Non-Cooperative Game Theory Based Rate Adaptation for Dynamic Video Streaming over HTTP

Hui Yuan, *Senior Member, IEEE*, Huayong Fu, Ju Liu, *Senior Member, IEEE*, Junhui Hou, *Member, IEEE*, and Sam Kwong, *Fellow, IEEE*

Abstract—Dynamic Adaptive Streaming over HTTP (DASH) has demonstrated to be an emerging and promising multimedia streaming technique, owing to its capability of dealing with the variability of networks. Rate adaptation mechanism, a challenging and open issue, plays an important role in DASH based systems since it affects Quality of Experience (*QoE*) of users, network utilization, etc. In this paper, based on non-cooperative game theory, we propose a novel algorithm to optimally allocate the limited export bandwidth of the server to multi-users to maximize their *QoE* with fairness guaranteed. The proposed algorithm is proxy-free. Specifically, a novel user *QoE* model is derived by taking a variety of factors into account, like the received video quality, the reference buffer length, and user accumulated buffer lengths, etc. Then, the bandwidth competing problem is formulated as a non-cooperation game with the existence of Nash Equilibrium that is theoretically proven. Finally, a distributed iterative algorithm with stability analysis is proposed to find the Nash Equilibrium. Compared with state-of-the-art methods, extensive experimental results in terms of both simulated and realistic networking scenarios demonstrate that the proposed algorithm can produce higher *QoE*, and the actual buffer lengths of all users keep nearly optimal states, i.e., moving around the reference buffer all the time. Besides, the proposed algorithm produces no playback interruption.

Index Terms—Non-cooperative game, nash equilibrium, DASH, bitrate adaptation, QoE

1 Introduction

Nowadays, with the increase of Internet bandwidth and the tremendous growth of web platforms, Hypertext Transfer Protocol (HTTP) streaming has become a cost-effective method for multimedia delivery [1], [2]. Dynamic Adaptive Streaming over HTTP (DASH) is a typical HTTP based multimedia delivery standard that can transmit multimedia content adaptively between multimedia servers and users with a limited and varied network bandwidth [3]. Fig. 1 illustrates a typical DASH-based video delivery system. In this system, the media content (e.g., videos or audios) is first divided into multiple segments (or chunks) [4] with the same display time. Each segment is then encoded/transcoded with different bitrates (corresponding to different quality levels). At the same time, the server generates a Media Presentation Description (MPD) file that records the information of the available video content, e.g., URL addresses, segment lengths, quality levels, resolutions, etc. The users first download the MPD file from the server using HTTP protocol, and then request segments with different quality levels to adapt to the bandwidth variation. The main advantage of DASH is that it can achieve bandwidth adaptation and reduce the number of playback interruptions under fluctuating network conditions [5]. An effective rate adaptation algorithm is necessary in a DASH system, with which the DASH user can adaptively request video segments with different bitrates based on the network condition and its buffer length. However, this challenging issue is not specified in the DASH standard. Without an effective rate adaptation algorithm, the DASH user might suffer from frequent interruptions. Moreover, recent studies show that the DASH user's selfish behavior (i.e., making requests without considering other users sharing the network resources) will result in network underutilization (or congestion), fluctuating and unfair throughout allocation [6]. This paper aims to develop an effective rate adaptation algorithm to address the above issues.

Many rate adaptation methods have been proposed (see Section 2). However, most of them optimize the HTTP streaming of multiple DASH users sharing the same network resources *separately*, regardless of the influence between each other; thus, user fairness cannot be well guaranteed. *In contrast, the proposed rate adaptation algorithm optimizes the HTTP streaming of multiple DASH users simultaneously who compete for higher quality video segments from a single server with a limited export bandwidth.*

As a branch of game theory, the non-cooperative game theory [7] can resolve the conflicts among interacting players

- H. Yuan, H. Fu, and J. Liu are with the School of Information Science and Engineering, Shandong University, Ji'nan 250100, China. E-mail: {huiyuan, juliu}@sdu.edu.cn, johy.fu@gmail.com.
- J. Hou and S. Kwong are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: {jh.hou, cssamk}@cityu.edu.hk.

Manuscript received 24 May 2017; revised 11 Dec. 2017; accepted 28 Jan. 2018. Date of publication 1 Feb. 2018; date of current version 29 Aug. 2018. (Corresponding author: Hui Yuan.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2800749

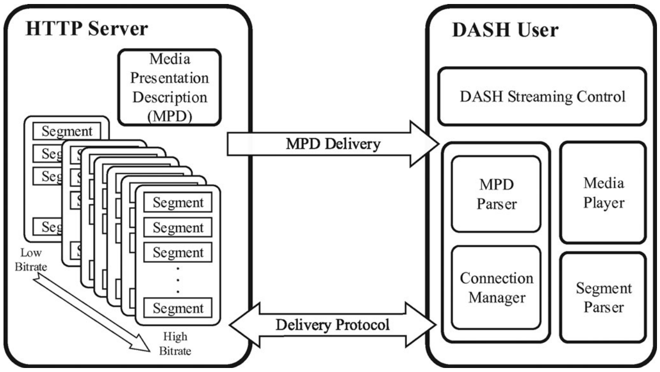


Fig. 1. DASH system architecture.

involved in a certain game, in which each player behaves selfishly to optimize its own profit usually quantified as an objective function. The non-cooperative game can provide meaningful solutions for many applications where the interaction among several players is negligible and centralized approaches are not suitable [8], [9]. The typical application of a non-cooperative game is the oligopoly market problem in economics, in which all corporations compete for market share of the same commodity in order to maximize their own profits [10], and the market share of each corporation tends to be stable and reaches Nash Equilibrium.

In this paper, we consider formulating the rate adaptation problem for improving user *QoE*, as well as preserving user fairness, as a non-cooperative game in which DASH users try to consume the limited export bandwidth of the server as much as possible to maximize their profits (i.e., *QoE*). The optimal bitrate that produces the optimal *QoE* can be obtained when the Nash Equilibrium of the contradiction problem is achieved. More specifically, the proposed non-cooperative game theory based rate adaptation algorithm determines the requested bitrate for a DASH user based on both the local information (e.g., the requested bitrate of the last segment, reference buffer length, and current buffer length) and the global payoff variation information obtained from the server. Each user can gradually adapt the requested bitrate to convergence. The bitrate's convergence speed is controlled by the learning rate. Note that there is a limitation that additional HTTP sessions between users and servers are needed in the proposed method, which may reduce the transmission efficiency of the streaming system. Even so, extensive experimental results demonstrate that the proposed algorithm can produce higher *QoE*, while the actual buffer lengths of all users move around the reference buffer all the time when compared with state-of-the-art algorithms. Moreover, the proposed algorithm is proxy-free and playback interruption-free. To the best of our knowledge, this is the first time to address the DASH rate adaptation problem using the non-cooperative game theory. The major contributions of this paper are summarized as follows.

- We formulated the rate adaptation problem as a non-cooperative game with the existence of the Nash Equilibrium that is theoretically proven. The proposed rate adaptation algorithm optimizes streaming for multiple DASH users simultaneously to guarantee their fairness and improve their *QoE*.
- We proposed a novel *QoE* model for the DASH users by taking the current buffer length, reference buffer length, and video quality into account.
- We designed an efficient distributed iterative algorithm to obtain the Nash Equilibrium of the game by additional HTTP sessions between the server and users, the stability of which is theoretically analyzed.

The rest of this paper is organized as follows. In Section 2, the related work on rate adaptation methods for DASH is presented. In Section 3, a user *QoE* model is proposed, and the corresponding non-cooperative game is formulated. Besides that, the existence of Nash Equilibrium and the stability of the non-cooperative game are also demonstrated. Simulation and realistic experimental results are given in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper and discusses the limitations of the proposed method.

2 Related Work

In order to adapt to the varying bandwidths, a straightforward method is to estimate the bandwidth or throughput of the transmission link. Thang et al. [11] proposed a channel throughput estimation-based adaptive request method to deal with short-term bandwidth fluctuations and stabilize the bitrates of segments. Romero [12] developed a Java client for HTTP streaming on the Android platform and proposed a smoothed throughput estimation method to cope with short-term fluctuations. But the user's buffer length (evaluated by remaining playback time) has not been considered. In [13], a round-trip time and previous values of the instant throughput based throughput estimation method is proposed for adaptive streaming so as to stabilize both the bitrates of segments and the buffer length of users. Liu et al. [14] developed a throughput estimation based rate adaptation method by using the ratio of the expected segment fetch time (ESFT) and the measured segment fetch time. However, in practical applications, since bandwidth and throughput are affected by a lot of factors, it is a non-trivial task to estimate them accurately. *Huang et al.* [15] showed that inaccurate throughput estimation at the user side can cause the degeneration of the video quality. Recently, Mao et al. [16] proposed a deep reinforcement learning based rate adaptation algorithm by accurately estimating the channel throughput accurately.

Besides, in order to improve the *QoE* [17], [18] of users directly, some researchers proposed dynamic bitrate selection methods based on *QoE* maximization [19], [20], [21], [22]. Zhang et al. [19] proposed a buffer management-based *QoE* model for HTTP adaptive bitrate streaming and formulated the adaptive request mechanisms as a constrained convex optimization problem which is then solved by the Lagrange multiplier method. Gheibi et al. [20] proposed a *QoE* metric by considering the probability of interruption in media playback and the number of initial buffered packets (initial waiting time) for streaming media applications. However, *QoE* is not only influenced by buffer length, but also by the requested bitrate and bitrate switching frequency, etc. Xu et al. [21] modeled the user *QoE* as a combination of bitrate, starvation probability of playback buffer, and continuous playback time, and they proposed two bitrate switching algorithms based on the channel variation and buffer length. Rodríguez et al. [22] proposed a non-reference *QoE* metric for DASH by taking initial buffer delay, temporal playback interruptions, and video

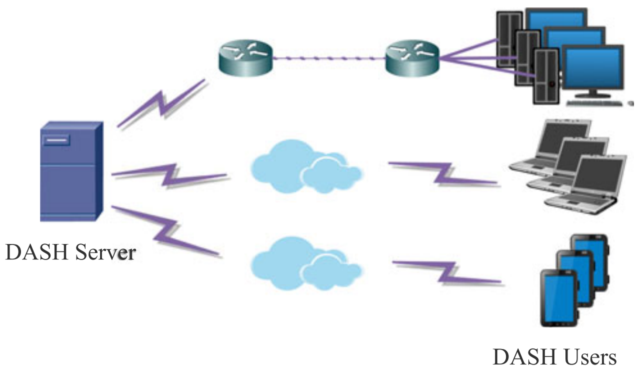


Fig. 2. Scenario for adaptive HTTP video delivery.

resolution changes into account. In addition, a Markov decision-based rate adaptation scheme for DASH aiming to maximize the user *QoE* under time-varying channel conditions was proposed by Zhou et al. [23] in which the video quality level, bitrate switching frequency and amplitude, buffer length, etc. are considered comprehensively. Similarly, Martín et al. [24] proposed a *Q*-Learning-based bitrate request method to efficiently control the selection of the segment quality by diminishing the quality switches and the occurrence of playback interruption. Bokani et al. [25] proposed another Markov Decision Process-based rate adaptation method based on *Q*-Learning to gradually learn the optimal decisions in order to avoid playback interruption, which has been found as the most important factor affecting user *QoE*.

It is worth noting that all of the abovementioned methods are designed based on the assumption that multiple DASH users make their rate adaptation decisions *separately*. In practical applications, it is more common that multi-users request multimedia content simultaneously from a single server or a relay server of a cell. Since the export bandwidth of the server is limited, a natural problem concerns how to allocate the limited bandwidth *jointly* to users so as to improve the performance of the whole system, especially to guarantee the fairness of multi-users. In [26], Jiang et al. proposed an optimized bandwidth estimator based on the mean of the previous bandwidths for multi-users to increase bandwidth utilization and stabilize the buffer lengths of multi-users. Li et al. [27] demonstrated that the discrete nature of the video bitrates leads to video bitrate oscillation that negatively affects the video viewing experience and presented a probe-and-adapt bandwidth estimation approach to further increase the bandwidth utilization and stabilize the requested video bitrates of multi-users. Essaili et al. [28] proposed *QoE*-maximization based traffic and resource management in a mobile network for multi-user adaptive HTTP streaming. However, a proxy is needed to intercept and rewrite the user HTTP requests in this method, which increases the system's complexity and the channel information for each user is based on the average channel statistics in the previous second that may not be accurate enough.

3 Proposed Rate Adaptation Algorithm

As shown in Fig. 2, in a DASH-based video delivery system, multi-users compete the limited export bandwidth of the server, and the information (i.e., the requested bitrate of the last segment and the current buffer length) of users is unknown by each other. The DASH users first send their current buffer lengths to the server, and then request video segments based on the payoff variation information that is calculated by the server based on the server export bandwidth and the buffer information of each user. We formulate the rate adaptation algorithm into a non-cooperative game as follows.

The *players* in the game are the DASH users. The *strategy* of each player is the requested bitrate (denoted by r_i for the ith user). The *payoff* or *profit* for the ith user (denoted by U_i) is its *QoE* determined by the requested video and accumulated buffer. The *commodity* of the competition is the video segments encoded into different bitrates. The solution of this game is Nash Equilibrium. Let $G = \{I, \{\mathbf{R}_i\}, \{U_i(\cdot)\}\}$ denote the non-cooperative rate adaptation game where $I = \{1, 2, \ldots, N\}$ is the index set for the users in the DASH system, and $\mathbf{R}_i$ and $U_i(\cdot)$ are the strategy space and utility function of the ith user, respectively. Each user determines the required bitrate r_i such that $r_i \in \mathbf{R}_i$. Let the rate vector $\mathbf{r} = \{r_1, \ldots, r_i, \ldots, r_N\}$ denote the outcome of the game in terms of the requested bitrates of all users. The resulting utility for the ith user is $U_i(\mathbf{r})$. We will occasionally use $U_i(r_i, \mathbf{r}_{-i})$ to replace $U_i(\mathbf{r})$ to indicate the dependence among DASH users, where $\mathbf{r}_{-i}$ denotes the vector that consists of elements of $\mathbf{r}$ without the ith element, i.e., $\mathbf{r}_{-i} = \{r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_N\}$.

In the following sections, we first propose a novel *QoE* model for DASH users by taking the current buffer length, reference buffer length, and video quality into account. Then, the existence of Nash Equilibrium of the non-cooperative game is theoretically proven. Finally, a distributed iterative algorithm with stability analysis is proposed to find the Nash Equilibrium.

3.1 Modeling of DASH User QoE

In a DASH based video delivery system, the user QoE depends on both the qualities of received video segments and playback interruptions.

1) Video quality: The video quality is directly determined by the bitrate of the requested video.

Although there are a lot of video quality-bitrate models, most of them can be uniformly represented as a logarithmic function of bitrate [29], i.e.,

$$q_i(r_i) = \alpha_i \log(1 + \beta_i r_i), \tag{1}$$

where q_i is the quality of the received video segment measured in peak-signal-noise-ratio (PSNR), structure similarity index metric (SSIM) [30], etc., and α_i and β_i are parameters depending on video content. Therefore, without loss of generality, Eq. (1) is used to evaluate the quality of received video segments in the proposed *QoE* model.

2) Playback interruption: We evaluate the influence of playback interruptions on user *QoEs* by explicitly modeling the relationship between the estimated buffer length and requested video bitrate. Usually, for the ith user, the buffer variation can be calculated as the difference between the cumulated buffer length (i.e., the remaining video playback time) caused by the downloaded video segment and the consumed buffer length caused by the played video content during the download time [19], [31], i.e.,

$$\Delta b_i(r_i) = T - T \cdot r_i / B_W^i, \tag{2}$$

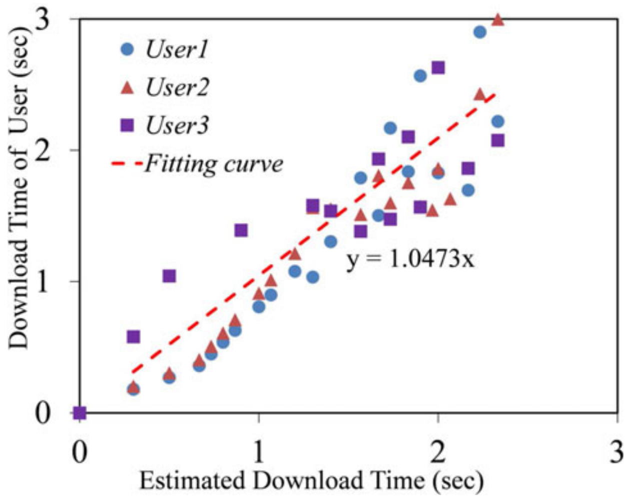


Fig. 3. Illustration of the relationship between the actual download time and estimated download time using $T \cdot \sum_{j=1}^{N} r_j / B_W$ of each user. Here, three users with varied channel throughputs compete the fixed export bandwidth of a server, and the correlation coefficient is 0.8658.

where $\Delta b_i(r_i)$ is the buffer variation caused by the requested video bitrate r_i, T is the length of a video segment, and B_W^i is the available channel bandwidth of the ith user. Unfortunately, as aforementioned, the channel state of each user is unknown, resulting in Eq. (2) being worthless. But the export bandwidth of the server is known to users. When the summation of the requested video bitrates of all users is larger than the export bandwidth of the server, the download time of all users will increase since the throughput of the system increases.

Therefore, we use the total requested video bitrates of all the users over the export bandwidth of the server to estimate the download time for each user, and the *average buffer variation* of all users in the system is

$$\Delta b(\mathbf{r}) = T - \omega \cdot T \cdot \sum_{i=1}^{N} r_i / B_W, \tag{3}$$

where ω is a coefficient, and B_W is the export bandwidth of a server. Such a simple approximation can facilitate the following theoretical analysis with reasonable accuracy. For a single user, the relationship between the actual download time and the estimated download time may not exactly be linear, as shown in "*User1*", "*User2*", and "*User3*" in Fig. 3. But for all three users, the overall relationship between the actual download time and estimated download time of all the users can be modeled using a proportional function with the correlation coefficient of 0.8658.

Furthermore, after the current video segment was downloaded, the estimated buffer length of the ith user considering other users denoted as $b_i^{est}(r_i, \mathbf{r}_{-i})$ can be derived by integrating $\Delta b(\mathbf{r})$ over r_i:

$$\begin{aligned} & b_i^{est}(r_i, \mathbf{r}_{-i}) \\ & = \int \Delta b(\mathbf{r}) \mathrm{d}r_i \\ & = T \cdot r_i - \omega \cdot T \cdot \left(\frac{1}{2} r_i^2 + r_i \sum_{\substack{j=1 \\ j \neq i}}^{N} r_j \right) / B_W + b_0 \\ & = \Phi(r_i) - \omega \cdot \Psi(\mathbf{r}) + b_0, \end{aligned} \tag{4}$$

where b_0 is a constant, denoting the initial average buffer length of all the users, $\Phi(r_i) = T \cdot r_i$ is the benefit gained from accumulated buffer, and $\Psi(\mathbf{r}) = T \cdot \left(\frac{1}{2} r_i^2 + r_i \sum_{j=1 j \neq i}^{N} r_j \right) / B_W$ represents the *system penalty* (buffer consumption) caused by the requested bitrates of all the users.

In order to ensure the buffer is equipped with an optimal state, i.e., the buffer length should be kept within a certain reference level, an adjustment factor (denoted as A_f) is adopted to modify the revenue function $\Phi(r_i)$, i.e.,

$$\Phi'(r_i) = A_f \cdot \Phi(r_i) = A_f \cdot T \cdot r_i. \tag{5}$$

Considering that a larger (resp. smaller) b_{curr} indicates more aggressive (resp. defensive) behaviors in requesting video segments, A_f is defined as a monotonically increasing function of the difference between the current buffer length b_{curr} and the reference buffer length b_{ref} [31]:

$$A_f = 2 \cdot \frac{e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}}, \tag{6}$$

where $p > 0$ is a constant, b_{ref} is the predefined reference buffer length, and b_{curr} is the current known buffer length before the video segment was downloaded for a certain user, respectively. The difference between b_{ref} and b_{curr} is an indicator to control the requested bitrates. When $b_{curr} > b_{ref}$, A_f is larger than 1, it indicates that the previously received video quality may be low. Then, the bitrates requested by users will be increased to achieve the optimal utility. When $b_{curr} < b_{ref}$, A_f is smaller than 1. This indicates that the probability of playback interruption is large. Then, the bitrates requested by users will be decreased to achieve the optimal utility.

Accordingly, the estimated buffer of the ith user with respect to all the other users can be rewritten as

$$\begin{aligned} & b_i^{est}(r_i, \mathbf{r}_{-i}) \\ & = \Phi'(r_i) - \omega \cdot \Psi(\mathbf{r}) + b_0 \\ & = 2 \cdot \frac{e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}} \cdot T \cdot r_i \\ & - \omega \cdot T \cdot \left(\frac{1}{2} r_i^2 + r_i \sum_{\substack{j=1 \\ j \neq i}}^{N} r_j \right) / B_W + b_0. \end{aligned} \tag{7}$$

Finally, we define the utility function of user i that considers the influence on other DASH users sharing the same network resources as the linear combination of the quality function in (1) and the buffer function in (7), i.e.,

$$\begin{aligned} & U_i(r_i, \mathbf{r}_{-i}) \\ & = q_i(r_i) + \mu \cdot b_i^{est}(r_i, \mathbf{r}_{-i}) \\ & = \alpha_i \log(1 + \beta_i r_i) + \mu \left(2 \cdot \frac{e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}} \cdot T \cdot r_i \right) \\ & \quad - \mu \cdot \omega \cdot T \cdot \left(\frac{1}{2} r_i^2 + r_i \sum_{\substack{j=1 \\ j \neq i}}^{N} r_j \right) / B_W + \mu \cdot b_0 \\ & = \alpha_i \log(1 + \beta_i r_i) + \mu \left(2 \cdot \frac{e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}} \cdot T \cdot r_i \right) \\ & \quad - \nu \cdot T \cdot \left(\frac{1}{2} r_i^2 + r_i \sum_{\substack{j=1 \\ j \neq i}}^{N} r_j \right) / B_W + \mu \cdot b_0, \end{aligned} \tag{8}$$

where $\mu > 0$ is a weight to balance the two parts, and $\nu = \mu \cdot \omega$.

3.2 Proof of the Existence of Nash Equilibrium

The Nash Equilibrium of a game is a strategy profile with the property in which no player can increase its utility by choosing a different action when the other players' actions are given [32]. A Nash Equilibrium exists for a game G if the following two conditions are met:

a) the strategy space $\mathbf{R}_i$ is a non-empty, convex, and compact subset of Euclidean space $\mathbf{R}^N$;
b) the utility $U_i(\mathbf{r})$ is continuous in $\mathbf{r}$ and (at least) quasi-concave in r_i.

For the proposed non-cooperative game, the strategy space is composed of the user requested bitrates with the range of $[0, r^{max}]$, where r^{max} is the maximum requested video bitrate of the ith user. Thereby, there is no doubt that $\mathbf{R}_i$ is a non-empty and compact subset of Euclidean space $\mathbf{R}^N$. According to the definition of the convex set [33], for any $r_x, r_y \in \mathbf{R}_i$ and any ζ with $0 \le \zeta \le 1$, we have $0 \le \zeta r_x \le \zeta r^{max}$ and $0 \le (1-\zeta) r_y \le (1-\zeta) r^{max}$. Then, we can get $0 \le \zeta r_x + (1-\zeta) r_y \le r^{max}$. Therefore, $\zeta r_x + (1-\zeta) r_y \in \mathbf{R}_i$, indicating $\mathbf{R}_i$ is a convex set. Thus, condition a) is satisfied. For the utility function in (8), it is obviously a continuous function in terms of $\mathbf{r}$. Besides, the second derivatives of $U_i(\mathbf{r})$ with respect to all the r_i are

$$\begin{cases} \frac{\partial^2 U_i}{\partial r_i^2} = -\frac{\alpha\beta^2}{(1+\beta r_i)^2} - \frac{\nu \cdot T}{B_W}, \\ \frac{\partial^2 U_i}{\partial r_i \partial r_j} = -\frac{\nu \cdot T}{B_W}\big|_{i \neq j}, \end{cases} \tag{9}$$

which are negative because α, β, ν, T, and B_W are non-negative. Accordingly, the utility $U_i(\mathbf{r})$ is a strictly concave function of r_i (for all i) [33]. Therefore, there must exist a Nash Equilibrium in the proposed rate adaptation game.

For a non-cooperative game, the Nash Equilibrium can be achieved by jointly maximizing the utility functions of all players, i.e., the corresponding best response function of a player is defined as the strategy of the player with those of other players fixed:

$$\mathbf{B}_i(\mathbf{r}_{-i}) = \arg\max_{r_i \in \mathbf{R}_i} U_i(r_i, \mathbf{r}_{-i}). \tag{10}$$

When the Nash Equilibrium is achieved, the strategies of all players can be represented as $\mathbf{r}^* = \{r_1^*, r_2^*, \ldots, r_N^*\}$, where $r_i^* = \mathbf{B}_i(\mathbf{r}_{-i}^*)$ is the optimal strategy of the ith user and $\mathbf{r}_{-i}^* = \{r_1^*, \ldots, r_{i-1}^*, r_{i+1}^*, \ldots, r_N^*,\}$ is the set of the Nash Equilibrium of all users except user i.

3.3 Distributed Iterative Algorithm for Nash Equilibrium

Theoretically, the Nash Equilibrium can be obtained by solving the following equations:

$$\begin{aligned} \frac{\partial U_i(\mathbf{r})}{\partial r_i} = & \frac{\alpha_i\beta_i}{1+\beta_i r_i} + \mu \cdot T \cdot \frac{2e^{p(b_{curr}-b_{ref})}}{1+e^{p(b_{curr}-b_{ref})}} \\ & - \nu \cdot T \cdot \frac{\sum_{j=1}^N r_j}{B_W} = 0, \quad \forall i. \end{aligned} \tag{11}$$

Taking a DASH system with only 2 users as an example, we have,

$$\begin{cases} \frac{Z_{1,1}}{1+\beta_1 r_1} + Z_{2,1} - Z_3(r_1+r_2) = 0, \\ \frac{Z_{1,2}}{1+\beta_2 r_2} + Z_{2,2} - Z_3(r_1+r_2) = 0, \end{cases} \tag{12}$$

where

$$\begin{cases} Z_{1,1} = \alpha_1\beta_1, Z_{2,1} = \mu \cdot T \cdot \frac{2e^{p(b_{curr,1}-b_{ref})}}{1+e^{p(b_{curr,1}-b_{ref})}}, \\ Z_{1,2} = \alpha_2\beta_2, Z_{2,2} = \mu \cdot T \cdot \frac{2e^{p(b_{curr,2}-b_{ref})}}{1+e^{p(b_{curr,2}-b_{ref})}}, \\ Z_3 = \nu \cdot T / B_W. \end{cases} \tag{13}$$

Assume the two users request the same video and have the same channel condition (i.e., two identical users), we have $Z_{1,1} = Z_{1,2} = Z_1$ and $Z_{2,1} = Z_{2,2} = Z_2$. Then, the Nash Equilibrium can be expressed as

$$r_1^* = r_2^* = \frac{-(2Z_3 - \beta_i Z_2) + \sqrt{(2Z_3 + \beta_i Z_2)^2 + 8\beta_i Z_1 Z_3}}{4\beta_i Z_3}. \tag{14}$$

From the above analysis, to determine the requested video bitrate for a certain user, the strategies of the other users must be available. However, such user strategies and information are unknown to each other in a practical DASH system. In order to adjust the requested bitrate r_i for the ith user, we propose to employ its own information (i.e., the requested bitrate of the last segment and the current buffer length) and communicate with the server to obtain the payoff variation that is induced by the varied download time. Therefore, the requested video bitrate r_i can be updated based on the sub-gradient method [34], [35], [36]:

$$r_i(t+1) = r_i(t) + \theta_i r_i(t) \frac{\partial U_i(\mathbf{r})}{\partial r_i(t)}, \tag{15}$$

where $\theta_i > 0$ is the speed adjustment parameter (i.e., learning rate) of user i.

In an actual system, the value of $\partial U_i(\mathbf{r}) / \partial r_i(t)$ (i.e., the payoff variation information in Fig. 2) can be estimated by the server and transmitted to the user as,

$$\frac{\partial U_i(\mathbf{r})}{\partial r_i(t)} \approx \frac{U_i^+(\mathbf{r}^+) - U_i^-(\mathbf{r}^-)}{2\varepsilon}, \tag{16}$$

with

$$\begin{cases} \mathbf{r}^+ = \{r_1(t), \ldots, r_i(t)+\varepsilon, \ldots, r_N(t)\}, \\ \mathbf{r}^- = \{r_1(t), \ldots, r_i(t)-\varepsilon, \ldots, r_N(t)\}, \end{cases} \tag{17}$$

where ε is an especially small value (e.g., $\varepsilon = 0.0001$). When the Nash Equilibrium is achieved, we have $r_i(t+1) = r_i(t)$ for any i, i.e.,

$$\begin{cases} \mathbf{r}(t+1) = \mathbf{r}(t), \\ \frac{\partial U_i(\mathbf{r})}{\partial r_i(t)} = 0. \end{cases} \tag{18}$$

3.4 Stability Analysis for the Distributed Iterative Algorithm

The stability of the distributed strategy update algorithm (15) is analyzed by using the *Routh-Hurvitz* stability condition [37], [38], [39], [40], which judges the distribution of the eigenvalues (denoted as λ_i) of the *Jacobian matrix*. That is, if all of the eigenvalues are inside a unit circle of the complex plane (i.e., $|\lambda_i| < 1$), the Nash Equilibrium point is stable. Taking a DASH system with only two users as an example, the *Jacobian matrix* can be expressed as,

$$\mathbf{J}(r_1, r_2) = \begin{bmatrix} \frac{\partial r_1(t+1)}{\partial r_1(t)} & \frac{\partial r_1(t+1)}{\partial r_2(t)} \\ \frac{\partial r_2(t+1)}{\partial r_1(t)} & \frac{\partial r_2(t+1)}{\partial r_2(t)} \end{bmatrix} = \begin{bmatrix} j_{1,1} & j_{1,2} \\ j_{2,1} & j_{2,2} \end{bmatrix}, \tag{19}$$

where

$$\begin{cases} j_{1,2} = -\theta_1 Z_3 r_1 \\ j_{2,1} = -\theta_2 Z_3 r_2 \\ j_{1,1} = 1 + \theta_1\Big(-\frac{\beta_1 Z_{1,1} r_1}{(1+\beta_1 r_1)^2} + \frac{Z_{1,1}}{1+\beta_1 r_1} + Z_{2,1} - Z_3(2r_1 + r_2)\Big) \\ j_{2,2} = 1 + \theta_2\Big(-\frac{\beta_2 Z_{1,2} r_2}{(1+\beta_2 r_2)^2} + \frac{Z_{1,2}}{1+\beta_2 r_2} + Z_{2,2} - Z_3(r_1 + 2r_2)\Big) \end{cases} \tag{20}$$

The two eigenvalues can be obtained by solving the characteristic equation:

$$\lambda^2 - \lambda(j_{1,1} + j_{2,2}) + (j_{1,1}j_{2,2} - j_{1,2}j_{2,1}) = 0, \tag{21}$$

whose solution is

$$\begin{cases} \lambda_1 = \frac{(j_{1,1}+j_{2,2}) + \sqrt{(j_{1,1}-j_{2,2})^2 + 4j_{1,2}j_{2,1}}}{2}, \\ \lambda_2 = \frac{(j_{1,1}+j_{2,2}) - \sqrt{(j_{1,1}-j_{2,2})^2 + 4j_{1,2}j_{2,1}}}{2}. \end{cases} \tag{22}$$

Assume the two users are identical (i.e., $Z_{1,1} = Z_{1,2} = Z_1 = \alpha\beta$ and $Z_{2,1} = Z_{2,2} = Z_2$), and when the Nash equilibrium point is achieved (i.e., $r_1^* = r_2^*$) and the buffer length is in a steady state (i.e., $b_{curr} = b_{ref}$ and $Z_{2,1} = Z_{2,2} = Z_2 = \mu \cdot T$), we can derive that $j_{1,1} = j_{2,2}$, $j_{1,2} = j_{2,1}$, $\lambda_1 = j_{1,1} - j_{1,2}$, and $\lambda_2 = j_{1,1} + j_{1,2}$. Therefore, the condition to ensure the stability of the proposed algorithm is expressed as

$$\begin{cases} -1 < j_{1,1} - j_{1,2} < 1, \\ -1 < j_{1,1} + j_{1,2} < 1. \end{cases} \tag{23}$$

For $|j_{1,1} - j_{1,2}| < 1$, substituting Eq. (20) into (23), we have

$$\begin{aligned} -2 < \theta_1 &\left[-\frac{\beta Z_1 r_1}{(1+\beta r_1)^2} + \frac{Z_1}{1+\beta r_1} + Z_2 - Z_3(2r_1 + r_2) \right] \\ &+ \theta_1 Z_3 r_1 < 0. \end{aligned} \tag{24}$$

At the Nash Equilibrium point, since $\theta_1 = \theta_2 = \theta^*$, and $r_1 = r_2 = r^*$, Eq. (24) can be rewritten as

$$-\frac{2}{\theta^*} < -\frac{\beta Z_1 r^*}{(1+\beta r^*)^2} + \frac{Z_1}{1+\beta r^*} + Z_2 - 2Z_3 r^* < 0. \tag{25}$$

Furthermore, Eq. (25) can be simplified as

$$\begin{cases} Z_1 + Z_2(1+\beta r^*)^2 < 2Z_3 r^*(1+\beta r^*)^2, \\ Z_1 + \big(Z_2 + \frac{2}{\theta^*}\big)(1+\beta r^*)^2 > 2Z_3 r^*(1+\beta r^*)^2. \end{cases} \tag{26}$$

Substituting $T = 2$, Z_1, Z_2, and Z_3 into (26), we can obtain

$$\begin{cases} \alpha\beta + 2\mu(1+\beta r^*)^2 < \frac{4\nu}{B_W} r^*(1+\beta r^*)^2, \\ \alpha\beta + \big(2\mu + \frac{2}{\theta^*}\big)(1+\beta r^*)^2 > \frac{4\nu}{B_W} r^*(1+\beta r^*)^2. \end{cases} \tag{27}$$

Similarly, for $|j_{1,1} + j_{1,2}| < 1$, the stability condition is,

$$\begin{cases} \alpha\beta + 2\mu(1+\beta r^*)^2 < \frac{8\nu}{B_W} r^*(1+\beta r^*)^2, \\ \alpha\beta + \big(2\mu + \frac{2}{\theta^*}\big)(1+\beta r^*)^2 > \frac{8\nu}{B_W} r^*(1+\beta r^*)^2. \end{cases} \tag{28}$$

Since α, β, μ, ν, θ^*, and B_W are all positive, we can conclude that the proposed distributed iterative updating algorithm is stable if the following conditions are satisfied:

$$\begin{cases} \alpha\beta + 2\mu(1+\beta r^*)^2 < \frac{4\nu}{B_W} r^*(1+\beta r^*)^2, \\ \alpha\beta + \big(2\mu + \frac{2}{\theta^*}\big)(1+\beta r^*)^2 > \frac{8\nu}{B_W} r^*(1+\beta r^*)^2. \end{cases} \tag{29}$$

When there are more than two users in the system, (30) must be satisfied for user i,

$$\frac{Z_{1,i}}{1+\beta_i r_i} + Z_{2,i} - Z_3\left(\sum_{j=1}^{N} r_j\right) = 0, \tag{30}$$

Equation (30) of all users can be expressed by matrix format as

$$\mathbf{Z}_1 + (\mathbf{1} + \mathbf{r} \cdot \beta)\mathbf{Z}_2 = Z_3(\mathbf{r} \cdot \mathbf{1}^T)(\mathbf{1} + \mathbf{r} \cdot \beta), \tag{31}$$

where

$$\begin{cases} \mathbf{r} = [r_1 \cdots r_i \cdots r_N], \\ \mathbf{1} = [1 \cdots 1 \cdots 1], \\ \mathbf{Z}_1 = [Z_{1,1} \cdots Z_{1,i} \cdots Z_{1,N}], \\ \beta = diag(\beta_1, \ldots, \beta_N), \\ \mathbf{Z}_2 = diag(Z_{2,1}, \ldots, Z_{2,N}). \end{cases} \tag{32}$$

The ***Jacobian matrix*** of the Nash Equilibrium for multi-users is given as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial r_1(t+1)}{\partial r_1(t)} & \cdots & \frac{\partial r_1(t+1)}{\partial r_N(t)} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_N(t+1)}{\partial r_1(t)} & \cdots & \frac{\partial r_N(t+1)}{\partial r_N(t)} \end{bmatrix}. \tag{33}$$

Then, similar to the DASH system with only 2 users, the local stability condition can also be analyzed.

Algorithm 1. Distributed Iterative Algorithm of Rate Adaptation

1: Initially, all users request the bitrate $r(1) = 0.1$ Mbps to the server so as to quickly establish the predefined initial buffer length (e.g. 2 s).
2: ***while*** $n \leq S$ (S is the total number of requested segments) ***do***
3: Each user sends the payoff request to the server;
4: The server sends the payoff information to corresponding user;
5: The users update the requested bitrate $r(n)$ according to (15) and request video segments from the server;
6: The server sends the video segments to the users;
7: Update the buffer information $b_{curr}(n)$;
8: $n = n + 1$;
9: ***end while***

Algorithm 1 shows the detailed procedure of the proposed method. The DASH users first request the bitrate of 0.1Mbps to quickly establish the initial buffer length. Then, the users send their buffer information to the server. Third, the server

TABLE 1
Detailed Information of the Tested DASH Dataset

Name	Resolution	Target Bitrate (Mbps)
BigBuckBunny	480×360 704×480 1280×720	0.1, 0.2, 0.3, 0.4 0.5, 0.6, 0.7, 0.9, 1.0 1.2, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0
ElephantsDream	640×360 704×480 1280×720	0.1, 0.2, 0.3, 0.4 0.5, 0.6, 0.7, 0.9, 1.0 1.2, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0
SitaSingstheBlues	640×360 704×480 1280×720	0.1, 0.2, 0.3, 0.4 0.5, 0.6, 0.7, 0.9, 1.0 1.2, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0

calculates and sends the payoff variation information for each user based on the export bandwidth and the buffer lengths of users. Finally, the users update the requested bitrates and request video segments from the server.

4 Simulation Results

4.1 Simulation Setup

To verify the performance of the proposed method, the *LibDASH* platform [41], [42] is used. Users request video segments from the server, which is hosting an Apache HTTP Web server [43]. And the server export bandwidth is controlled by *DummyNet* [44]. The video dataset includes *BigBuckBunny* [45], [46], *ElephantsDream* [47], and *SitaSingstheBlues* [48], [49]. Each video was encoded by *FFMPEG* [50] with 20 various bitrates from low to high, as shown in Table 1. Fig. 4 shows the two parameters of the quality model in (1) (i.e., α and β) for each video. The parameters α and β are obtained by fitting Eq. (1) using the actual qualities and bitrates of each segment. The lengths of each video segment and the initial buffer of each user are set as 2s. For the proposed method, all users are equipped with the same learning rate in (15), i.e., $\theta_1 = \cdots = \theta_N = \theta$, for all i, to ensure the synchronization of the convergence of the distributed algorithm among all users in the system. Meanwhile, the initial requested bitrates of all users are set as 0.1 Mbps.

The proposed algorithm is validated under four cases:

Case 1, two identical users (*without limitations on user channel throughputs*) request the same video content (i.e., *BigBuckBunny*) with a *fixed* server export bandwidth;

Case 2, two identical users (*without limitations on user channel throughputs*) request the same video content (i.e., *BigBuckBunny*) with a ***varied*** server export bandwidth;

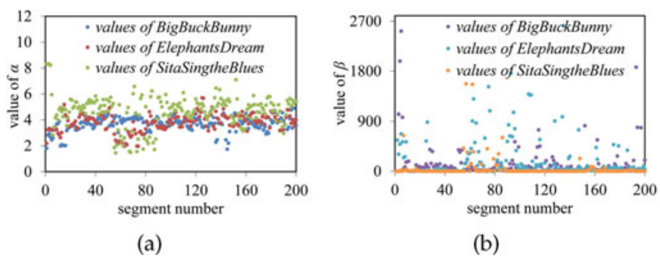


Fig. 4. Quality model parameters (a) α_1, α_2, α_3 and (b) β_1, β_2, β_3 of the video datasets in the simulation.

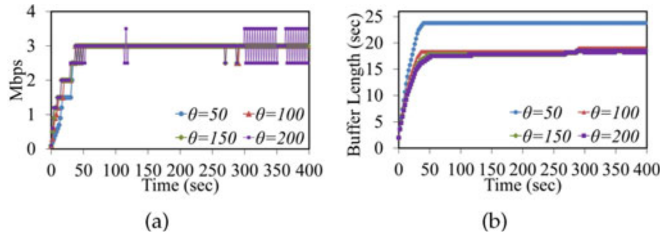


Fig. 5. Results of case 1, in which two identical users compete the server export bandwidth of 6 Mbps, and request the *BigBuckBunny* video sequence with learning rate $\theta = 50$, 100, 150, and 200. Here, $\mu = 0.003$, $\nu = 0.0041$, $\alpha = 2.15$, $\beta = 0.0827$, $r^* = 3$ Mbps, and $B_W = 6$ Mbps and the inequality relation of (29) is tenable. (a) Dynamic behavior of requested bitrates, and (b) actual buffer lengths of the two users. *Note that the states of the two users are identical.*

Case 3, three users (*with fixed limitations on user channel throughputs*) request different video contents (i.e., *User1*, *User2*, and *User3* request *BigBuckBunny*, *ElephantsDream*, and *SitaSingstheBlues*, respectively) with a *varied* server export bandwidth;

Case 4, three users (*with random limitations on user channel throughputs*) request different video contents with both *fixed* and *varied* server export bandwidth. Besides we also compare the *Proposed* method with three algorithms, i.e., the *Quality_First* method (*QF*) [51], *Buffer_First* method (*BF*) [51], and *QoE-based Buffer-aware Resource Allocation* method (*QBA*) [28].

4.2 Results of Case 1

In this case, two users compete the limited server export bandwidth of size 6Mbps. The reference buffer length is set as 15 s. Fig. 5 shows the requested bitrates and buffer lengths of the two users by the proposed method with different learning rates. We can observe that the Nash Equilibrium is achieved at $r^* = 3$ Mbps, and the actual buffer length converges to 15-20 s except for the case where the learning rate equals 50. When the learning rate increases (e.g., $\theta = 200$), the requested bitrate varies severely; however, the reference buffer is achieved more accurately as shown in Fig. 5b. Table 2 compares the average requested bitrate and bitrate switching frequency of the proposed method with different learning rates (θ). We can observe that the average requested bitrates are similar, while the minimum bitrate switching frequency is achieved when $\theta = 100$.

Taking the learning rate of $\theta = 100$ as an example, Fig. 6 shows that the Nash Equilibrium is achieved in a slower pace with the reference buffer increasing. Specifically, the converged buffer lengths are 13.32 s, 17.93 s, and 22.68 s when the reference buffer lengths are set as 10 s, 15 s, and 20 s, respectively. The reason is that the buffer needs more time to accumulate.

TABLE 2
Comparisons of Average Quality and Average Number of Switches of the Proposed Method with Different Learning Rates under Case 1

Learning Rate θ	50	100	150	200
Average Bitrate (Mbps)	2.822	2.858	2.862	2.862
Average PSNR (dB)	44.602	44.700	44.683	44.654
Average SSIM	0.996	0.997	0.997	0.997
Average Number of Switches	12	11	17	67

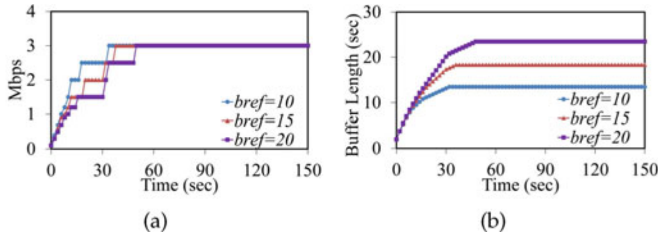


Fig. 6. Results of case 1, in which two identical users compete the server export bandwidth of 6Mbps, and request the *BigBuckBunny* video sequence with learning rate $\theta = 100$. Here, $\mu = 0.003$, $\nu = 0.0041$, $\alpha = 2.15$, $\beta = 0.0827$, $r^* = 3$ Mbps, and $B_W = 6$ Mbps and the inequality relation of (29) is tenable. (a) Dynamic behavior of requested bitrates, and (b) actual buffer lengths of the two users with the reference buffer lengths of 10 s, 15 s, and 20 s, respectively. *Note that the states of the two users are identical*.

4.3 Results of Case 2

We verify the proposed method with a varied server export bandwidth that is realized via three types of variations [52], i.e., persistent variation, staged variation, and short-term variation, as shown in Fig. 7. The persistent and staged bandwidth variations (both increment and decrement) that last for tens of seconds appear frequently in practice when the cross traffic in the path's bottleneck varies significantly due to arriving or departing traffic of some users. A good rate adaptation method should adapt to such variations by decreasing or increasing the requested bitrate. The short-term variation that lasts for only a few seconds is usually caused by burst change of channel states. For such short-term variations, the user should be able to keep requested bitrate stable to avoid unnecessary bitrate variations.

Fig. 8 shows the requested bitrates and buffer lengths of two users by the proposed method when the server export bandwidth exhibits persistent variation. We can observe that the requested bitrate increases to the Nash Equilibrium rapidly, and the reference buffer is also reached before $t = 50$ s. When the server export bandwidth varies to 9 Mbps at the time of 100 s, the requested bitrate with $\theta = 100$ quickly increases to 4.5 Mbps, while the requested bitrate with $\theta = 50$ first increases to 5 Mbps, which consumes about 5 s playout buffer before dropping to 4.5 Mbps. When the available bandwidth decreases back to 6 Mbps at the time of 200s, the requested bitrate with $\theta = 100$ drops to 3 Mbps directly,

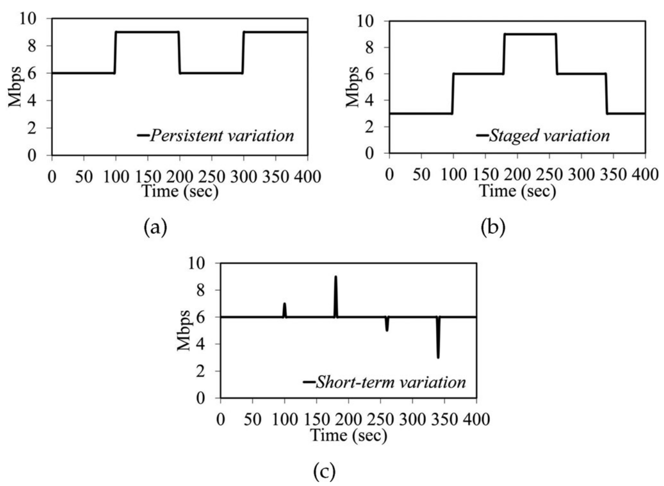


Fig. 7. Three kinds of server bandwidth variations, (a) persistent variation, (b) staged variation, and (c) short-term variation.

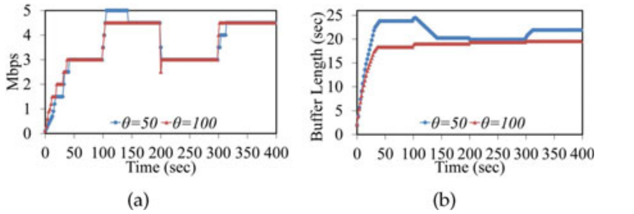


Fig. 8. Results of case 2, in which two identical users compete the server export bandwidth with *persistent variation*, and request the *BigBuckBunny* video sequence with learning rate $\theta = 50$ and 100 respectively. Here, $\mu = 0.003$, $\nu = 0.0041$. (a) Dynamic behavior of requested bitrates, and (b) actual buffer lengths of the two users with the reference buffer length of 15 s. The initial server export bandwidth is 6 Mbps. *Note that the states of the two users are identical*.

while the requested bitrate with $\theta = 50$ first drops to 3.5 Mbps. When the available bandwidth increases back to 9 Mbps at the time of 300 s, the requested bitrate with $\theta = 50$ steps to 4.5 Mbps, while the requested bitrate with $\theta = 100$ increases to 4.5 Mbps directly. We can conclude that a larger learning rate can follow the bandwidth variation accurately and keep the buffer lengths more stable, while the requested bitrate is more stable for a smaller learning rate.

Fig. 9 shows the requested bitrates and buffer lengths of two users by the proposed method when the server export bandwidth exhibits staged variation. Similarly, we can also observe that the requested bitrate with a smaller learning rate (e.g., $\theta = 50$) is more stable and the bitrate fluctuations are smaller at the time of bandwidth changing (i.e., at 100 s, 180 s, 260 s, and 340 s). However, the buffer length is further away from the reference buffer length than the learning rate of $\theta = 100$. Nevertheless, the Nash Equilibrium can be achieved under both cases.

For the case of short-term server export bandwidth variation, the requested bitrate with a smaller learning rate ($\theta = 50$) is more stable, as shown in Fig. 10. The requested bitrate with a smaller learning rate can avoid abrupt short-term changing by accumulating/consuming buffered video segments when encountering a small amplitude bandwidth variation, e.g., at the time of 100 s and 260 s, as shown in Fig. 10a. But, the difference between the actual buffer length and the reference buffer length of $\theta = 50$ is larger than that of the larger learning rate, i.e., $\theta = 100$, as shown in Fig. 10b.

4.4 Results of Case 3

This bsection presents the performance of the proposed algorithm under *Case 3*, in which three users request three

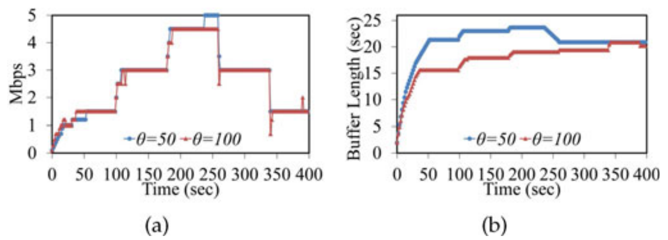


Fig. 9. Results of case 2, in which two identical users compete the server export bandwidth with *staged variation*, and request the *BigBuckBunny* video sequence with learning rate $\theta = 50$ and 100 respectively. Here, $\mu = 0.003$, $\nu = 0.0041$. (a) Dynamic behavior of requested bitrates, and (b) actual buffer lengths of the two users with the reference buffer length of 15 s. The initial server export bandwidth is 6 Mbps. *Note that the states of the two users are identical*.

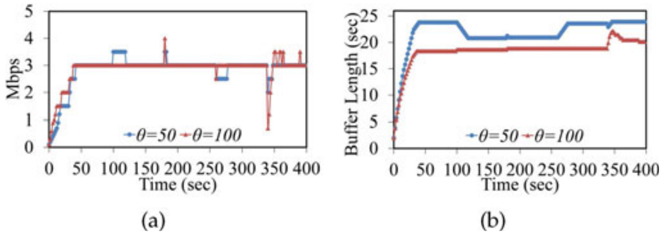


Fig. 10. Results of case 2, in which two identical users compete the server export bandwidth with *short-term variation*, and request the *BigBuckBunny* video sequence with learning rate $\theta = 50$ and 100 respectively. Here, $\mu = 0.003$, $\nu = 0.0041$. (a) Dynamic behavior of requested bitrates, and (b) actual buffer lengths of the two users with the reference buffer length of 15 s. The initial server export bandwidth is 6 Mbps. *Note that the states of the two users are identical*.

different videos (i.e., *BigBuckBunny* for *User1*, *ElephantsDream* for *User2*, and *SitaSingstheBlues* for *User3*) with a varied server export bandwidth. Note that the channel throughput of each user is limited up to 1.5 Mbps, which is unknown to the users, and the server export bandwidth is set as 6 Mbps. The reference buffer length is set as 15s.

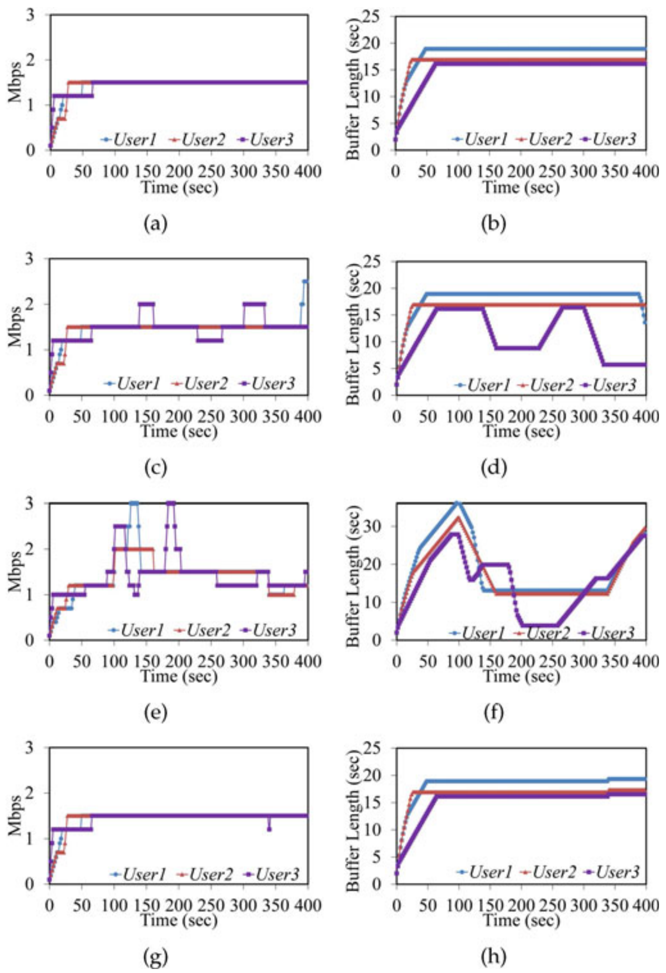


Fig. 11. Results of case 3, in which three users (with *fixed and limited* throughputs of 1.5 Mbps) compete the server export bandwidth (6 Mbps), and request *BigBuckBunny*, *ElephantsDream*, and *SitaSingstheBlues*, respectively, with learning rate $\theta = 50$. Here, $\mu = 0.003$, $\nu = 0.0041$. (a), (c), (e), and (g) show the requested bitrates for fixed, persistent variation, staged variation, and short-term variation of server export bandwidth. (b), (d), (f), and (h) show the corresponding buffer length of each user.

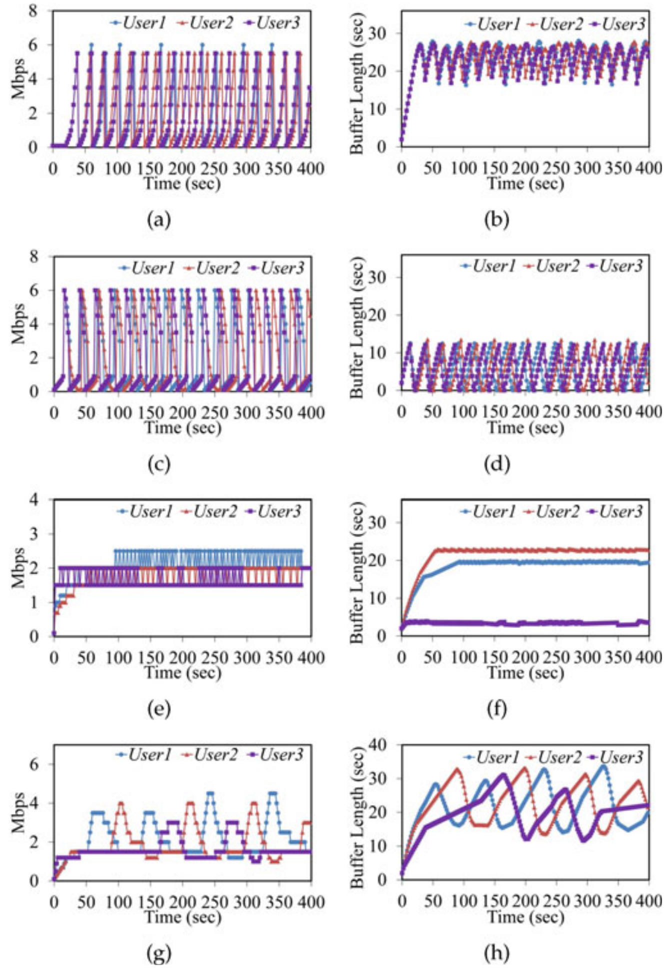


Fig. 12. Results of case 4, in which three users (with *random and limited* throughputs) compete the server export bandwidth (*fixed* 6 Mbps), and request *BigBuckBunny*, *ElephantsDream*, and *SitaSingstheBlues*, respectively, with learning rate $\theta = 50$. Here, $\mu = 0.003$, $\nu = 0.0041$. (a), (c), (e), and (g) show the requested bitrates of *BF*, *QF*, *QBA*, and the *Proposed* methods; while (b), (d), (f), and (h) show the corresponding buffer length of each user.

From Fig. 11, we can observe that the Nash Equilibrium is achieved with $r_1^* = 1.5$ Mbps, $r_2^* = 1.5$ Mbps, and $r_3^* = 1.5$ Mbps, for different server export bandwidth variation scenarios, and no playback interruption occurs.

4.5 Results of Case 4

In addition, we also compare the proposed method (denoted by *Proposed*) with other methods (i.e., *BF*, *QF*, and *QBA*) with *random* user channel throughput limitations that are also unknown for each user.

Fig. 12 compares the requested bitrate and the buffer length of each user when the server export bandwidth is fixed to 6 Mbps. We can observe that the *BF* method switches the video bitrate frequently (see Fig. 12a) in order to ensure the actual buffer length is close to the reference buffer length (see Fig. 12b). As shown in Figs. 12c and 12d, the *QF* method first accumulates a certain buffer length, and then struggles to request the highest bitrates, resulting in frequent re-buffering and bitrate switching. For the *QBA* method, its requested bitrates are more stable than those of *BF* and *QF* methods, but the buffer lengths of the three users are not fair, i.e., the buffer length of *User3* is much smaller than that of

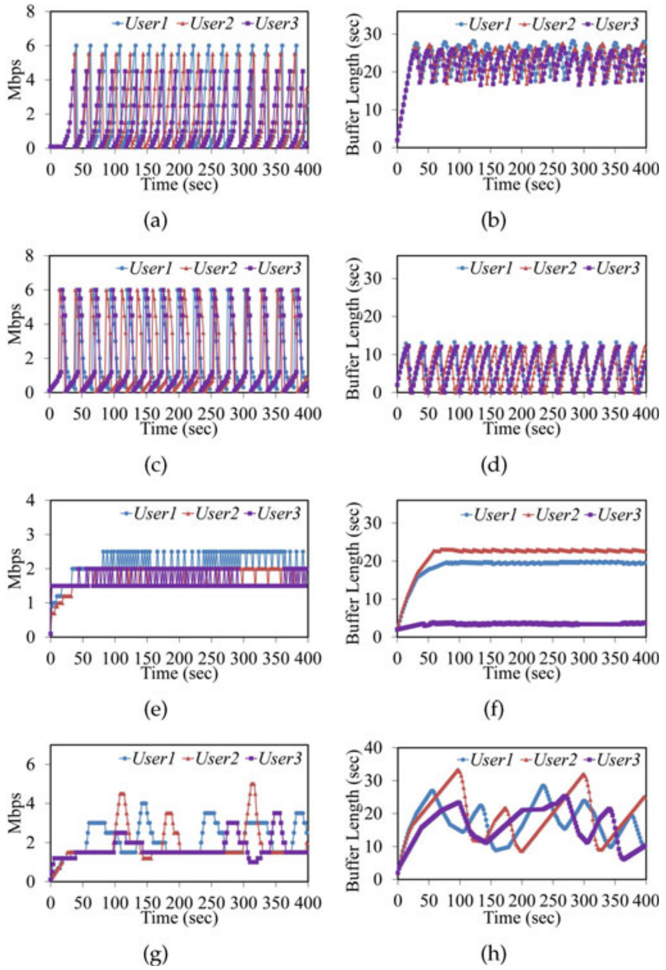


Fig. 13. Results of case 4, in which three users (with *random and limited* throughputs) compete the server export bandwidth (*persistent variation*), and request *BigBuckBunny*, *ElephantsDream*, and *SitaSingstheBlues*, respectively, with learning rate $\theta = 50$. Here, $\mu = 0.003$, $\nu = 0.0041$. (a), (c), (e), and (g) show the requested bitrates of *BF*, *QF*, *QBA*, and the *Proposed* methods; while (b), (d), (f), and (h) show the corresponding buffer length of each user.

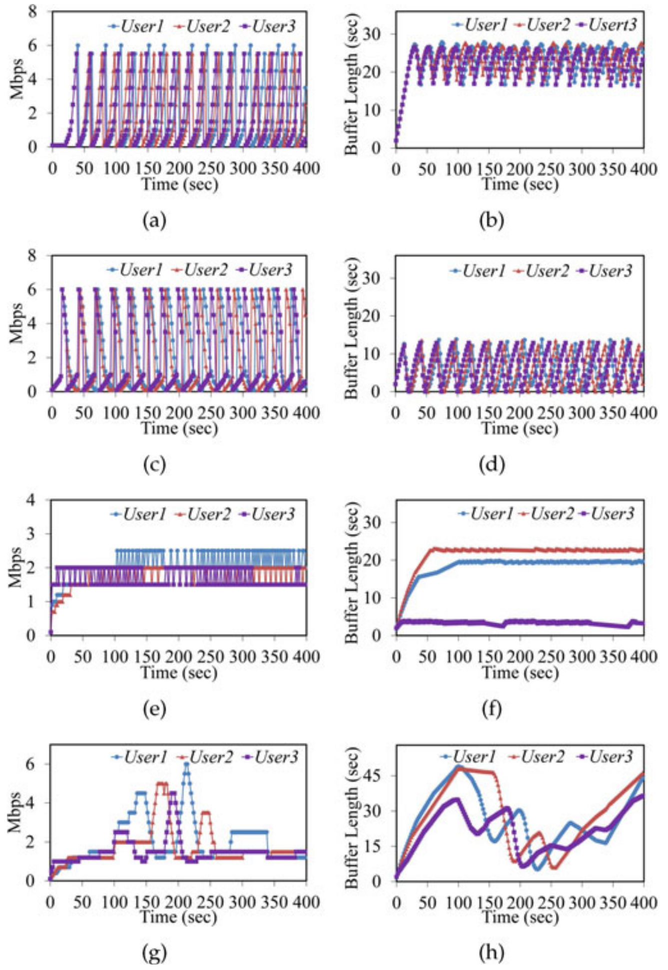


Fig. 14. Results of case 4, in which three users (with *random and limited* throughputs) compete the server export bandwidth (*staged variation*), and request *BigBuckBunny*, *ElephantsDream*, and *SitaSingstheBlues*, respectively, with learning rate $\theta = 50$. Here, $\mu = 0.003$, $\nu = 0.0041$. (a), (c), (e), and (g) show the requested bitrates of *BF*, *QF*, *QBA*, and the *Proposed* methods; while (b), (d), (f), and (h) show the corresponding buffer length of each user.

User1 and *User2*, as shown in Figs. 12e and 12f. The reason is that the *QBA* method does not take the fairness of users into consideration too much. From Figs. 12g and 12h, it can be observed that the requested video bitrates by the *Proposed* method are stable for all of the three users, and their buffer lengths of the three users vary around the reference buffer length (set as 15 s). Besides, there is no re-buffering (playback interruption) for the *Proposed* method.

Similar to Fig. 12, the corresponding comparison results of the four methods with respect to the other three types of bandwidth variations, i.e., *persistent variation*, *staged variation*, and *short-term variation*, are given in Figs. 13, 14, and 15, respectively. Similar conclusions can be drawn, which consistently demonstrates the superiority of the proposed algorithm.

Detailed numerical comparisons of the four methods are given in Table 3, from which we can observe that the average received bitrate of the *QF* method is largest, but the bitrate fluctuations (see the standard deviation, average number of switches and average switching amplitude of received bitrates) are also the maximum, and there exist playback interruptions, while the bitrate fluctuation of the *BF* method is a little small but still large. It can also be observed that the average video quality (by observing the average SSIMs) of the *QF* and *BF* methods is obviously lower than that of the *QBA* and the *Proposed* methods, and the quality variations (by observing the standard deviation of SSIM values of received video segments) of the *QF* and the *BF* methods are larger than those of the *QBA* and the *Proposed* methods. Moreover, although the average bitrates and SSIM values of the *Proposed* method are similar to those of the *QBA* method, it is obvious that the amplitudes of bitrate switching and the numbers of switches of *Proposed* method are smaller, which means that the performance of the *Proposed* method is the best.

Last, we evaluate the performance of the four algorithms, i.e., *BF*, *QF*, *QBA*, and *Proposed*, by comparing their produced *QoE* values that are measured via two extensively used *QoE* models [53], [54]:

$$QoE_1 = \sum_{k=1}^{M} r[k] - \xi \sum_{k=1}^{M-1} |r[k+1] - r[k]| - \psi \sum_{k=1}^{M} \max\{0, T_{down}[k] - b[k]\}, \tag{34}$$

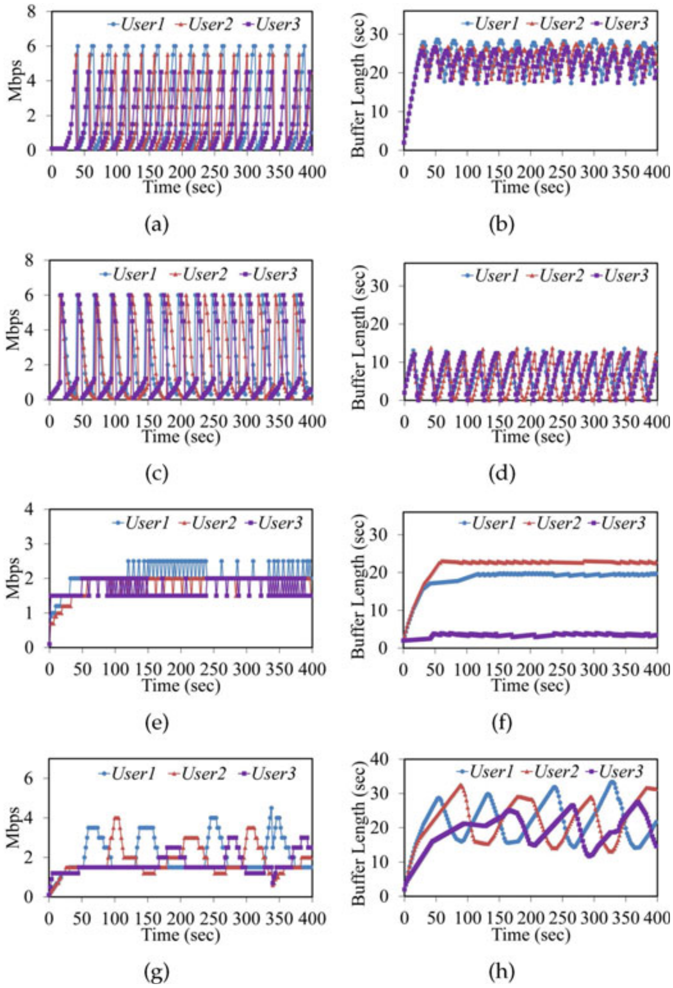


Fig. 15. Results of case 4, in which three users (with *random and limited* throughputs) compete the server export bandwidth (*short-term variation*), and request *BigBuckBunny*, *ElephantsDream*, and *SitaSingstheBlues*, respectively, with learning rate $\theta = 50$. Here, $\mu = 0.003$, $\nu = 0.0041$. (a), (c), (e), and (g) show the requested bitrates of *BF*, *QF*, *QBA*, and the *Proposed* methods; while (b), (d), (f), and (h) show the corresponding buffer length of each user.

$$\begin{aligned} QoE_2 = &\sum_{k=1}^{M} q[k] - \varphi \sum_{k=1}^{M-1} |q[k+1] - q[k]| \\ &- \sigma \sum_{k=1}^{M-1} \big(\max(0, b_{ref} - b[k+1])\big)^2 \\ &- \eta \sum_{k=1}^{M} \max\{0, T_{down}[k] - b[k]\}, \end{aligned} \tag{35}$$

where $\xi = 1$, $\psi = 6$, $\varphi = 2$, $\sigma = 0.001$ and $\eta = 2$ are model parameters that are empirically defined in [53] and [54], M is the number of received segments, $r[k]$ is the bitrate of the kth requested video segment, $q[k]$ is the corresponding SSIM value, $T_{down}[k]$ is the download time of the kth segment, and $b[k]$ is the buffer length at the end time of the kth segment and $b_{ref} = 15$s. Note that we set η to 2 instead of 50 in [54] to ensure that the QoE values are positive. From Tables 4 and 5, it can be observed that, compared with the other three methods, the proposed algorithm always produces the highest average QoE with respect to different types of bandwidth variations. Besides, the proposed method provides optimal QoE for most users, i.e., at least 2 out of 3 users.

5 Experimental Result for Realistically Modeled Networking Scenarios

We also established realistically modeled wireless and wired networking scenarios, in which 6 users requested different videos from a single server. In the wireless network, they were connected by a movable router (2.4 GHz, automatic frequency channel bandwidth, 802.11b/g/n mixed wireless mode, 1480 Byte maximum transmission unit configuration); yet in the wired network, the users connected to the server via the campus network of Shandong University which includes many switches and routers. To ensure a large server export bandwidth (which can be constrained to 6 Mbps by *DummyNet*), the experiments were conducted at night. The initial buffer of each user was set as 20 s in the experiments in order to calculate the initial playout delay.

TABLE 3
Detailed Comparisons of the Four Methods under Case 4 (The Best Results Are Bolded)

Bandwidth Variations	Methods	Average Bitrate (Mbps)	Standard Deviation of Bitrate	Average SSIM	Standard Deviation of SSIM	Average Switching Amplitude (Mbps)	Average Number of Switches	Average Buffer Length (sec)	Times of Interruption	Average Interruption Time Length (sec)
Fixed Bandwidth	***BF***	1.855	1.827	0.955	0.077	3.169	182	**22.988**	**0**	**0**
	QF	**1.968**	2.141	0.956	0.074	2.744	197	5.908	13	25.330
	QBA	1.851	**0.321**	**0.987**	0.031	1.491	86	14.429	**0**	**0**
	Proposed	1.845	0.783	0.986	**0.030**	**1.226**	**42**	21.111	**0**	**0**
Bandwidth with Persistent Variation	***BF***	1.767	1.766	0.953	0.077	3.082	177	**22.650**	**0**	**0**
	QF	**1.923**	2.075	0.962	0.062	2.714	199	6.424	10	20.670
	QBA	1.867	**0.320**	**0.987**	0.031	1.490	81	14.478	**0**	**0**
	Proposed	1.882	0.762	**0.987**	**0.028**	**1.279**	**39**	17.627	**0**	**0**
Bandwidth with Staged Variation	***BF***	1.858	1.859	0.954	0.080	3.191	178	**22.750**	**0**	**0**
	QF	**1.960**	2.092	0.956	0.073	2.653	194	6.028	7	14.670
	QBA	1.861	**0.321**	**0.987**	**0.031**	1.490	83	14.369	**0**	**0**
	Proposed	1.730	0.915	0.984	0.032	**1.218**	**39**	25.362	**0**	**0**
Bandwidth with Short-term Variation	***BF***	1.879	1.815	0.956	0.075	3.088	179	**23.050**	**0**	**0**
	QF	**1.989**	2.050	0.959	0.066	2.631	195	6.339	6	11.300
	QBA	1.868	**0.315**	**0.987**	**0.031**	1.487	63	14.412	**0**	**0**
	Proposed	1.843	0.728	0.986	**0.031**	**1.262**	**40**	20.911	**0**	**0**

TABLE 4
Comparisons of the Four Methods under Case 4 in terms of the *QoE* Metric in [53] (The Best Results Are Bolded)

Bandwidth Variations	Methods	*QoE* of *User1*	*QoE* of *User2*	*QoE* of *User3*	Average *QoE*
Fixed Bandwidth	*BF*	196.40	172.90	167.00	178.77
	QF	80.12	71.72	33.32	61.72
	QBA	363.10	333.30	286.70	327.70
	Proposed	**396.30**	**348.70**	**311.10**	**352.03**
Bandwidth with Persistent Variation	*BF*	234.70	167.00	113.10	171.60
	QF	151.78	61.28	28.58	80.55
	QBA	370.40	**348.50**	280.70	333.20
	Proposed	**405.70**	347.40	**325.80**	**359.63**
Bandwidth with Staged Variation	*BF*	212.60	172.90	161.30	182.27
	QF	163.18	135.68	98.58	132.48
	QBA	363.60	**339.30**	**289.20**	**330.70**
	Proposed	**374.40**	327.80	288.90	330.37
Bandwidth with Short-term Variation	*BF*	278.20	178.70	118.00	191.63
	QF	238.30	154.70	84.10	159.03
	QBA	375.90	**348.00**	303.20	342.37
	Proposed	**389.70**	343.80	**321.10**	**351.53**

TABLE 5
Comparisons of the Four Methods under Case 4 in terms of the *QoE* Metric in [54] (The Best Results are Bolded)

Bandwidth Variations	Methods	*QoE* of *User1*	*QoE* of *User2*	*QoE* of *User3*	Average *QoE*
Fixed Bandwidth	*BF*	181.75	177.65	167.45	175.62
	QF	113.95	107.87	99.55	107.12
	QBA	**197.39**	192.79	161.23	183.80
	Proposed	197.28	**193.51**	**186.98**	**192.59**
Bandwidth with Persistent Variation	*BF*	183.54	176.64	165.83	175.34
	QF	134.62	120.89	109.93	121.81
	QBA	**197.42**	192.69	161.62	183.91
	Proposed	196.41	**192.90**	**186.15**	**191.82**
Bandwidth with Staged Variation	*BF*	180.93	177.85	166.50	175.09
	QF	136.39	129.23	122.04	129.22
	QBA	**197.24**	**192.70**	160.91	183.62
	Proposed	195.76	191.08	**186.34**	**191.06**
Bandwidth with Short-term Variation	*BF*	182.10	179.28	165.01	175.46
	QF	156.81	132.90	128.42	139.38
	QBA	**197.32**	192.61	161.37	183.77
	Proposed	197.25	**193.34**	**187.17**	**192.59**

The performance of the proposed algorithm was verified under 2 cases:

Case 1, 6 users requested different video contents all the time;

Case 2, 6 users requested different video contents at the beginning, and then some users leaved or joined the network.

5.1 Results of Case 1

The results of the wireless and wired networks are given in Figs. 16 and 17, respectively. We can observe that the fluctuations of the requested bitrates and buffer lengths are larger than those of the simulated networking scenario because of the complexity of the realistic networking environment. From Table 6, we can observe that the average initial playout delays of the *QF* and the *BF* methods are similar, i.e., 3.61 s and 2.70 s, for the wireless networking environment and 0.88 s and 1.47 s for the wired networking environment. Similar to the results of the simulation, the average received bitrates of the *Proposed* method are not the largest but are comparable with the other methods. More importantly, we can see that there is no playback interruption under both the wireless and wired networking environment for the proposed algorithm, which is also demonstrated in Figs. 16b and 17b. As shown in Table 6, although there is also no playback interruption for the *BF* method, the received bitrate fluctuations (average standard deviation) of all the 6 users are much larger than the *Proposed* method. From Figs. 16a and 17a, we can see that the requested bitrates of the *Proposed* method fluctuate around 1 Mbps (the total server export bandwidth is set as 6 Mbps). Moreover, the average buffer length of the *Proposed* method is the largest, which means that the performance of the *Proposed* method can cope with network variations better.

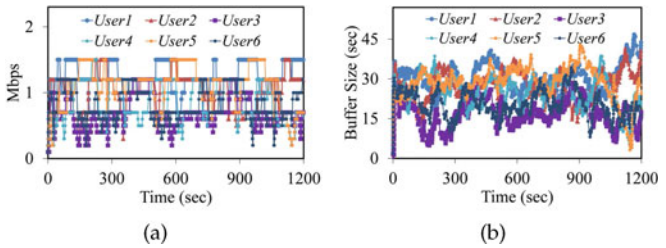


Fig. 16. Experimental Results of realistically modeled wireless network, in which six users compete the server export bandwidth (6 Mbps) with $\theta = 40$, $\mu = 0.003$, and $\nu = 0.0041$, (a) dynamic behavior of requested bitrates, and (b) the actual buffer lengths of the six users with the reference buffer length of 20 s.

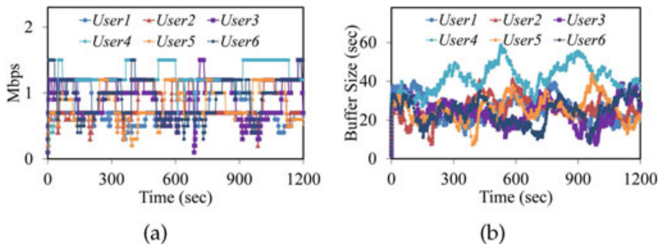


Fig. 17. Experimental Results of realistically modeled wired network, in which six users compete the server export bandwidth (6 Mbps) with $\theta = 30$, $\mu = 0.003$, and $\nu = 0.0041$, (a) dynamic behavior of requested bitrates, and (b) the actual buffer lengths of the six users with the reference buffer length of 20 s.

Because DASH achieves lossless transmission to improve the *QoE* of users at the cost of transmitting additional signaling information of TCP for HTTP sessions between the server and user, the overhead problem is common and inevitable in DASH. The signaling overhead is essentially determined by the number of HTTP sessions. Therefore, we also investigated the signaling overhead of the information exchanges between servers and users in the proposed algorithm, as shown in Table 7. We have to clarify that the reported signaling overhead in our manuscript is calculated as the *time ratio of the download time of overhead information to that of the video segments*. It can be observed that the average proportion of the signaling overhead induced by the information

TABLE 6
Performance Comparisons of Different Methods

Connection	Methods	Average Initial playout delays (sec)	Average Bitrate (Mbps)	Standard Deviation of Bitrate	Average Buffer Length (sec)	Average Number of Interruptions	Average Interruption Length (sec)
Wireless	*BF*	3.61s	0.96	**1.20**	20.41	**0.00**	**0.00**
	QF	2.70s	**1.01**	**1.02**	4.35	33.00	3.83
	Proposed	**2.47s**	0.90	**0.30**	**24.64**	**0.00**	**0.00**
Wired	*BF*	**0.88s**	0.90	**1.10**	19.68	**1.00**	0.57
	QF	1.47s	**0.93**	0.95	4.05	41.00	4.74
	Proposed	0.93s	0.88	**0.25**	**27.78**	**0.00**	**0.00**

exchanges is about 30 percent of the whole download time. *It is worth pointing out that by using additional HTTP sessions, the performance of a DASH system can be improved.*

5.2 Results of Case 2

We verified the performance of the proposed algorithm when users' requests dynamically come and leave at different times (i.e., *User5* leaves at about 600 s and joins at 900 s, *User6* leaves at about 300s and joins at 1200 s). Experimental results are shown in Figs. 18 (wireless) and 19 (wired). From Figs. 18a and 19a, we can observe that when *User* 5 and 6 leave, the requested bitrates of the remaining users increase to 1.5 Mbps gradually, while the requested video bitrates of all the users gradually converge to 1Mbps when *User* 5 and 6 join again. Accordingly, the effectiveness of the proposed algorithm is demonstrated for the realistic network scenario with dynamic user leaving or joining.

6 Conclusion and Discussion

In this paper, we have presented a novel non-cooperative game theory based algorithm to address the rate adaptation issue posed in a DASH system with single-server and multi-users. The proposed algorithm can not only guarantee user fairness but also improve user *QoE*. Moreover, no proxy is required with our algorithm. We have formulated the rate adaptation problem as a non-cooperative game by building a novel user *QoE* model that considers the received video quality, reference buffer length, and accumulated buffer lengths of users. We have theoretically proven the existence of the Nash Equilibrium of our specific game, which can be found by our distributed iterative algorithm with stability analysis. Simulation and experimental results show that the quality and bitrates of received videos by the proposed algorithm are more stable than the state-of-the-art methods, while the actual buffer length of each user moves around the reference buffer all the time. Besides, there is no playback interruption for the proposed algorithm.

Although the proposed rate adaptation algorithm can achieve impressive performance compared with existing algorithm, we believe it can be further improved by addressing the following limitations:

(1) Restriction on the *QoE* model. In order to guarantee the existence of the Nash Equilibrium, the *QoE* model must be designed as a continuous and quasi-concave function with respect to the bitrates of all the users.

(2) Stability analysis of the scenarios with multi-users or users joining and leaving dynamically. In the proposed method, we used a distributed iterative algorithm to obtain the Nash Equilibrium. However, the stability of the distributed iterative algorithm is analyzed in detail for the scenario with only 2 users. When there are more than 2 users, the stability analysis will be very complex, and we only provide

TABLE 7
The Proportion of Signaling Overhead Introduced by the Information Exchanges between Server and Users

Connection	*User1*	*User2*	*User3*	*User4*	*User5*	*User6*	Average
Wireless	24%	27%	29%	32%	25%	28%	28%
Wired	34%	30%	30%	26%	28%	26%	29%

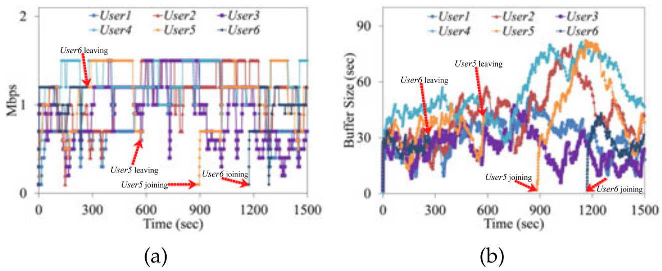


Fig. 18. Experimental Results of realistically modeled wireless network, in which users' requests come and leave at different time (*User5* leaving at about 600 s and joining at 900 s, *User6* leaving at about 300 s and joining at 1200 s), with $\theta = 40$, $\mu = 0.003$, and $\nu = 0.0041$, and the server export bandwidth is 6Mbps, (a) dynamic behavior of requested bitrates, and (b) the actual buffer lengths of the six users with the reference buffer length of 20 s.

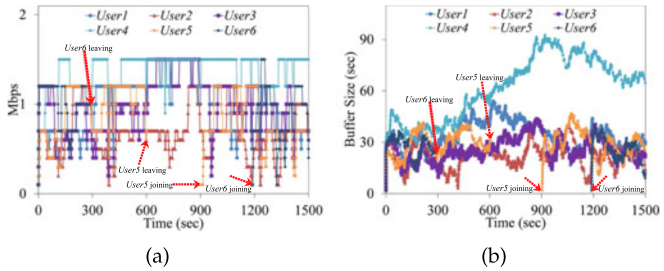


Fig. 19. Experimental Results of realistically modeled wired network, in which users' requests come and leave at different time (*User5* leaving at about 600 s and joining at 900 s, *User6* leaving at about 300 s and joining at 1200 s), with $\theta = 40$, $\mu = 0.003$, and $\nu = 0.0041$, and the server export bandwidth is 6 Mbps, (a) dynamic behavior of requested bitrates, and (b) the actual buffer lengths of the six users with the reference buffer length of 20 s.

a sketch. Besides, for the scenario with new users joining/leaving dynamically, the stability analysis of the proposed method is not analyzed well.

(3) Additional HTTP sessions. For the proposed method, additional HTTP sessions between users and servers are needed to achieve the Nash Equilibrium, which will introduce additional signaling overhead. Such signaling overhead may result in latency. Although this is a common drawback of DASH, it is also highly desirable to develop new algorithms to reduce the additional HTTP sessions as much as possible.

Acknowledgments

The authors would like to thank the Institute of Information Technology (ITEC) at Klagenfurt University for the valuable and basis work of DASH. They would also like to thank the editors and anonymous reviewers for their valuable comments. This work was supported in part by the National Natural Science Foundation of China under Grants 61571274,61672443; in part by the Shandong Natural Science Funds for Distinguished Young Scholar under Grant JQ201614; in part by Hong Kong RGC General Research Fund(GRF) under Grant 9042322 (CityU 11200116); and in part by the Young Scholars Program of Shandong University (YSPSDU) under Grant 2015WLJH39.

References

[1] C. Begen, T. Akgul, and M. Baugher, "Watching video over the web, Part I: Streaming protocols," *IEEE Int. Comput.*, vol. 15, no. 2, pp. 54–63, Mar. 2011.
[2] A. C. Begen, T. Akgul, and M. Baugher, "Applications, standardization, and open issues," *IEEE Int. Comput.*, vol. 15, no. 3, pp. 59–63, Apr. 2011.
[3] ISO/IEC IS 23009–1: "Information technology—Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation Description and Segment Formats," May 2014.
[4] T. Stockhammer, "Dynamic adaptive streaming over HTTP - standards and design principles," in *Proc. ACM Multimedia Syst. Conf.*, Feb. 2011, pp. 133–144.
[5] D. K. Krishnappa, D. Bhat, and M. Zink, "DASHing YouTube: An analysis of using DASH in YouTube video service," in *Proc. IEEE Conf. Local Comput. Netw.*, 2013, pp. 407–415.
[6] C. Wang, A. Rizk, and M. Zink, "SQUAD: A spectrum-based quality adaptation for dynamic adaptive streaming over HTTP," in *Proc. Int. Conf. Multimedia Syst.*, May. 2016, Art. no. 1.
[7] J. Nash, "Non-cooperative games," *Ann. Math.*, pp. 286–295, vol. 54, 1951.
[8] J. Luo, I. Ahmad, and Y. Sun, "Controlling the bit rate of multi-object videos with noncooperative game theory," *IEEE Trans. Multimedia*, vol. 12, no. 2, pp. 97–107, Feb. 2010.
[9] P. Goudarzi, "A non-cooperative quality optimization game for scalable video delivery over MANETs," *Wireless Netw.*, vol. 19, no. 5, pp. 755–770, 2013.
[10] A. Cournot, *Researches into the Mathematical Principles of the Theory of Wealth*, A. M. Kelley, New York, 1927.
[11] T. C. Thang, Q. D. Ho, J. W. Kang, and A. T. Pham, "Adaptive streaming of audiovisual content using MPEG DASH," *IEEE Trans. Consum. Electron.*, vol. 58, no. 1, pp. 78–85, Feb. 2012.
[12] L. R. Romero, "A dynamic adaptive HTTP streaming video service for google android," M.S. Thesis, School Inf. Commun. Technol., Royal Inst. Technol. (KTH), Stockholm, Sweden, Oct. 2011.
[13] T. C. Thang, H. T. Le, H. X. Nguyen, A. T. Pham, J. W. Kang, and Y. M. Ro, "Adaptive Video Streaming over HTTP with Dynamic Resource Estimation," *Commun. and Netw.*, vol. 15, no. 6, pp. 635–644, Dec. 2013.
[14] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj, "Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network," *Signal Process. Image Commun.*, vol. 27, no. 4, pp. 288–311, 2012.
[15] 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.*, Nov. 2012, pp. 225–238.
[16] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with Pensieve," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 21, pp. 197–210.
[17] M. Li, C.-Y. Lee, "A cost-effective and real-time QoE evaluation method for multimedia streaming services," *Telecommun. Syst.*, vol. 59, no. 3, pp. 317–327, Jul. 2015.
[18] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, "A quality-of-experience index for streaming video," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 1, pp. 154–166, Feb. 2017.
[19] 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.
[20] A. Parandeh Gheibi, M. Medard, A. Ozdaglar, and S. Shakkottai, "Avoiding interruptions—A QoE reliability function for streaming media applications," *IEEE J. Sel. Area Commun.*, vol. 29, no. 5, pp. 1064–1074, May. 2011.
[21] Y. Xu, Y. Zhou, and D. M. Chiu, "Analytical QoE models for bit-rate switching in dynamic adaptive streaming systems," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2734–2748, Dec. 2014.
[22] D. Zegarra Rodríguez, R. Lopes Rosa, E. Costa Alfaia, J. Issy Abrahão, and G. Bressan, "Video quality metric for streaming service using DASH standard," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 628–639, Sep. 2016.
[23] C. Zhou, C. W. Lin, and Z. Guo, "mDASH: A markov decision-based rate adaptation approach for dynamic HTTP streaming," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 738–751, Apr. 2016.
[24] V. Martín, J. Cabrera, and N. García, "Design, optimization and evaluation of A Q-Learning HTTP adaptive streaming client," *IEEE Trans. Consum. Electron.*, vol. 62, no. 4, pp. 380–388, Nov. 2016.
[25] A. Bokani, M. Hassan, S. Kanhere, and X. Zhu, "Optimizing HTTP-based adaptive streaming in vehicular environment using markov decision process," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2297–2309, Dec. 2015.
[26] 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, no. 1, pp. 326–340, Feb. 2014.
[27] Z. Li, et al., "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE J. Select. Areas Commun.*, vol. 32, no. 4, pp. 719–733, Apr. 2014.
[28] A. E. Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-based traffic and resource management for adaptive HTTP video delivery in LTE," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 6, pp. 988–1001, Jun. 2015.
[29] Z. Su, Q. Xu, M. Fei, and M. Dong, "Game theoretic resource allocation in media cloud with mobile social users," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1650–1660, Aug. 2016.
[30] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "SSIM-motivated rate-distortion optimization for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 516–529, Apr. 2012.
[31] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in HTTP adaptive streaming," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2386–2399, Aug. 2016.
[32] P. D. Straffin, *Game theory and strategy*, Washington, DC, USA: Mathematical Association of America, 1993.
[33] S. Boyd and L. Vandenberghe, *Convex Optimization*, 6th ed. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 67–71.
[34] S. Boyd, X. Lin, and A. Mutapcic, *Subgradient Methods*, Stanford, CA, USA: Stanford Univ., Oct. 2003.
[35] P. Tsiaflakis, I. Necoara, J. A. K. Suykens, and M. Moonen, "Improved dual decomposition based optimization for DSL dynamic spectrum management," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2230–2245, Apr. 2010.
[36] Z. Zhou, M. Dong, K. Ota, J. Wu, and T. Sato, "Energy efficiency and spectral efficiency tradeoff in device-to-device (D2D) communications," *IEEE Wirel. Commun. Lett.*, vol. 3, no. 5, pp. 485–488, Oct. 2014.
[37] D. Niyato and E. Hossain, "Competitive spectrum sharing in cognitive radio networks: A dynamic game approach," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2651–2660, Jul. 2008.
[38] M. Sonis, "Once more on henon map: Analysis of bifurcations" *Chaos, Solitons Fractals*, vol. 7, no. 12, pp. 2215–2234, 1996.

[39] H. N. Agiza, G.-I. Bischi, and M. Kopel, "Multistability in a dynamic cournot game with three oligopolists," *Math. Comput. Simul.*, vol. 51, pp. 63–90, 1999.

[40] E. Ahmed and H. N. Agiza, "Dynamics of a cournot game with n competitors," *Chaos, Solitons Fractals*, vol. 9, pp. 1513–1517, 1998.

[41] C. Mueller, S. Lederer, J. Poecher, and C. Timmerer, "Demo paper: Libdash—An open source software library for the MPEG-DASH standard," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2013, pp. 1–2.

[42] C. Alberti et al., "Automated QoE evaluation of dynamic adaptive streaming over HTTP," in *Proc. IEEE 5th Int. Workshop Quality Multimedia Experience*, 2013, pp. 58–63.

[43] Apache HTTP Server Project, [Online]. Available: http://httpd.apache.org/, Access on: Jan. 2017.

[44] L. Rizzo, "Dummynet: A simple approach to the evaluation of network protocols," *ACM Comput. Commun. Rev.*, vol. 27, no. 1, pp. 31–41, Jan. 1997.

[45] Big Buck Bunny Movie, [Online]. Available: http://www.bigbuckbunny.org/, Access on: Jan. 2017.

[46] C. Mueller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," in *Proc. 4th Annu. ACM SIGMM Workshop Mobile Video*, 2012, pp. 37–42.

[47] Elephants Dream Movie, [Online]. Available: http://www.elephantsdream.org/, Access on: Jan. 2017.

[48] Sita Sing the Blues Movie, [Online]. Available: http://www.sitasingstheblues.com/, Access on: Jan. 2017.

[49] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in *Proc. ACM Multimedia Syst. Conf.*, 2012, Feb. 22–24.

[50] S. Lederer, C. Mueller, C. Timmerer, C. Concolato, J. L. Feuvre, and K. Fliegel, "Distributed DASH dataset," in *Proc. 4th ACM Multimedia Syst. Conf.*, 2013, pp. 131–135.

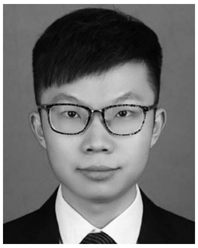
[51] M. Dräxler and H. Karl, "Cross-layer scheduling for multi-quality video streaming in cellular Wireless networks," in *Proc. Int. Wirel. Commun. Mobile Comput.*, 2013, pp. 1181–1186.

[52] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," *Signal Process. Image Commun.*, vol. 27, no. 4, pp. 271–287, 2012.

[53] X. Yin, V. Sekar, and B. Sinopoli, "Toward a principled framework to design dynamic adaptive streaming algorithms over HTTP," in *Proc. Workshop Hot Topics Netw.*, Sep. 2014.

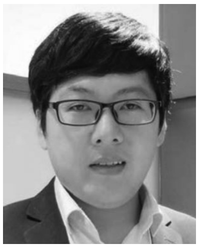
[54] F. Chiariotti, S. D'Aronco, L. Toni, and P. Frossard, "Online learning adaptation strategy for DASH clients," in *Proc. Multimedia Syst.*, ACM, Aug. 2016, Art. no. 8.

Hui Yuan (S'08-M'12-SM'17) received the BE and PhD degrees in telecommunication engineering from Xidian University, Xi'an, China, in 2006 and 2011, respectively. From April 2011 to now, he works as lecturer, associate professor, and full professor in the School of Information Science and Engineering, Shandong University (SDU), Jinan, China. From January 2013.– December 2014, he also worked as a post-doctor fellow in the department of computer science, City University of Hong Kong (CityU). His current research interests include video/image compression, adaptive video streaming, and computer vision, etc. He is a senior member of the IEEE.



Huayong Fu received the BE degree in telecommunication engineering from Shandong University, Jinan, China, in 2015. He is currently working toward the MS degree in information and communication engineering in the School of Information Science and Engineering, Shandong University, Jinan, China. His current research interests include video transmission and multimedia communication.

Ju Liu (M'02-SM'09) received the BS and MS degrees in electronics engineering from Shandong University (SDU), Jinan, China, in 1986 and 1989, respectively, and the PhD degree in signal processing from the Southeast University (SEU), Nanjing, China, in 2000. Since July 1989, he has been with the School of Information Science and Engineering, Shandong University. From July 2002 to December 2003, he was a visiting professor with the Department of Signal Theory and Communication, Polytechnic University of Catalonia and the Telecommunications Technological Center of Catalonia, Barcelona, Spain. From November 2005 to January 2006, he was a visiting researcher with the Department of Communications Engineering, University of Bremen, Bremen, Germany, and the Department of Communication Systems, University of Duisburg, Essen, Germany. From June to December, 2009, he was a senior research fellow with the Department of Electrical Engineering, University of Washington, Seattle. He is the author or a coauthor of more than 200 journal papers and conference proceedings. He is a member of editorial committee of the *Journal of Swarm Intelligence Research, the Journal of Communications*, and the *Journal of Circuits and Systems in China*. His research interests include space-time processing in wireless communication, blind signal separation, and multimedia communications. He received the Program for New Century Excellent Talents in University of China Award, three Conference Best Paper awards, and three national and local academic awards in blind signal processing. He has been an active Reviewer for the *IEEE Transactions on Signal Processing*, the *IEEE Signal Processing Letters*, and the *IEEE Communications Letters*. He is a senior member of the IEEE.



Junhui Hou (S'13-M'16) received the BEng degree in information engineering (Talented Students Program) from the South China University of Technology, Guangzhou, China, in 2009, the MEng degree in signal and information processing from Northwestern Polytechnical University, Xi'an, China, in 2012, and the PhD degree from the School of Electrical and Electronic Engineering, NTU, Singapore, in 2016. He is an assistant professor with the Department of Computer Science, City University of Hong Kong (CityU) since Jan 2017. His research interests fall into the general areas of multimedia signal processing, such as adaptive image/video representations and analysis (RGB/depth/light field/hyperspectral), static/dynamic 3D geometry representations and processing (mesh/point cloud/MoCap), and graph signal processing. He is the receipt of the prestigious award of *Chinese Government for Outstanding Self-financed Students Abroad* in 2015. He is a member of the IEEE.

Sam Kwong (M'93-SM'04-F'13) received the BS degree from the State University of New York at Buffalo, in 1983, the MS degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1985, and the PhD degree from the University of Hagen, Germany, in 1996. From 1985 to 1987, he was a diagnostic engineer with Control Data Canada, Mississauga, ON, Canada. He joined Bell Northern Research Canada, Ottawa, ON, Canada, as a member of Scientific Staff. In 1990, he became a lecturer with the Department of Electronic Engineering, The City University of Hong Kong, where he is currently a professor with the Department of Computer Science. His research interests are video and image coding, and evolutionary algorithms. He serves as an associate editor of *IEEE Transactions on Industrial Electronics* and the *IEEE Transactions on Industrial Informatics*. He is a fellow of the IEEE.