+1} + a_k\left(u_k\right)\right) \tag{18}$$

$$\sigma_k = \sqrt{\frac{\sum_{i=1}^{N-1}\left(v_{k+i-N+1} - m_k\right)^2 + \left(a_k\left(u_k\right) - m_k\right)^2}{N-1}} \tag{19}$$

and the rewards of these two factors by taking action a_k are denoted by r_{m_k} and r_{σ_k} respectively.

On the other hand, from the control point of view, there may exist fundamental conflict between stabilizing video rate and maintaining stable buffer occupancy, due to unavoidable network bandwidth variations. Nevertheless, from the end user point of view, video rate fluctuations are much more visually perceivable than buffer size oscillations. The recent work in [18] shows that switching back-and-forth between different rates can significantly degrade a user's viewing experience, whereas buffer occupancy variations do not have direct impact on visual quality as long as the video buffer dose not deplete. But playback freeze (due to buffer underflow) generally has a much more negative impact on user experience than rate switching [23], [29].

$$v_{k+1}^{PD} = \begin{cases} v_0 + \dfrac{q_k}{T}\left(K_p\left(q\left(t_{k+1}^s\right) - q_0\right) + K_d q'_k\right), & \text{if } q\left(t_{k+1}^s\right) > q_{\text{high}} \text{ or } q\left(t_{k+1}^s\right) > q_{\text{low}} \\ v_k, & \text{else} \end{cases} \tag{20}$$

To address this problem, Our previous work [13] proposed to use two buffer thresholds, q_{high} and q_{low}, as the operating points for rate switching decision to avoid buffer overflow/underflow. In this method, the video rate is selected by a PD controller when the buffered video time is higher than q_{high} or lower than q_{low}; otherwise, it keeps unchanged. Thus, the video rate for fragment $k+1$ is given in (20), at the bottom of the previous page, where v_0 denotes the video rate at the operating point, which equals the estimated bandwidth q_k, K_p and K_d are the proportional coefficient and differential coefficient, respectively, and q_0 is the operating point. More details about this algorithm can be found in [13], which shows that video rate selected by the PD controller can mitigate buffer underflow/overflow effectively. Therefore, we use the difference between video rates v_{k+1}^{PD} and $a_k(u_k)$ to measure the reward of buffer overflow/underflow (denoted by r_{f_k}) associated with action a_k.

Moreover, to maintain continuous video playback, the buffer occupancy needs to be controlled in a certain range, too high or too low of the buffer occupancy runs the risk of buffer overflow or underflow. Therefore, the decision on action a_k must consider the influence of buffer occupancy, and the reward of buffer occupancy associated with action a_k is denoted as r_{o_k}.

At last, the overall reward of action a_k under state u_k is defined as the linear combination of the factors discussed above

$$r_k = \mathcal{R}(u_k) = a * r_{m_k} + b * r_{\sigma_k} + c * r_{f_k} + d * r_{o_k} \quad (21)$$

where parameters a, b, c, d are used to weight the four factors properly with $a+b+c+d=1$.

In the following, we will derive the four reward functions involved in r_k under three scenarios, since those factors that a user concerns are generally different under different streaming scenarios. For example, when the buffer occupancy is high, we should pay more attention on avoiding buffer overflow by increasing the video playback quality (rate). On the other hand, when the buffer occupancy is low, reducing the video rate to avoid playback interruptions is more urgent.

A. Buffer Overflow Control

When $q\left(t_{k+1}^s\right) > q_{\text{high}}$, buffer overflow needs to be avoided by selecting a higher video rate. As reported in [30], user experience follows the logarithmic law, and the QoE function can be modeled in a logarithmic form for applications of file downloading and web browsing. As such, we use the logarithmic function of the mean video rate as the reward of video rate switching amplitude

$$r_{m_k} = \ln(m_k + \varepsilon) \quad (22)$$

where ε is a small positive number with $\varepsilon > 1$ to ensure that $r_{m_k} > 0$.

While for the video switching frequency, if the video rates for the previous N fragments remain stable, we can ignore its effect on visual quality since user is sensitive to frequent short-term rate fluctuations rather than long-term rate switchings. Otherwise, we use the following logarithmic function to represent the temporal variance:

$$r_{\sigma_k} = \begin{cases} -\ln(\sigma_k + \varepsilon), & \text{if } \lambda_k = 0 \\ 0, & \text{if } \lambda_k = 1. \end{cases} \quad (23)$$

On the other hand, it is expected to take an action to drag the buffer occupancy to be not higher than q_{high} to avoid buffer overflow. For this single purpose, the PD controller in (20) has proven to be effective for rate selection [13]. Thus, r_{f_k} is defined as

$$r_{f_k} = -\ln\left(\left|a_k(u_k) - v_{k+1}^{PD}\right| + \varepsilon\right) \quad (24)$$

and the reward of buffer occupancy is simply measured by the gap with the bound of buffer overflow q_{high}

$$r_{o_k} = q_{\text{high}} - q_{k+1}. \quad (25)$$

B. Buffer Underflow Control

When $q\left(t_{k+1}^s\right) < q_{\text{low}}$, buffer underflow needs to be avoided by taking action a_k, i.e., a lower video rate should be selected to ensure continuous video playback. Similar to the case that $q\left(t_{k+1}^s\right) > q_{\text{high}}$, the rewards of video rate switching amplitude, switching frequency, and buffer underflow associated with action a_k are the same as in (22)–(24). The reward of buffer occupancy in (25) is rewritten as

$$r_{o_k} = q_{k+1} - q_{\text{low}}. \quad (26)$$

C. Smooth Rate Control

When $q_{\text{low}} \le q\left(t_{k+1}^s\right) \le q_{\text{high}}$, the probability of buffer underflow/overflow is low. The reward of video rate switching amplitude and switching frequency associated with action a_k are therefore set as the same as (22)–(23). Besides, the reward of buffer underflow/overflow is simply set as

$$r_{f_k} = 0 \quad (27)$$

considering the low risk of buffer underflow/overflow when the buffer occupancy stays at a moderate level.

Moreover, the ideal case is to keep the buffer occupancy at the middle of the two buffer thresholds so as to maximize the safe margin for avoiding buffer underflow/overflow. Therefore, the reward of buffer occupancy is defined as

$$r_{o_k} = -\left|q_{k+1} - \frac{q_{\text{low}} + q_{\text{high}}}{2}\right|. \quad (28)$$

With the transition probability in Section IV and the reward functions mentions above, the optimal rate adaption policy can be found by solving the optimization in (5).

VI. Low-Complexity Sub-Optimal Greedy Algorithm

Directly solving the optimization problem in (5), however, poses several challenges. First, solving (5) implies to determine the video rates for all fragments when making rate adaptation

decisions, which is impractical since there are quite a few uncertainties during a streaming session due to time-varying network and user conditions. Besides, the computational complexity increases exponentially with the total number of fragments K, making it too expensive to support on-line video streaming services. Therefore, we propose a sub-optimal algorithm that greedily selects the video rates for individual fragments sequentially, instead of determining the video rates for all fragments concurrently.

In *m*DASH, the current state involves all relevant information contained in historical data. Once the state is known, the historic data can then be thrown away, since the current state provides sufficient statistics for estimating the future states. On the other hand, action a_k at stage k not only affects reward r_k, but also affects state u_{k+1}, which further affects action a_{k+1} at stage $k+1$. This effect will propagate to the future until all the fragments are completely downloaded.

Since more future states are considered in (4), generally better actions can be selected at stage k. Meanwhile, the search time consumed by a rate-switching action increases exponentially with the number of considered future states. Therefore, there is a trade-off between visual quality and computational complexity. In *m*DASH, when the buffered video time satisfies that $q_{\text{low}} \leq q\left(t^s_{k+1}\right) \leq q_{\text{high}}$, continuous video playback can be guaranteed in the near future, and there is sufficient time to take more future states into consideration to improve QoE. Otherwise, when the buffered video time satisfies that $q\left(t^s_{k+1}\right) > q_{\text{high}}$ or $q\left(t^s_{k+1}\right) < q_{\text{low}}$, it is likely that a buffer overflow/underflow may happen soon, and thus less future states are preferred. Therefore, in this work, the reward gained at stage k is revised as shown in (29) at the bottom of the page, which indicates that when the buffer occupancy is too high or too low, we make the action decision without taking any future states into consideration; otherwise, one additional future state is considered. Though it is a sub-optimal strategy compared to the reward in (4), the state and reward definitions have considered the long-term effects of video rates, thereby still doing a good job as will be validated by our experimental results.

VII. Performance Evaluation

In this section, we evaluate our rate adaption algorithms by conducting both controlled experiments on a network test-bed and uncontrolled experiments in real Internet. In addition, the subjective quality of received video is also evaluated.

A. Experiment Setup

We implement our *m*DASH system in linux/unix platforms. Our testbed consists of three nodes: one web server used for media delivery, one router, and one DASH client. The server and client run the standard Ubuntu of version 12.04.1. The server is installed with the Apache HTTP server of version 2.4.1. Dummynet is used in the server to control the upload bandwidth [31], therefore, the bottleneck is the bandwidth between the server and the router. In our experiments, same as Netflix, the server provides five different versions of video bitrates: 300 Kbps, 700 Kbps, 1.5 Mbps, 2.5 Mbps, and 3.5 Mbps. Each video is divided into equal-length video fragments with a length of 2 s.

Due to the complex network characteristics, it is hard to find the optimal values of the two thresholds $q_{\max}$ and $q_{\min}$ [13]. In our experiments, we set $q_{\min} = 5$ s since this start-up delay can be tolerated by most existing streaming systems, and $q_{\max} = 25$ s considering that it is reasonable for current existing devices, including mobile phones, to buffer such a length of media data. The maximal buffer size is set to 30 s for all schemes in this work. However, In general, a better performance can be achieved with a larger buffer size if the buffer delay is allowed. This is because with a large buffer size, the bandwidth variations can be compensated with the buffered video and a smooth video rate is guaranteed. Besides, we can also set a relatively large buffer occupancy threshold $q_{\min}$ to maintain high buffer occupancy so as to ensure continuous video playback. We evaluate the performances of all schemes with buffer sizes of 30 s, 60 s, and 90 s, but due to the space limit, we only show the results with a 30 s buffer size, and put the complete results in the supplementary material.

For performance comparison, besides our *m*DASH method, we also implement the famous DASH clients, i.e., Netflix [15], the latest SDP-based (named *sdp*DASH) [11] and typical buffer-based (named *buf*DASH) [6], [7] scheme. In Netflix, the video rate is selected mainly based on the bandwidth. In *sdp*DASH, the video rate is selected by the Markov decision. In order to mitigate the influence of bandwidth estimation errors in video rate selection, a map between the buffer occupancy and video rate is defined [6], [7], and the video rate is switched according to the map and buffer occupancy in *buf*DASH. Besides, to evaluate the performance of our proposed greedy algorithm, the results of the optimal solution (namely Optimal), i.e., set the search depth as the number of fragments, are also shown.

B. Impact of Short-Term Bandwidth Variations

Here we summarize the results in the case that the available bandwidth goes through some positive or negative spikes that last for 3 s–6 s. As shown in Fig. 3, *avail_bw* denotes the available bandwidth controlled by Dummynet, and fragment throughput refers to the download throughput for a particular

$$\mathcal{R}^{\psi}\left(u_k\right) = \begin{cases} \sum_{u_{k+1}} \mathcal{P}\left(u_{k+1} \left| u_k, a_k\left(u_k\right)\right.\right) r_k, \\ \qquad \text{if } q\left(t^s_{k+1}\right) < q_{\text{low}} \text{ or } q\left(t^s_{k+1}\right) > q_{\text{high}} \\ \sum\limits_{u_{k+1}} \mathcal{P}\left(u_{k+1} \left| u_k, a_k\left(u_k\right)\right.\right) \left(r_k + \sum_{u_{k+2}} \mathcal{P}\left(u_{k+2} \left| u_{k+1}, a_{k+1}\left(u_{k+1}\right)\right.\right) r_{k+1}\right), \\ \qquad \text{if } q_{\text{low}} \leq q\left(t^s_{k+1}\right) \leq q_{\text{high}} \end{cases} \tag{29}$$

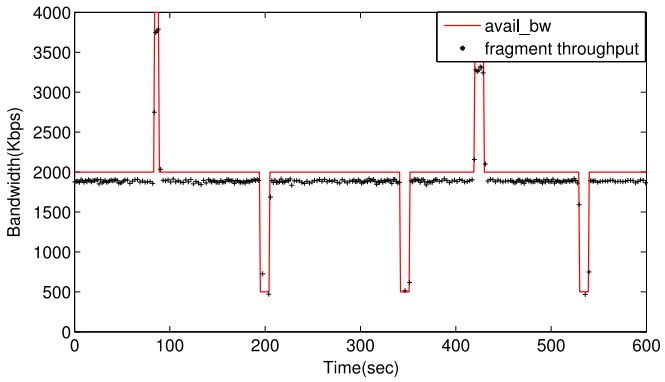


Fig. 3. Test pattern with short-term bandwidth variations.

fragment, which tracks the available bandwidth *avail-bw* quite well. Since such short-term variations are common in practice, an effective rate adaption scheme should be able to well compensate for such spikes using its buffered video, without causing short-term rate switchings.

We evaluate the performances of all the schemes under the short-term bandwidth variations as shown in Fig. 4. Fig. 4(a) shows that in Netflix, when there are positive or negative bandwidth spikes, the video rate is switched immediately. This is because the video rate in Netflix is selected solely based on the bandwidth with some predefined threshold. Besides, the selected video bitrate is never allowed to be higher than the available bandwidth, therefore, its buffer occupancy usually stays at a very high level (about 25 s, the maximal buffer size is 30 s). In contrast, although the average bandwidth remains stable as shown in Fig. 3, the video rate with *buf*DASH is periodically switched up and down to avoid buffer overflow and underflow, making the buffer occupancy fluctuates periodically. This is because its video rate is solely a function of buffer occupancy. Though buffer overflow and underflow are avoided, such frequent video rate fluctuations usually deteriorate QoE significantly. In *sdp*DASH, the video rate fluctuates heavily while it stabilizes the buffer occupancy. This is because the cost function with *sdp*DASH is mainly focused on stabilizing the buffer occupancy, whereas the smoothness of video rate is not well considered. Thus, its buffer occupancy keeps stable to guarantee continuous video playback while sacrificing the smoothness of video rate. Besides, since the video rate with *buf*DASH is only dependent on the current buffer occupancy, when a negative spike occurs, the buffer occupancy decreases rapidly, leading to an unnecessary rate switching down, and vice versa when there is a positive spike.

Different from these schemes, with *m*DASH, the rate is switched between 1.5 Mbps and 2.5 Mbps. This is because *m*DASH selects the video rate based on Markov decision, which is further dependent on the system state and reward function, where a video rate higher than the available bandwidth is allowed, thus achieving better bandwidth utilization. Compared to Netflix, *sdp*DASH, and *buf*DASH, besides continuous video playback, our method takes into account the smoothness of video rate in estimating the system state and reward. We can observe that the video playback rate keeps unchanged for at least 80 s (about 40 fragments), making rate switching infrequent and thereby avoiding significant degradation on QoE. Moreover, Fig. 4(d) shows that the lowest buffer occupancy is about 6 s (three fragments) which is sufficient to guarantee continuous video playback. Furthermore, the performance of the optimal algorithm is close to that of *m*DASH. The reasons why *m*DASH can achieve comparable performance with the optimal algorithm are threefold. First, the effect of future actions on the decision becomes smaller and smaller due to the discount parameter λ. Besides, since the available video bitrate is discrete, the same video bitrate may be selected even if the action costs are different. Finally, the most important reason is that *m*DASH dynamically adapts the reward function to three different scenarios of buffer occupancy. Under a certain scenario, the effect of some factors on the cost function may be negligible even if more future actions are considered. For example, when the buffer occupancy is low, the video rate switching decision is mainly decided by avoiding playback freeze.

Furthermore, in Table I, we compare the performance of various methods in terms of several quality metrics, including the average video bitrate, bandwidth utilization, average buffer occupancy, maximal buffer occupancy, minimal buffer occupancy, and instability [32]. The instability metric is used to measure the smoothness of video rate, where a small value indicates high smoothness. From Table I we can observe that Netflix leads to the lowest average video bitrate and bandwidth utilization, while the other schemes achieve very close performance in average video bitrate and bandwidth utilization. This is mainly because the Netflix method is too conservative to select a video bitrate higher than the available bandwidth. As a result, the buffer occupancy usually stays at a relatively high level, making the bandwidth utilization low. In contrast, with the other schemes, the buffer occupancy fluctuates between empty and the full buffer size (30 s) to ensure continuous video playback. At last, we also compare the instability performance. The results in Table I shows that *m*DASH achieves the smallest value of instability (i.e., the smoothest video bitrate). This is also consistent with the results in Fig. 4.

C. Impact of Long-Term Bandwidth Variations

We then evaluate the impact of long-term bandwidth variations on QoE under the scenario depicted in Fig. 5. As shown in Fig. 6, in the intervals where the available bandwidth keeps unchanged, all the schemes perform similarly to that in Fig. 4. When the bandwidth changes, all the schemes switch the requested video rate to avoid buffer overflow/underflow as demonstrated in the buffer occupancy results in Fig. 6.

Again, the video bitrate with Netflix is switched immediately when the bandwidth changes, and smooth video bitrate is obtained under the long-term bandwidth variations depicted in Fig. 6(a). However, such bandwidth-based scheme tends to select video bitrates lower than the available bandwidth, which leads to low bandwidth utilization as shown in Table II. In contrast, with both *buf*DASH and *sdp*DASH, similar to that in Fig. 4, the smoothness of video bitrate is sacrificed in order to stabilize the buffer occupancy. Compared to *buf*DASH

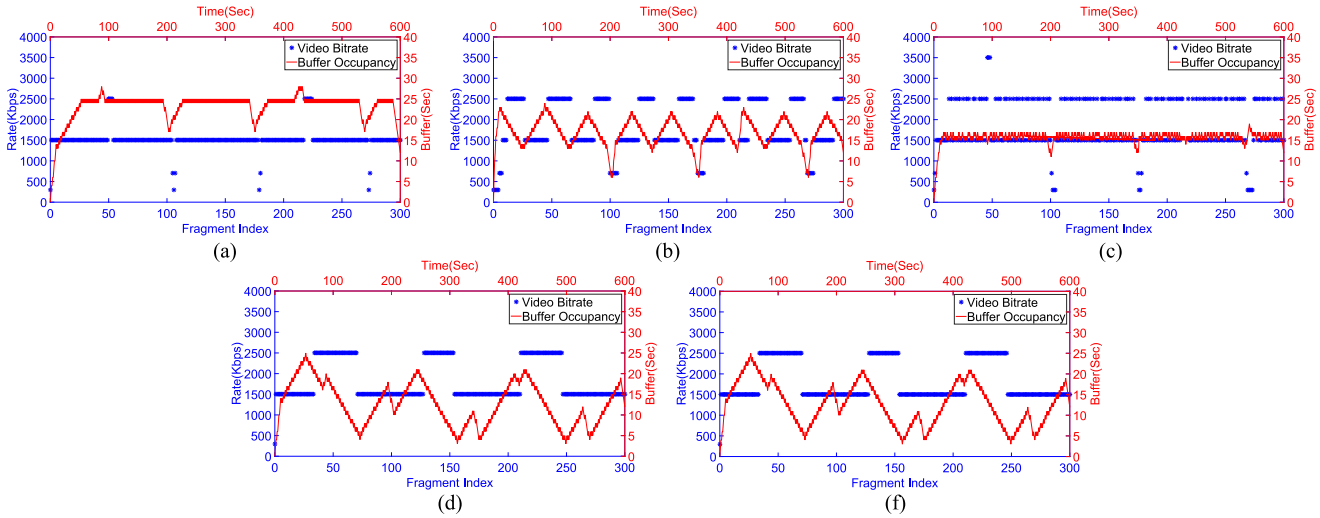


Fig. 4. Video bitrate adaptation under short-term bandwidth variations: video bitrate and buffer size evolution results. (a) Netflix. (b) *buf*DASH. (c) *sdp*DASH. (d) *m*DASH. (e) Optimal.

TABLE I
PERFORMANCE COMPARISON UNDER SHORT-TERM BANDWIDTH VARIATIONS

Scheme	Average video bitrate(Mbps)	Bandwidth utilization(%)	Average buffer occupancy(sec)	Maximum buffer occupancy(sec)	Minimum buffer occupancy(sec)	Instability
Netflix	1510	81.4	23.2	28	12	0.027
*buf*DASH	1831	98.7	14.9	24	6	0.046
*sdp*DASH	1832	98.8	15.7	19	11	0.370
*m*DASH	1826	98.6	16.9	24	6	0.010
Optimal	1828	98.7	16.7	24	6	0.010

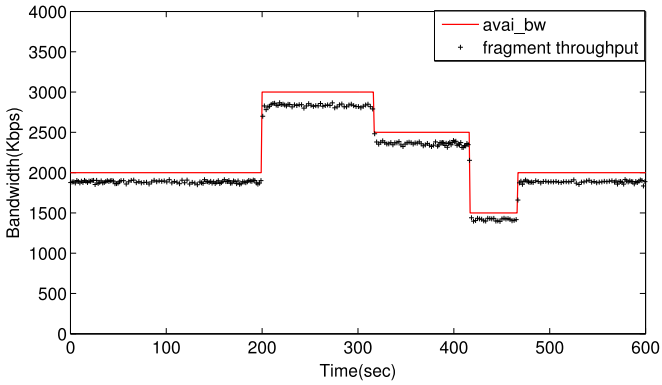


Fig. 5. Test pattern with long-term bandwidth variations.

and *sdp*DASH, *m*DASH achieves a better tradeoff between the smoothness of video rate and bandwidth utilization as illustrated in Fig. 6. Based on the reward function used in *m*DASH, when the buffer occupancy is too high or too low, rate switching is performed to avoid buffer overflow or underflow. Otherwise, it tends to maintain the smoothness of video rate. From Fig. 6(d) we can observe that when the video rate is smaller than the available bandwidth, the buffer occupancy increases without causing video rate switchings to maintain the smoothness of video rate. When the number of consecutive fragments requested at the same video rate increases to a certain threshold, rate switching is preferred to rate smoothing. But the switching amplitude needs to be constrained so that the rate will not increase sharply (e.g., see fragment 110 at 220 s). At last, when the buffer occupancy increases to q_{high}, avoiding buffer overflow becomes the most important, so video rate is further switched up. Conversely, when the bandwidth goes low, the video rate is switched from 3.5 Mbps to 2.5 Mbps, and is then further switched down to 1.5 Mbps to avoid buffer underflow. The proposed switching scheme is advantageous because, on one hand, it avoids buffer overflow and underflow, and, On the other hand, it can also avoid frequent video rate fluctuations and maintain the smoothness of video rate. Furthermore, sharp rate switching up/down can also be avoided which is useful in improving the QoE.

The comparison of various quality metrics with long-term bandwidth variations are summarized in Table II. The buffer occupancy results indicate that all the schemes effectively mitigate buffer overflow/underflow. Besides, Netflix again leads to the lowest average video bitrate and bandwidth utilization due to its conservative rate adaptation scheme. On the other hand, it achieves smoother video bitrate than *buf*DASH and *sdp*DASH as measured by the instability metric. Overall, *m*DASH achieves comparable performance with the optimal scheme as evidenced in Fig. 6(d) and Table II, while achieving much lower complexity than the optimal scheme which takes about 5 min on average to make a decision. This also demonstrates the effectiveness of our proposed rate adaptation approach.

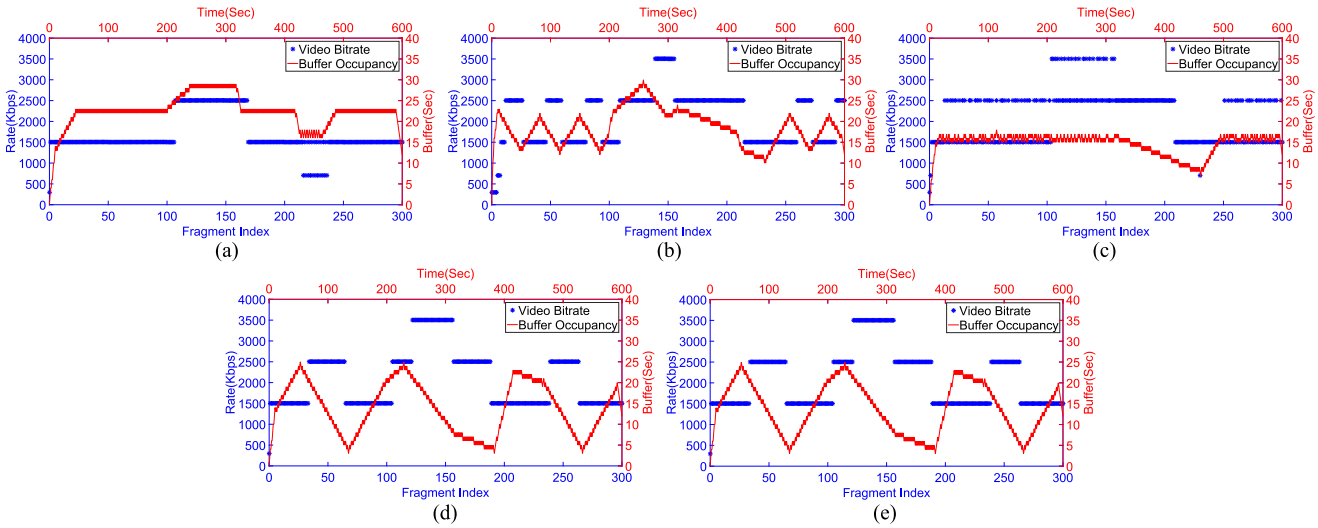


Fig. 6. Video bitrate adaptation under long-term bandwidth variations: video bitrate and buffer size evolution results. (a) Netflix. (b) *buf*DASH. (c) *sdp*DASH. (d) *m*DASH. (e) Optimal.

TABLE II
PERFORMANCE COMPARISON UNDER LONG-TERM BANDWIDTH VARIATIONS

Scheme	Average video bitrate(Mbps)	Bandwidth utilization(%)	Average buffer occupancy(sec)	Maximum buffer occupancy(sec)	Minimum buffer occupancy(sec)	Instability
Netflix	1464	69.3	23.1	25	12	0.053
*buf*DASH	2085	98.8	18.6	30	10	0.023
*sdp*DASH	2087	98.9	14.6	18	7	0.239
*m*DASH	2079	98.5	13.9	25	3	0.012
Optimal	2087	98.6	13.4	25	3	0.012

D. Internet Experiments

We then test our proposed scheme in the Internet. We set up one Planetlab node as the DASH server which is located in Hong Kong, China (*plab1.cs.ust.hk*) and another Planetlab node as the DASH client in Beijing, China (*p11.pku.edu.cn*). We do not inject any background traffic between the server and client.

We conduct different experiments sequentially, in which the bandwidth patterns are not controllable. The real bandwidth traces for all compared schemes are illustrated in Fig. 7. For the optimal scheme, since it is too time consuming to be performed on-line, we implement it off-line using the bandwidth traces collected from the results of *m*DASH. In all experiments, we can observe long-term shift and short-term fluctuations of bandwidth along the Internet transmission path. The video bitrate (blue curves) and buffer size (red curves) evolution results of various schemes are compared in Fig. 8. The results demonstrate that our rate adaptation algorithm can well adapt to the varying network conditions, which is consistent with our testbed-based results. The results show that *m*DASH can effectively absorb short-term spikes using buffered video, without causing short-term rate switching. While for long-term bandwidth changes, it switches to an appropriate rate without causing buffer overflow/underflow or playback interruptions. In contrast, Netflix implements a conservative rate adaptation scheme that tends to maintain a high buffer occupancy. As for *buf*DASH, both the buffer occupancy and the video bitrate fluctuate frequently since it solely relies on switching the video bitrate to avoid buffer overflow/underflow. Compared to *m*DASH, although *sdp*DASH, can better stabilize buffer occupancy, it also leads to frequent unnecessary bitrate switching since it does not maintain the smoothness of video bitrate well. The buffer occupancy with *buf*DASH and *m*DASH fluctuates within a certain range, whereas *m*DASH more effectively suppress short-term rate switching compared to *sdp*DASH. This is mainly because *m*DASH performs rate selection by jointly considering the buffer occupancy, video rate switching frequency and amplitude, buffer overflow/underflow, and video playback quality. Note, the results show that sacrificing the smoothness of video rate to stabilize the buffer occupancy usually cannot do a good job in maintaining QoE since from the end user point of view, visual quality degradation due to rate fluctuations are much more perceivable than that due to buffer occupancy oscillations as long as no playback freeze happens.

The comparison of quality metrics shown in Table III shows that, similar to results in Tables I and II, all the schemes lead to very close average bandwidth utilization except Netflix. Besides, all schemes can maintain that continuous playback well. Again, *m*DASH achieves comparable performances with the optimal scheme.

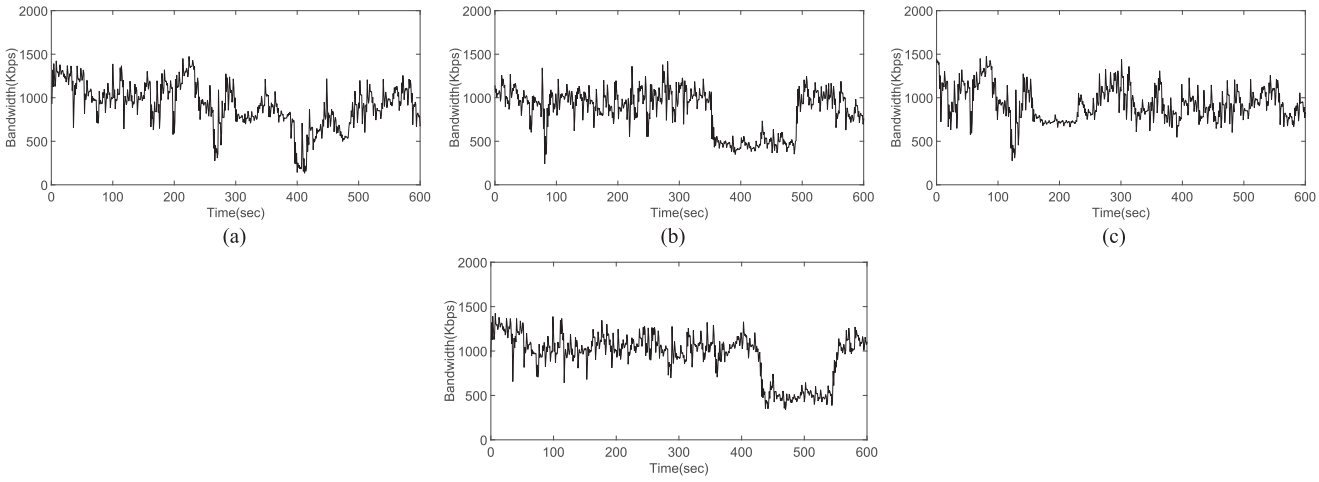


Fig. 7. Internet (Planetlab) traces with real bandwidth variations. (a) Netflix. (b) *buf*DASH. (c) *sdp*DASH. (d) *m*DASH and optimal.

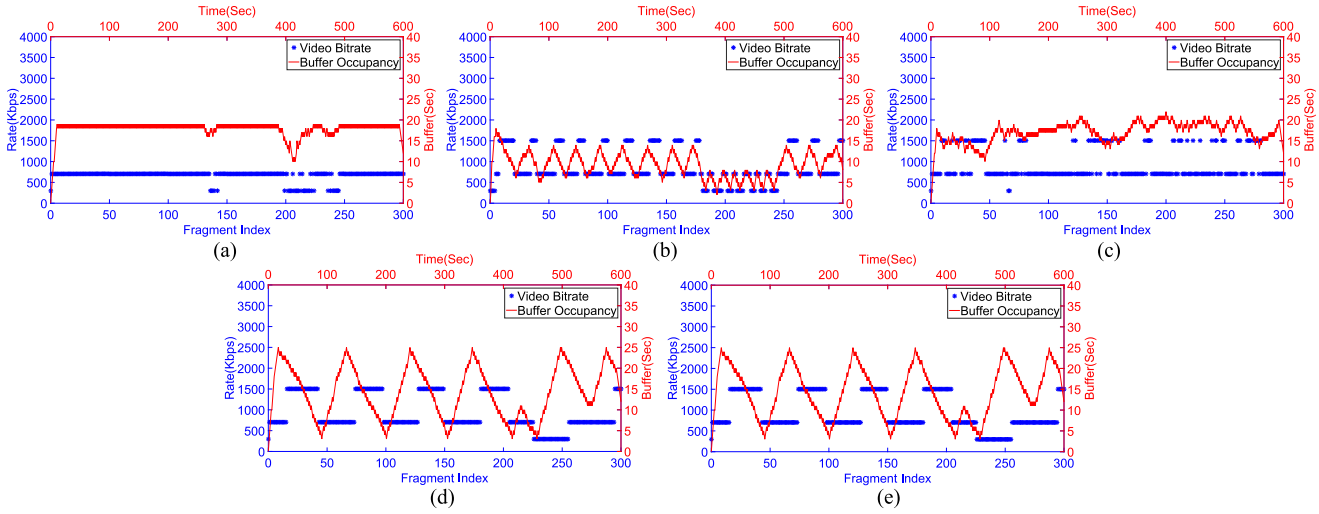


Fig. 8. Video bitrate adaptation over the Internet (Planetlab): video bitrate and buffer size evolution results. (a) Netflix. (b) *buf*DASH. (c) *sdp*DASH. (d) *m*DASH. (e) Optimal.

TABLE III
PERFORMANCE COMPARISON OF RATE ADAPTATION OF DASH IN THE INTERNET (PLANETLAB)

Scheme	Average video bitrate(Mbps)	Bandwidth utilization(%)	Average buffer occupancy(sec)	Maximum buffer occupancy(sec)	Minimum buffer occupancy(sec)	Instability
Netflix	656	71.3	17.8	19	10	0.069
*buf*DASH	854	98.9	9.2	18	2	0.100
*sdp*DASH	914	98.3	17	22	10	0.185
*m*DASH	930	98.4	14.3	25	3	0.028
Optimal	941	98.5	14.2	25	3	0.027

Furthermore, to make a fair comparison, we also used the same bandwidth trace collected from the Planetlab experiment to evaluate all methods off-line. The bandwidth trace used for the evaluation is shown in Fig. 8(d), and the bitrate and buffer results for all methods are shown in Fig. 9. The results clearly show that *m*DASH achieves much smoother video bitrate (i.e., much fewer short-term bitrate switching) compared with the other methods. Besides, the buffer occupancy result demonstrates that no playback freeze happens with *m*DASH. Furthermore, we also compare several quality metrics for the five methods as summarized in Table IV, which all consistently show that *m*DASH achieves almost the same performance as the Optimal scheme and outperforms the other schemes in terms of all metrics including the smoothness of video bitrate

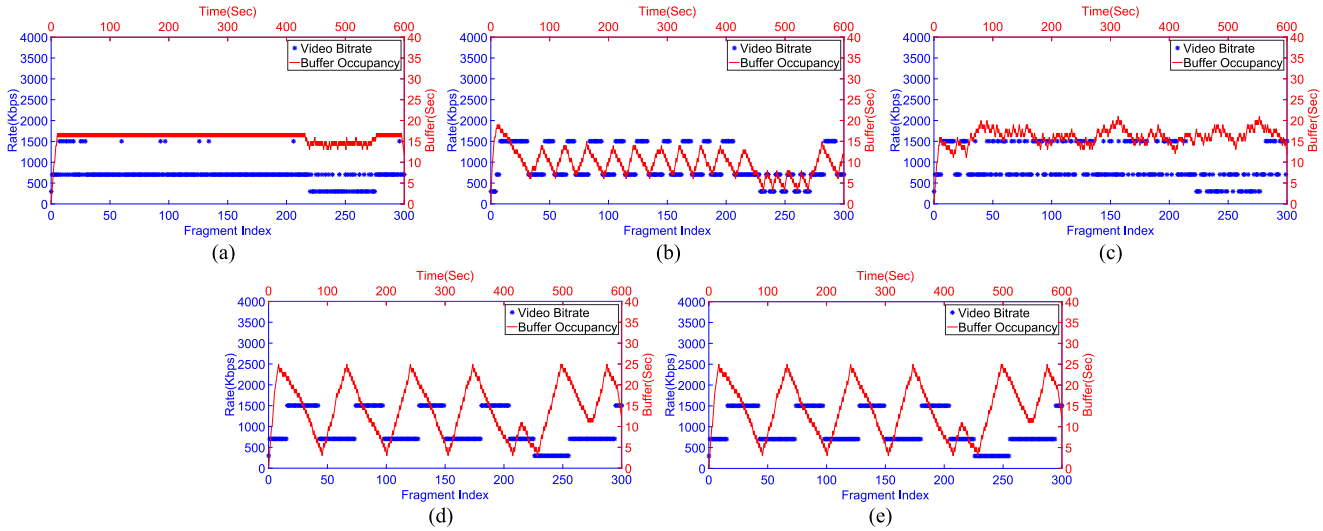


Fig. 9. Video bitrate adaptation under the same Internet (Planetlab) trace: video bitrate and buffer size evolution results under the same bandwidth trace. (a) Netflix. (b) *buf*DASH. (c) *sdp*DASH. (d) *m*DASH. (e) Optimal.

TABLE IV
PERFORMANCE COMPARISON UNDER THE SAME INTERNET (PLANETLAB) BANDWIDTH TRACE

Scheme	Average video bitrate(Kbps)	Bandwidth utilization(%)	Average buffer occupancy(sec)	Maximum buffer occupancy(sec)	Minimum buffer occupancy(sec)	Instability
Netflix	692	72.6	15.9	17	12	0.209
*buf*DASH	945	99.1	9.83	19	3	0.084
*sdp*DASH	940	98.6	16.1	21	11	0.299
*m*DASH	947	99.2	14.3	25	3	0.028
Optimal	951	99.4	14.2	26	3	0.028

TABLE V
DETAILED INFORMATION ABOUT THE TEST SEQUENCES

	Content	Version	Length	Resolution	Average bitrate (Kbps)	PSNR (dB)
Big	Movie with	1	9m56s	720 × 480	286	31.7
Buck	fast and	1	9m56s	720 × 480	693	33.2
Bunny	slow motion	3	9m56s	1920 × 1080	1482	36.1
		4	9m56s	1920 × 1080	2469	38.6
		5	9m56s	1920 × 1080	3451	39.7
Tears	Movie with	1	12m14s	720 × 480	291	30.8
of	fast and	2	12m14s	720 × 480	678	32.1
Steel	slow motion	3	12m14s	1920 × 800	1479	35.1
		4	12m14s	1920 × 800	2589	36.9
		5	12m14s	1920 × 800	3421	37.8
Sintel	Movie with	1	14m48s	720 × 480	276	30.1
	fast and	1	14m48s	720 × 480	657	31.9
	slow motion	3	14m48s	1920 × 818	1446	33.7
		4	14m48s	1920 × 818	2456	35.6
		5	14m48s	1920 × 818	3411	36.9

(i.e., smaller instability), average video rate and bandwidth utilization.

E. Subjective Visual Quality Evaluation

At last, we conduct subjective tests in order to validate the performance of the proposed algorithm compared to the others. We use three video sequences {Big Buck Bunny, Tears of Steel, and Sintel} in the tests, which are all encoded into five versions with a rate of {300 Kbps, 700 Kbps, 1.5 Mbps, 2.5 Mbps, 3.5 Mbps}. Each version is further divided into equal-length video fragments with a length of 2 s. The details of the test sequences are listed in Table V. For fair comparison, all the methods are tested against the same bandwidth traces collected from the Planetlab. Then, for each rate adaptation scheme, according to the video bitrate results, we combine the video fragments into

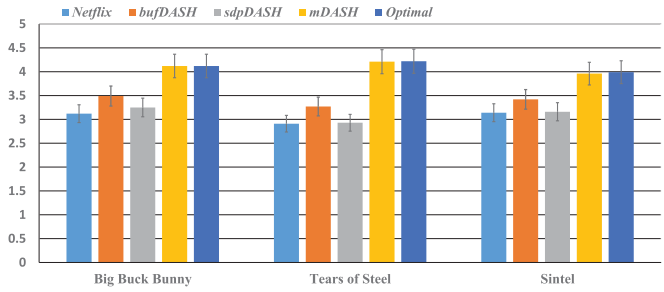


Fig. 10. Subjective visual quality comparison in terms of average MOS scorers for different methods.

a single video file. In our experiments, we invited 32 subjects to participate in the subjective tests, all having normal visual acuity, and color vision. None of the subjects had knowledge about the algorithm implementations. Since there were 15 sequences to be evaluated, asking subjects to evaluate all test sequence would give them too much workload to keep their concentration on the evaluation. To avoid the problem, each subject was asked to evaluate nine sequences, which are randomly chosen from the 15 test sequences. Therefore, each test sequence was thus evaluated by 19 subjects, which is enough to ensure the results not be biased by a few subjects. The subjective evaluations were carried out in a lab with controlled ambient light. The displays used for presenting the sequences were 22 inches, with a resolution of 1920×1080 and an aspect ratio of 16:9. The subjects are asked to give a score immediately after watching a sequence. The score ranges from 0 to 5 (a continuous scale), where 0 indicates very poor quality of experience and 5 indicates perfect quality of experience. The scores are given based on the comprehensive feeling of the subjects themselves without any suggestions.

For each test sequence, the highest and lowest scores are removed and the MOS scores of all schemes are summarized in Fig. 10. The results show that, consistent with the objective test results, *m*DASH and the *Optimal* scheme receive near the same MOS, which is much higher than the others. In contrast, Netflix and *sdp*DASH receive the worst subjective scores. The performance loss of Netflix mainly comes from its low bandwidth utilization that leads to a much lower average video bitrate, whereas the poor performance of *sdp*DASH is mainly due to its frequent bitrate switching. At last, the MOS of *sdp*DASH is slightly higher than that of *buf*DASH because both *sdp*DASH and *buf*DASH achieve high and comparable bandwidth utilization, but the video rate with *buf*DASH is smoother than that with *sdp*DASH. Compared with the existing schemes, our proposed *m*DASH jointly considers the bandwidth utilization, buffer state and video bitrate's smoothness, thereby leading to superior subjective user experience as justified by its high MOS rating.

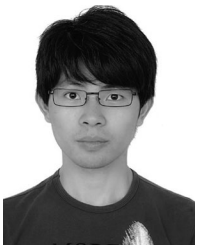
VIII. Conclusion

In this paper, we proposed a Markov decision based rate adaption approach for dynamic HTTP streaming. We proposed to take into account several system parameters that have critical impact on visual quality, including video playback quality, video rate switching frequency and amplitude, buffer overflow/underflow, and buffer occupancy. By carefully analyzing the probabilistic characteristics of the DASH system, we have proposed effective reward functions in terms of the considered system parameters for rate switching decision, so as to maximize the quality of user experience. Moreover, to reduce the computational complexity, we have also proposed a low-complexity sub-optimal greedy algorithm to support online real-time video streaming. Our experiment results in both the network test-bed and the real-world Internet all demonstrate that our proposed method achieves significantly higher video quality while maintaining smoother video rate compared to existing methods. In addition, the subjective quality tests further demonstrate the effectiveness of our proposed method in terms of the subjective user experience.

References

[1] A. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 54–63, Mar. 2011.
[2] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proc. ACM Multimedia Syst.*, Feb. 2011, pp. 133–144.
[3] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. ACM Multimedia Syst.*, Feb. 2011, pp. 169–174.
[4] B. Zhou, J. Wang, Z. Zou, and J. Wen, "Bandwidth estimation and rate adaptation in HTTP streaming," in *Proc. IEEE Int. Conf. Comput. Netw. Commun.*, Feb. 2012, pp. 734–738.
[5] L. D. Cicco, S. Mascolo, and V. Palmisano, "Feedback control for adaptive live video streaming," in *Proc. ACM Multimedia Syst.*, Feb. 2011, pp. 145–156.
[6] T.-Y. Huang, R. Johari, and N. McKeown, "Downton abbey without the hiccups: Buffer-based rate adaptation for http video streaming," in *Proc. ACM SIGCOMM Workshop Future Human-Centric Multimedia Netw.*, Aug. 2013, pp. 9–11.
[7] 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 SIGCOMM*, 2014, pp. 187–198.
[8] N. Cranley, P. Perry, and L. Murphy, "User perception of adaption video quality," *Int. J. Human-Comput. Studies*, vol. 64, no. 8, pp. 637–647, 2006.
[9] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 795–805, Apr. 2014.
[10] M. Xing, S. Xiang, and L. Cai, "Rate adaptation strategy for video streaming over multiple wireless access networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2012, pp. 5745–5750.
[11] S. Garcia, J. Cabrera, and N. Garcia, "Quality-control algorithm for adaptive streaming services over wireless channels," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 1, pp. 50–59, Feb. 2015.
[12] T. Andelin, V. Chetty, D. Harbaughs, S. Warnick, and D. Zappala, "Quality selection for dynamic adaptive streaming over HTTP with scalable video coding," in *Proc. ACM Multimedia Syst.*, Feb. 2012, pp. 149–154.
[13] 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.
[14] C. Zhou and C.-W. Lin, "A Markov decision based rate adaption approach for dynamic HTTP streaming," in *Proc. IEEE Vis. Commun. Image Process*, Dec. 2015, pp. 1–6.
[15] M. Watson, "HTTP adaptive streaming in practice," presented at the ACM Multimedia Syst. Conf., San Jose, CA, USA, 2011.
[16] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. ACM Multimedia Syst.*, Feb. 2011, pp. 169–174.
[17] A. Zambelli, "MS-SSTR: Microsoft Smooth Streaming protocol technical report," Microsoft Corp., Redmond, WA, USA, Jun. 2010. [Online]. Available: http://www.iis.net/downloads/microsoft/smooth-streaming.
[18] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware DASH system," in *Proc. ACM Multimedia Syst.*, Feb. 2012, pp. 11–22.

[19] *Definition of Quality of Experience*, CoM12-LS 62-E, TD 109rev2 (PLEN/12) ITU-T SG12, Geneva, Switzerland, Jan. 2007.
[20] T. Hoßfeld *et al.*, "Quantification of YouTube QoE via crowdsourcing," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2011, pp. 494–499.
[21] 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.
[22] S. Tavakoli, J. Gutierrez, and N. Garcia, "Subjective quality study of adaptive streaming of monoscopic and stereoscopic video," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 684–692, Apr. 2014.
[23] Y. Liu *et al.*, "A study on quality of experience for adaptive streaming service," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2003, pp. 682–686.
[24] D. Suh, I. Jang, and S. Pack, "QoE-enhanced adaptation algorithm over DASH for multimedia streaming," in *Proc. IEEE Int. Conf. Inf. Netw.*, Feb. 2014, pp. 497–501.
[25] S. Xiang, L. Cai, and J. Pan, "Adaptive scalable video streaming in wireless networks," in *Proc. ACM Multimedia Syst.*, Feb. 2012, pp. 167–172.
[26] G. Tian and Y. Liu, "Towards agile and smooth video adaption in dynamic HTTP streaming," in *Proc. ACM CoNEXT*, Dec. 2012, pp. 109–120.
[27] T. Andelin, V. Chetty, D. Harbaugh, S. Warnick, and D. Zappala, "Quality selection for dynamic adaptive streaming over HTTP with scalable video coding," in *Proc. ACM Multimedia Syst.*, Feb. 2012, pp. 149–154.
[28] M. Xing, S. Xiang, and L. Cai, "Rate adaptation strategy for video streaming over multiple wireless access networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2012, pp. 5745–5750.
[29] S. Tavakoli *et al.*, "Subjective quality assessment of an adaptive video streaming model," in *Proc. SPIE*, vol. 9016: Image Quality Syst. Perform. XI, Feb. 2014, pp. 90160K-1–90160K-12.
[30] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1431–1445, Oct. 2013.
[31] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.
[32] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," *IEEE/ACM Trans. Netw.*, vol. 22, pp. 326–340, Feb. 2014.



Chao Zhou received the Ph.D. degree from the Institute of Computer Science & Technology, Peking University, Beijing, China, in 2014.

He has been with PlanWin Information Technology as a Senior Researcher since August 2015. From July 2014 to August 2015, he was a Senior Research Engineer with the Media Technology Lab, CRI, Huawei Technologies Co. Ltd., Beijing, China. His research interests include HTTP video streaming, joint source-channel coding, and multimedia communications and processing.

Dr. Zhou was the recipient of the Best Paper Award presented at the IEEE VCIP 2015, and the Best Student Paper Award at the IEEE VCIP 2012.

Chia-Wen Lin (S'95–M'00–SM'04) received the Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, R.O.C., in 2000.

He is currently a Professor with the Department of Electrical Engineering and the Institute of Communications Engineering, NTHU. He was with the Department of Computer Science and Information Engineering, National Chung Cheng University, Minhsiung, Taiwan, R.O.C., from 2000 to 2007. Prior to joining academia, he worked for the Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan, R.O.C., from 1992 to 2000. His research interests include image and video processing and video networking.

Dr. Lin has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the *IEEE MultiMedia Magazine*, and the *Journal of Visual Communication and Image Representation*. He was a Steering Committee Member of the IEEE TRANSACTIONS ON MULTIMEDIA from 2014 to 2015. He was Chair of the Multimedia Systems and Applications Technical Committee of the IEEE Circuits and Systems Society from 2013 to 2015. He served as Technical Program Co-Chair of the IEEE International Conference on Multimedia & Expo (ICME) in 2010, and Special Session Co-Chair of the IEEE ICME in 2009. He was the recipient of the Best Paper Award of IEEE VCIP 2015, the Top 10% Paper Award of IEEE MMSP 2013, and the Young Investigator Award of VCIP 2005. He was the recipient of the Young Investigator Award presented by the Ministry of Science and Technology, Taiwan, in 2006.

Zongming Guo received the B.Sc. degree in mathematics and the M.Sc. and Ph.D. degrees in computer science from Peking University, Beijing, China, in 1987, 1990, and 1994, respectively.

He is currently the Dean of the Institute of Computer Science & Technology, Peking University. He has authored or coauthored over 80 technical articles in refereed journals and conference proceedings in the areas of multimedia, image and video compression, image and video retrieval, and watermarking. His current research interests include streaming media technology, IPTV and mobile multimedia, and image and video processing.

Dr. Guo led the research and developing team of the Institute of Computer Science & Technology, Peking University, which was the recipient of the first prize of The State Administration of Radio Film and Television, the first prize of Ministry of Education Science and Technology Progress Award, and the second prize of the National Science and Technology Award in 2004, 2006, and 2007, respectively. He was the recipient of the Government Allowance granted by the State Council in 2009.