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 nearby 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) 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 II). 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 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 inter-
action 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 II, the related work on rate adaptation methods for DASH is presented. In Section III, 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 Section IV and V, respectively. Finally, Section VI 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 of users directly, some researchers proposed dynamic bitrate selection methods based on QoE maximization [19-22]. 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 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, bit rate switching frequency and amplitude, buffer length, etc. are considered comprehensively. Similarly, Martín et al. [24] proposed a Q-Learning-based bit rate 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-adopt bandwidth estimation approach to further increase the bandwidth utilization and stabilize the requested video bitrates of multi-users. Essa’lli 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 \( i \)-th user). The payoff or profit for the \( i \)-th 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 \( G = \{ I, \left\{ R_i \right\}, \left\{ U_i(\cdot) \right\} \} \) denote the non-cooperative rate adaptation game where \( I = \{ 1, 2, \cdots, N \} \) is the index set for the users in the DASH system, and \( R_i \) and \( U_i(\cdot) \) are the strategy space and utility function of the \( i \)-th user, respectively. Each user determines the required bitrate \( r_i \) such that \( r_i \in R_i \). Let the rate vector \( r = \{ r_1, \cdots, r_i, \cdots, r_N \} \) denote the outcome of the game in terms of the requested bitrates of all users. The resulting utility for the \( i \)-th user is \( U_i(r) \). We will occasionally use \( U_i(r_i, r_{-i}) \) to replace \( U_i(r) \) to indicate the dependence among DASH users, where \( r_{-i} \) denotes the vector that consists of elements of \( r \) without the \( i \)-th element, i.e., \( r_{-i} = \{ r_1, \cdots, r_{i-1}, r_{i+1}, \cdots, r_N \} \).

In the following subsections, 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.,

\[
q_i(r_i) = \alpha_i \log(1 + \beta_i r_i),
\]

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 \( \alpha_i \) and \( \beta_i \) 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 \( i \)-th 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.,
\[
\Delta b_i(r_i) = T - T \cdot r_i / B_W,
\]
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 \(B_W\) is the available channel bandwidth of the \(i\)-th 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 downloaded 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
\[
\Delta b(r) = T - \omega \cdot T \cdot \sum_{i=1}^{N} r_i / B_W,
\]
where \(\omega\) 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 \(i\)-th user considering other users denoted as \(b^e_{est}(r_i, r_{-i})\) can be derived by integrating \(\Delta b(r)\) over \(r_i\):
\[
b^e_{est}(r_i, r_{-i}) = \int \Delta b(r)dr_i = T \cdot r_i - \omega \cdot T \cdot \left( \frac{1}{2} r_i^2 + r_i \sum_{j=1}^{N} r_j \right) / B_W + b_0 = \Phi(r_i) - \omega \cdot \Psi(r) + b_0,
\]
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 \(\Psi(r) = T \cdot \left( \frac{1}{2} r_i^2 + r_i \sum_{j=1}^{N} r_j \right) / B_W\) 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.,
\[
\Phi'(r_i) = A_f \cdot \Phi(r_i) = A_f \cdot T \cdot r_i.
\]
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 \(b_{ref}\):
\[
A_f = 2 \cdot \frac{e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}} = 2 \cdot \frac{e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}} \cdot T \cdot r_i - \omega \cdot T \cdot \left( \frac{1}{2} r_i^2 + r_i \sum_{j=1}^{N} r_j \right) / B_W + b_0.
\]
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.,
\[
U_i(r_i, r_{-i}) = q_i(r_i) + \mu \cdot b^e_{est}(r_i, r_{-i}) = \alpha_i \log(1 + \beta_i r_i) + \mu \cdot \frac{2 \cdot e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}} \cdot T \cdot r_i - \mu \cdot \omega \cdot T \cdot \left( \frac{1}{2} r_i^2 + r_i \sum_{j=1}^{N} r_j \right) / B_W + \mu \cdot b_0 = \alpha_i \log(1 + \beta_i r_i) + \mu \cdot \frac{2 \cdot e^{p(b_{curr} - b_{ref})}}{1 + e^{p(b_{curr} - b_{ref})}} \cdot T \cdot r_i - \mu \cdot \omega \cdot T \cdot \left( \frac{1}{2} r_i^2 + r_i \sum_{j=1}^{N} r_j \right) / B_W + \mu \cdot b_0,
\]
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 \( R_i \) is a non-empty, convex, and compact subset of Euclidean space \( \mathbb{R}^N \);

b) the utility \( U_i(r) \) is continuous in \( r \) 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 \( i \)-th user. Thereby, there is no doubt that \( R_i \) is a non-empty and compact subset of the Euclidean space \( \mathbb{R}^N \). According to the definition of the convex set \([33]\), for any \( r_x, r_y \in R_i \) and any \( \zeta \) with \( 0 \leq \zeta \leq 1 \), we have \( 0 \leq \zeta r_x \leq r^{\max} \) and \( 0 \leq (1 - \zeta) r_y \leq (1 - \zeta) r^{\max} \). Then, we can get \( 0 \leq \zeta r_x + (1 - \zeta) r_y \leq r^{\max} \). Therefore, \( \zeta r_x + (1 - \zeta) r_y \in R_i \), indicating \( R_i \) is a convex set. Thus, condition a) is satisfied. For the utility function in \([3]\), it is obviously a continuous function in terms of \( r \). Besides, the second derivatives of \( U_i(r) \) with respect to all the \( r_i \) are

\[
\begin{align*}
\frac{\partial^2 U_i(r)}{\partial r_i^2} & = -\frac{a \beta^2}{(1 + \beta r_i)^2} - \frac{\nu T}{B_W}, \\
\frac{\partial^2 U_i(r)}{\partial r_j \partial r_i} & = -\frac{\nu T}{B_W} \iota_{ij},
\end{align*}
\]

which are negative because \( a, \beta, \nu, T, \) and \( B_W \) are non-negative. Accordingly, the utility \( U_i(r) \) 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:

\[
B_i(r_{-i}) = \arg \max_{r_i \in R_i} U_i(r_i, r_{-i}).
\]

When the Nash Equilibrium is achieved, the strategies of all players can be represented as \( r^* = \{r_1^*, r_2^*, \ldots, r_N^*\} \), where \( r_i^* = B_i(r_{-i}^*) \) is the optimal strategy of the \( i \)-th user and \( r_{-i}^* = \{r_1^*, \ldots, r_{i-1}^*, r_{i+1}^*, \ldots, r_N^*\} \) 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:

\[
\frac{\partial U_i(r)}{\partial r_i} = \frac{\alpha_i \beta_i}{1 + \beta_i r_i} + \mu \cdot T \cdot \frac{2 \rho(b_{\text{curr}} - b_{\text{ref}})}{1 + \rho(b_{\text{curr}} - b_{\text{ref}})} - \nu \cdot T \cdot \sum_{j=1}^N \frac{r_j}{B_W} = 0, \quad \forall i.
\]

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

\[
\begin{align*}
Z_{1,1} + Z_{1,2} - Z_3(r_1 + r_2) &= 0, \\
Z_{2,1} + Z_{2,2} - Z_3(r_1 + r_2) &= 0,
\end{align*}
\]

where

\[
\begin{align*}
Z_{1,1} &= \alpha_1 \beta_1, \\
Z_{1,2} &= \alpha_2 \beta_2, \\
Z_{2,1} &= \alpha_1 \beta_1, \\
Z_{2,2} &= \alpha_2 \beta_2, \\
Z_3 &= \nu \cdot T / B_W.
\end{align*}
\]

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

\[
r_1^* = r_2^* = \frac{-2(Z_3 - \beta Z_2) + \sqrt{(2Z_3 + \beta Z_2)^2 + 8 \beta Z_1 Z_3}}{4 \beta Z_3}. \tag{13}
\]

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 for the \( i \)-th 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 \([34][35][36]\):

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

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

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

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

with

\[
\begin{align*}
&U_i^+(r^+ \iota_j) = \max \{r_1(t), \ldots, r_i(t) + \varepsilon, \ldots, r_N(t)\}, \\
&U_i^-(r^- \iota_j) = \min \{r_1(t), \ldots, r_i(t) - \varepsilon, \ldots, r_N(t)\},
\end{align*}
\]

where \( \varepsilon \) 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.,

\[
\begin{align*}
r(t + 1) &= r(t), \\
\frac{\partial U_i(r)}{\partial r_i(t)} &= 0.
\end{align*}
\]

### 3.4 Stability Analysis for the Distributed Iterative Algorithm

The stability of the distributed strategy update algorithm \([15]\) is analyzed by using the Routh-Hurwitz stability condition \([37][38][39][40]\), which judges the distribution of the eigenvalues (denoted as \( \lambda_i \)) 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,

\[
J(r_1, r_2) = \begin{bmatrix}
\frac{\partial r_1(t + 1)}{\partial r_1(t)} & \frac{\partial r_1(t + 1)}{\partial r_2(t)} \\
\frac{\partial r_2(t + 1)}{\partial r_1(t)} & \frac{\partial r_2(t + 1)}{\partial r_2(t)}
\end{bmatrix} = \begin{bmatrix}
J_{1,1} & J_{1,2} \\
J_{2,1} & J_{2,2}
\end{bmatrix}, \tag{19}
\]
Furthermore, Eq. (25) can be simplified as
\[ j_{1,1} = -\theta_1 Z_1 r_1 \]
\[ j_{2,1} = -\theta_2 Z_2 r_2 \]
\[ j_{1,2} = 1 + \theta_1 \left( -\frac{\beta Z_1 r_1}{(1 + \beta r_1)} + \frac{Z_{1,1}}{1 + \beta r_1} + Z_{2,1} - Z_2 (2r_1 + r_2) \right) \]
\[ j_{2,2} = 1 + \theta_2 \left( -\frac{\beta Z_2 r_2}{(1 + \beta r_2)} + \frac{Z_{1,2}}{1 + \beta r_2} + Z_{2,2} - Z_2 (r_1 + 2r_2) \right) \]  

(20)
The two eigenvalues can be obtained by solving the characteristic equation:
\[ \lambda^2 - \lambda (j_{1,1} + j_{2,2}) + (j_{1,2}j_{2,1} - j_{1,1}j_{2,2}) = 0, \]
whose solution is
\[ \lambda_1 = \frac{(j_{1,1} + j_{2,2}) + \sqrt{(j_{1,1} - j_{2,2})^2 + 4j_{1,2}j_{2,1}}}{2} \]
\[ \lambda_2 = \frac{(j_{1,1} + j_{2,2}) - \sqrt{(j_{1,1} - j_{2,2})^2 + 4j_{1,2}j_{2,1}}}{2} \]  

(22)
Assume the two users are identical (i.e., \( Z_{1,1} = Z_{1,2} = Z_1 = \alpha \beta \) and \( Z_{2,1} = Z_{2,2} = Z_2 = \mu \)), 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., \( b_{\text{curr}} = b_{\text{ref}} \) 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
\[ -1 < j_{1,1} - j_{1,2} < 1, \]
\[ -1 < j_{1,1} + j_{1,2} < 1. \]  

(23)
For \( |j_{1,1} - j_{1,2}| < 1 \), substituting Eq. (20) into (23), we have
\[ -2 < \theta_1 \left[ -\frac{\beta Z_1 r_1}{(1 + \beta r_1)^2} + \frac{Z_{1,1}}{1 + \beta r_1} + Z_2 - Z_3 (2r_1 + r_2) \right] \]
+ \( \theta_1 Z_3 r_1 < 0 \).  

(24)
At the Nash Equilibrium point, since \( \theta_1 = \theta_2 = \theta^* \), and \( r_1 = r_2 = r^* \), Eq. (24) can be rewritten as
\[ -\frac{2}{\theta^*} - \frac{\beta Z_1 r^*}{(1 + \beta r^*)^2} + \frac{Z_{1,1}}{1 + \beta r^*} + Z_2 - 2Z_3 r^* < 0. \]  

(25)
Furthermore, Eq. (25) can be simplified as
\[ \left\{ \begin{array}{l} Z_1 + Z_2 (1 + \beta r^*) < 2Z_3 r^* (1 + \beta r^*), \\
Z_1 + (Z_2 + \frac{\theta^*}{2}) (1 + \beta r^*) > 2Z_3 r^* (1 + \beta r^*) \end{array} \right. \]  

(26)
Substituting \( T = 2, Z_1, Z_2, \) and \( Z_3 \) into (26), we can obtain
\[ \alpha \beta + 2 \mu (1 + \beta r^*) < \frac{\theta^*}{2} \frac{Z_1}{1 + \beta r^*} (1 + \beta r^*), \]
\[ \alpha \beta + (2 \mu + \frac{\theta^*}{2}) (1 + \beta r^*) > \frac{\theta^*}{2} \frac{Z_1}{1 + \beta r^*} (1 + \beta r^*). \]  

(27)
Similarly, for \( |j_{1,1} + j_{1,2}| < 1 \), the stability condition is,
\[ \left\{ \begin{array}{l} \alpha \beta + 2 \mu (1 + \beta r^*) < \frac{\theta^*}{2} \frac{Z_1}{1 + \beta r^*} (1 + \beta r^*), \\
\alpha \beta + (2 \mu + \frac{\theta^*}{2}) (1 + \beta r^*) > \frac{\theta^*}{2} \frac{Z_1}{1 + \beta r^*} (1 + \beta r^*) \end{array} \right. \]  

(28)
Since \( \alpha, \beta, \mu, \nu, \theta^* \), and \( B_W \) are all positive, we can conclude that the proposed distributed iterative updating algorithm is stable if the following conditions are satisfied:
\[ \left\{ \begin{array}{l} \alpha \beta + 2 \mu (1 + \beta r^*) < \frac{\theta^*}{2} \frac{Z_1}{1 + \beta r^*} (1 + \beta r^*), \\
\alpha \beta + (2 \mu + \frac{\theta^*}{2}) (1 + \beta r^*) > \frac{\theta^*}{2} \frac{Z_1}{1 + \beta r^*} (1 + \beta r^*) \end{array} \right. \]  

(29)
When there are more than two users in the system, \( Z_{1,i} \)
must be satisfied for user \( i \),
\[ \frac{Z_{1,i}}{1 + \beta r_i} + \frac{Z_{2,i} - Z_3 \left( \sum_{j=1}^{N} r_j \right)}{1 + \beta r_i} = 0, \]  

(30)
Equation (30) of all users can be expressed by matrix format as
\[ Z_1 + (1 + \beta r) Z_2 = Z_3 (r \cdot 1^T) (1 + r \cdot \beta), \]  

(31)
where
\[ r = [r_1 \cdots r_i \cdots r_N]^T, \]
\[ 1 = [1 \cdots 1 \cdots 1]^T, \]
\[ Z_1 = [Z_{1,1} \cdots Z_{1,i} \cdots Z_{1,N}]^T, \]
\[ Z_2 = \text{diag}(\beta_1, \cdots, \beta_N), \]
\[ Z_2 = \text{diag}(Z_{2,1}, \cdots, Z_{2,N}). \]  

The Jacobian matrix of the Nash Equilibrium for multi-users is given as
\[ J = \begin{bmatrix} \frac{\partial r_1(t+1)}{r_1(t)} & \cdots & \frac{\partial r_1(t+1)}{r_N(t)} \\
\frac{\partial r_2(t+1)}{r_1(t)} & \cdots & \frac{\partial r_2(t+1)}{r_N(t)} \\
\vdots & \ddots & \vdots \\
\frac{\partial r_N(t+1)}{r_1(t)} & \cdots & \frac{\partial r_N(t+1)}{r_N(t)} \end{bmatrix}. \]  

(33)
Then, similar to the DASH system with only 2 users, the local stability condition can also be analyzed.

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

**Algorithm 1** Distributed Iterative Algorithm of Rate Adaptation

1: Initially, all users request the bitrate \( r(1) = 0.1 \text{Mbps} \) to the server so as to quickly establish the predefined initial buffer length (e.g., 2s).
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 \( b_{\text{curr}}(n) \);
8: \( n = n + 1; \)
9: end while

**4 Simulation Results**

**4.1 Simulation Setup**

To verify the performance of the proposed method, the Lib-DASH 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 SitaSings-theBlues [48][49]. Each video was encoded by FFmpeg [50] with 20 various bitrates from low to high, as shown in TABLE I. Fig. 3 shows the two parameters of the quality model in (1) (i.e., \( \alpha \) and \( \beta \)) for each video. The parameters \( \alpha \) and \( \beta \) 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.

**TABLE 1**

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

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;

**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) [25].

**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 15s. 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 15-20s 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. 5(b).

**TABLE 2**

| Learning Rate $\theta$ | 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     |
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 [32], 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 $t = 50s$. When the server export bandwidth varies to 9 Mbps at the time of 100s, the requested bitrate with $\theta = 100$ quickly increases to 4.5 Mbps, while the requested bitrate with $\theta = 50$ firstly increases to 5 Mbps, which consumes about 5s 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, while the requested bitrate with $\theta = 50$ drops to 3.5 Mbps. When the available bandwidth increases back to 9 Mbps at the time of 300s, the requested bitrate with $\theta = 50$ drops 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 100s, 180s, 260s, and 340s). 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 100s and 260s, as shown in Fig. 10(a). 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. 10(b).

4.4 Results of Case 3

This subsection presents the performance of the proposed algorithm under Case 3, in which three users request three 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. From Fig. 11 we can observe that the Nash Equilibrium...
is achieved with \( r_1^* = 1.5 \text{ Mbps}, r_2^* = 1.5 \text{ Mbps}, \) and \( r_3^* = 1.5 \text{ Mbps}, \) 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. 12(a)) in order to ensure the actual buffer length is close to the reference buffer length (see Fig. 12(b)). As shown in Figs. 12(c) and (d), 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 User1 and User2, as shown in Figs. 12(e) and (f). The reason is that the QBA method does not take the fairness of users into consideration too much. From Figs. 12(g) and (h), 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 15s). 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 and 15 respectively. Similar conclusions can be drawn, which consistently demonstrate 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 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...
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                                    |
| 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                                    |
| 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                          | 6.038                   | 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                                    |
| 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                                    |

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.

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.
used QoE models [53,54]:

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

\[ QoE_2 = \sum_{k=1}^{M} q[k] - \varphi \sum_{k=1}^{M-1} |q[k+1] - q[k]| - \sigma \sum_{k=1}^{M-1} \left( \max\{0, b_{ref} - b[k+1]\}\right)^2 \tag{35} \]

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 \( k^{th} \) requested video segment, \( q[k] \) is the corresponding SSIM value, \( T_{down}[k] \) is the download time of the \( k^{th} \) segment, and \( b[k] \) is the buffer length at the end time of the \( k^{th} \) segment and \( b_{ref} = 15s \). Note that we set \( \eta \) to 2 instead of 50 in [54] to ensure that the QoE values are positive. From TABLE 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 (2.4GHz, 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

---

**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.

**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.
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       |
| 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       |

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      | 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       |
| Persistent 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       |

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 6Mbps by DummyNet), the experiments were conducted at night. The initial buffer of each user was set as 20s in the experiments in order to calculate the initial playout delay. 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 left or joined the network.

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

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 BF methods are similar, i.e., 3.61s and 2.70s, for the wireless networking environment and 0.88s and 1.47s 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. [16(b) and 17(b). 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. [16(a)] and [17(a)], we can see that the requested bitrates of the Proposed method fluctuate around 1Mbps (the total server export bandwidth is set as 6Mbps). 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.

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 exchanges is about 30% 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 600s and joins at 900s, User6 leaves at about 300s and joins at 1200s). Experimental results are shown in Figs. 18 (wireless) and 19 (wired). From Figs. 18(a) and 19(a), we can observe that when User 5 and 6 leave, the requested bitrates of the remaining users increase to 1.5Mbps 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 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.

Acknowledgment

The authors would like to thank 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.

References

[1] C. Begon, T. Akgul, 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. Begon, T. Akgul, 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,” 2012.
[4] T. Stockhammer, “Dynamic Adaptive Streaming over HTTP - Standards and Design Principles,” in Proc. ACM Multimedia Systems Conference, California, Feb. 2013, 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 Computer Networks, Sydney, NSW, 2013, pp. 407-415.
[6] C. Wang, A. Rizk, M. Zink, “SQUAD: A Spectrum-Based Quality Adaptation for Dynamic Adaptive Streaming over HTTP,” in Proc. International Conference on Multimedia Systems, Klagenfurt, Austria, May 2016.
[7] J. Nash, “Non-cooperative games,” Annals of Mathematics, pp. 286-295, vol. 54, 1951.
[8] J. Luo, I. Ahmad and Y. Sun, “Controlling the Bit Rate of Multi-Object Videos with NoCooperative Game Theory,” IEEE Multimedia, vol. 12, no. 2, pp. 97-107, Feb. 2010.
[9] F. Goudarzi, “A Non-cooperative Quality Optimization Game for Scalable Video Delivery over MANETs,” Wireless Networks, vol. 19, no. 5, pp. 755-770, 2013.
[10] A. Cournot, “Researches into the Mathematical Principles of the Theory of Wealth,” NT Bacon Trans., 1927.
[11] S. Boyd, X. Lin and A. Mutapcic, “Subgradient Methods,” Convex Optimization, 2004.
[12] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begon and D. Oran, “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.
[13] 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.
[14] Z. Su, Q. Xu, M. Fei and M. Dong, “Game Theoretic Resource Allocation in Media Cloud with Mobile Social Users,” IEEE Multimedia, vol. 18, no. 8, pp. 1650-1660, Aug. 2016.
[15] 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.
[16] G. Tian and Y. Liu, “Towards Agile and Smooth Video Adaptation in HTTP Adaptive Streaming,” IEEE/ACM Trans. Networking, vol. 24, no. 4, pp. 2386-2399, Aug. 2016.
[17] R. D. Straffin, “Game Theory and Strategy,” The Mathematical Association of America, Washington, DC, 1993.
[18] S. Boyd and L. Vandenberghe, Convex Optimization, 6th ed. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 67-71.
[19] S. Boyd, X. Lin and A. Mutapcic, Subgradient Methods, Stanford University, Oct. 2003.
[20] 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. Letters, vol. 3, no. 5, pp. 485-488, Oct. 2014.
[21] C. Liu, I. Bouazizi, M. M. Hannuksela and M. Gabbouj, “Rate Adaptation for Dynamic Adaptive Streaming over HTTP in Content Distribution Network,” Signal Processor. Image Commun., vol. 27, no. 4, pp. 288-311, 2012.
[22] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, “A quality-of-experience index for streaming video,” IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 1, pp. 154-166, Feb. 2017.
[23] 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.
[24] A. Parandehe Gheibi, M. Medard, A. Ozdaglar, and S. Shakkottai, “Avoiding Intermittents - A QoE Reliability Function for Streaming Media Applications,” IEEE J. Sel. Area Commun., vol. 29, no. 5, pp. 1064-1074, May 2011.
[25] Y. Xu, Z. Zhou and D. M. Chiu, “Analytical QoE Models for Bit-Rate Switching in Dynamic Adaptive Streaming Systems,” IEEE Trans. Mobile Computing, vol. 13, no. 12, pp. 2734-2748, Dec. 2014.
[26] D. Zegarra Rodriguez, 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, Sept. 2016.
[27] V. Martin, J. Cabrera and N. García, “Design, Optimization and Evaluation of A Q-Learning HTTP Adaptive Streaming Client,” IEEE Trans. Consumer Electron., vol. 62, no. 4, pp. 380-388, Nov. 2016.
[28] 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.
[29] J. Jiang, V. Sekar and H. Zhang, “Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive,” IEEE/ACM Trans. Networking, vol. 22, no. 1, pp. 326-340, Feb. 2014.
[30] H. Zhang, M. Dong, Q. Xu, Y. Gao, “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.
[31] 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.
[32] Z. Su, Q. Xu, M. Fei and M. Dong, “Game Theoretic Resource Allocation in Media Cloud with Mobile Social Users,” IEEE Multimedia, vol. 18, no. 8, pp. 1650-1660, Aug. 2016.
[33] 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.
[34] Z. Tian and Y. Liu, “Towards Agile and Smooth Video Adaptation in HTTP Adaptive Streaming,” IEEE/ACM Trans. Networking, vol. 24, no. 4, pp. 2386-2399, Aug. 2016.
[35] P. D. Straffin, “Game Theory and Strategy,” The Mathematical Association of America, Washington, DC, 1993.
[36] S. Boyd and L. Vandenberghe, Convex Optimization, 6th ed. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 67-71.
[37] S. Boyd, X. Lin and A. Mutapcic, Subgradient Methods, Stanford University, Oct. 2003.
[38] Z. Su, Q. Xu, M. Fei and M. Dong, “Game Theoretic Resource Allocation in Media Cloud with Mobile Social Users,” IEEE Multimedia, vol. 18, no. 8, pp. 1650-1660, Aug. 2016.
[39] 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.
[40] Z. Tian and Y. Liu, “Towards Agile and Smooth Video Adaptation in HTTP Adaptive Streaming,” IEEE/ACM Trans. Networking, vol. 24, no. 4, pp. 2386-2399, Aug. 2016.
[43] Apache HTTP Server Project, http://httpd.apache.org/, (last access: Jan. 2017).
[44] L. Rizzo, “Dummynet: A Simple Approach to The Evaluation of Network Protocols,” ACM Computer Communication Review, vol. 27, no. 1, pp. 31-41, Jan. 1997.
[45] Big Buck Bunny Movie, http://www.bigbuckbunny.org/, (last access: Jan. 2017).
[46] C. Mueller, S. Lederer and C. Timmerer, “An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments,” in Proc. the Fourth Annual ACM SIGMM Workshop on Mobile Video, ACM, New York, NY, USA, 2012, pp. 37-42.
[47] Elephants Dream Movie, http://www.elephantsdream.org/, (last access: Jan. 2017).
[48] Sita Sing the Blues Movie, http://www.sitasingstheblues.com/, (last access: Jan. 2017).
[49] S. Lederer, C. Müller and C. Timmerer, “Dynamic Adaptive Streaming over HTTP Dataset,” in Proc. the ACM Multimedia Systems Conference 2012, Chapel Hill, North Carolina, 2012, Feb. 22-24.
[50] S. Lederer, C. Mueller, C. Timmerer, C. Concolato, J. L. Feuvre and K. Fliegel, “Distributed DASH Dataset,” in Proc. the 4th ACM Multimedia Systems Conference, ACM, New York, NY, USA, 2013, pp. 131-135.
[51] M. Drixler and H. Karl, “Cross-layer Scheduling for Multi-Quality Video Streaming in Cellular Wireless Networks,” in Proc. Int. Wirel. Commun. and Mobile Comput. (IWCMC), Sardinia, 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 Processing Image Communication, 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 on Hot Topics in Networks, ACM, Sept. 2014.
[54] F. Chiariotti, S. D’Aronco, L. Toni and P. Frossard, “Online Learning Adaptation Strategy for DASH Clients,” in Proc. Multimedia Systems, ACM, Aug. 2016.