IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 16, 2017, accepted December 20, 2017, date of publication December 29, 2017,
date of current version March 13, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2788057

A Bandwidth Variation Pattern-Differentiated Rate Adaptation for HTTP Adaptive Streaming Over an LTE Cellular Network

HAIPENG DU[1,2], QINGHUA ZHENG[1], (Member, IEEE), WEIZHAN ZHANG[1], (Member, IEEE), AND XIANG GAO[1]

[1]SPKLSTN Lab, Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China
[2]Research and Development, School of Continuing Education, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Weizhan Zhang (zhangwzh@xjtu.edu.cn)

This work was supported in part by the fundamental theory and applications of big data with knowledge engineering under the National Key Research and Development Program of China under Grant 2016YFB1000903, in part by the National Science Foundation of China under Grant 61772414, Grant 61532015, Grant 61532004, Grant 61721002, Grant 61472317, and Grant 61502379, in part by the MOE Innovation Research Team under Grant IRT17R86, in part by the Innovative Research Group of the National Natural Science Foundation of China under Grant 61721002, and in part by the Project of the China Knowledge Centre for Engineering Science and Technology.

ABSTRACT Currently, HTTP adaptive streaming (HAS) is the state-of-the-art technology for mobile video streaming. The rate adaptation of HAS has been designed to make a trade-off between two contrasting requirements, i.e., enhancing the quality of a video, and reducing the probability of video freezes, by adaptively switching between different video bitrates during a video playback session. This process becomes more challenging when moving onto an long term evolution (LTE) cellular network due to the unstable nature of the wireless channel. In this paper, we propose the bandwidth variation pattern-differentiated rate adaptation (BVPDRA) algorithm for LTE cellular networks. Unlike prior works, BVPDRA does not strike a balance between the stableness and responsiveness of bitrate switching in the case of bandwidth capacity variations. BVPDRA differentiates between bandwidth variation patterns of the LTE cellular network as either constant bandwidth fluctuations or instantaneous bandwidth hopping. Accordingly, BVPDRA operates with a dual character: for the constant bandwidth fluctuations, BVPDRA performs smoothed bandwidth prediction and conservative rate switching to minimize video quality version oscillations; for the instantaneous bandwidth hopping, BVPDRA performs positive bandwidth prediction and aggressive rate switching to maximize the bandwidth utilization and minimize the risk of playback stalling. We empirically evaluate the performance of BVPDRA on an LTE cellular network testbed. The results demonstrate that BVPDRA achieves a higher average bitrate, and lower rebuffering ratio with a reduced bitrate switching frequency.

INDEX TERMS HTTP adaptive streaming, long term evolution (LTE) cellular network, rate adaptation.

I. INTRODUCTION

With the maturing of long term evolution (LTE) mobile communication, the number of LTE subscriptions globally is forecast to rise from 1.73 billion (end of 2016) to 4.33 billion (end of 2021), at which time LTE will account for over 52% of all mobile subscriptions [1]. As a fundamental application in an LTE network [2], mobile video streaming is expected to generate 3/4 of the global mobile data traffic by 2019. Providing good quality of experience (QoE) [3] has always been a research focus. There is a trade-off between two contrasting requirements: enhancing the quality of a video and reducing the probability of video freezes. Watching a video encoded at a bitrate of 1000 kbps provides a better QoE than watching the same video encoded at 500 kbps. However, the end-to-end available bandwidth between a service provider and a user constrains the highest encoded bitrate that can be smoothly played. When moving on to an LTE cellular network, due to the unstable nature of the wireless channel, the requesting bitrate may be too high, leading to frequent video freezes and rebuffering when the link capacity undergoes fluctuations. Reducing the requesting bitrate can help to alleviate this problem, but it will result in a bad video quality. Currently, HTTP adaptive streaming (HAS) [4] is the state-of-the-art technology used to relieve the issues caused by the variations in the available bandwidth.

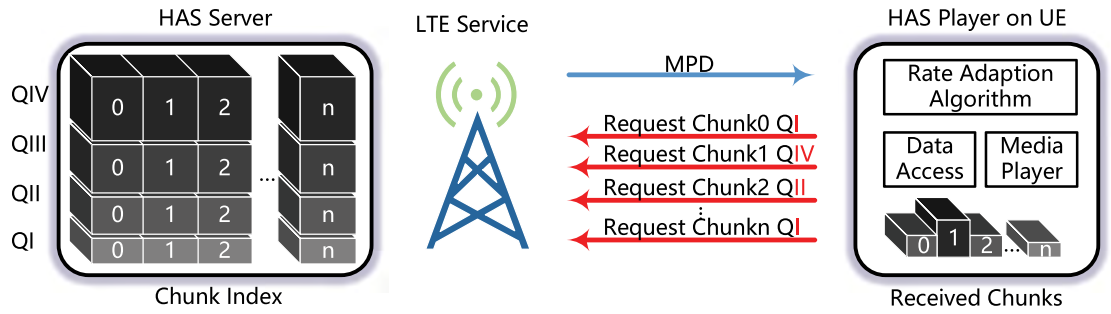

FIGURE 1. General architecture of the HAS system on an LTE cellular network.

Fig. 1 shows the general architecture of the HAS system on an LTE cellular network; it consists of an HAS server, a delivery network, and an HAS player in the user equipment (UE). On the server side, each video is encoded at multiple bitrates to accommodate a range of network capacities. The alternatives are segmented into short chunks, each of which lasts several seconds. For each video, there is a corresponding media presentation description (MPD) file that declares the content profiles, including the number of versions, the available audio and video bitrates, codecs, download URLs and other information for the HAS player. The HAS player operates as a set of sequential HTTP requests to the segmentations of a video. During the viewing session, the HAS player may switch to an alternative with a lower bitrate to avoid rebuffering when the available bandwidth varies or to an alternative with the maximum sustainable bitrate when the available bandwidth is stable to maximize the bandwidth utilization. The client-side rate adaptation algorithm adaptively schedules the target bitrate for each chunk at each switching point.

Regarding an LTE cellular network, rate adaptation is more challenging in the mobile environment than on a wired network. Due to the unstable nature of the radio channel, the disturbances, including multipath effects, interference and noise, will influence the wireless link quality and cause constant bandwidth capacity fluctuations. On the other hand, the limited radio channel resources of a base station are shared among all the attached UE. A base station usually sets a cap on the downlink peak rate for each piece of UE to relieve the cross-traffic competition. When the upper limit is readjusted, there will be instantaneous step-form available bandwidth hopping. Prior works usually struck a balance between the stableness and responsiveness of bitrate switching. They are not applicable to coping with the different bandwidth capacity variation patterns in an LTE cellular network. The rate adaptation should be conservative with respect to constant bandwidth fluctuations to prevent frequent bitrate oscillations; on the other hand, it should be aggressive to quickly respond to bandwidth hopping. How to schedule the bitrate switching in an LTE cellular network is one major problem that needs to be solved.

In this paper, we propose the bandwidth variation pattern-differentiated rate adaptation (BVPDRA) algorithm for HAS service on an LTE cellular network. BVPDRA predicts the near-future bandwidth based on the historical actual measured bandwidth and accordingly outputs the proper video bitrate of the chunk at each switching point. Unlike prior works, BVPDRA does not strike a balance between the stableness and responsiveness of bitrate switching with respect to different types of bandwidth capacity variations. With bandwidth variation pattern differentiation, BVPDRA works with a dual character to simultaneously meet the demand for stability during intervals of constant bandwidth fluctuations and be responsive to instantaneous bandwidth hopping.

Particularly, the contributions of this paper are as follows.

- BVPDRA differentiates the bandwidth variation patterns in an LTE cellular network as constant bandwidth fluctuations and instantaneous bandwidth hopping.
- In the case of constant bandwidth fluctuations, BVPDRA predicts the future bandwidth smoothly to filter out slight bandwidth jitters, and performs conservative rate switching to prevent frequent video quality oscillations.
- In the case of instantaneous bandwidth hopping, BVPDRA predicts the future bandwidth positively to quickly respond to bandwidth variations, and performs aggressive rate switching to prevent stalling and improve bandwidth utilization.

On an LTE network testbed, we extensively evaluate the performance of BVPDRA and make comparisons with existing works. The results show that BVPDRA leads to a higher average bitrate, lower rebuffering ratio and reduced bitrate switching frequency compared to those of other existing works.

The rest of this paper is organized as follows. In Section II, we review the existing works on HAS rate adaptation. We describe the system model of BVPDRA in Section III and the design of BVPDRA in Section IV. In Section V, we show the performance evaluation results of BVPDRA on an LTE cellular network testbed. We summarize the contributions in Section VI.

II. RELATED WORK

As a recent advance in HTTP streaming [5], HAS [6] has generated much research interest from both industry and academia. In the standardization of HAS, the

3rd Generation Partnership Project (3GPP) and the Moving Picture Experts Group (MPEG) have finalized the standard of HAS as MPEG-DASH [7]. There has been broad industry support for HAS, including Microsoft Smooth Streaming [8], Apple Live HTTP Streaming [9] and Adobe HTTP Dynamic Streaming [10]. YouTube has introduced HAS as its default playout method [11]. Netflix [12] is the largest DASH content provider, with up to 14 different quality versions, from 3.6 Mbps to 100 kbps. Akhshabi *et al.* [13] conducted an experimental evaluation of Netflix, Microsoft Smooth Streaming and Adobe HTTP Dynamic Streaming to investigate how they react to available bandwidth variations and concluded that none of them is good enough. The optimization of HAS is still a challenging problem.

Kua *et al.* [14] gave a comprehensive survey of recent research efforts on HAS. In this paper, we focus on client-based and throughput-based rate adaptation approaches. For throughput-based HAS, the available bandwidth measurement and prediction are important in deciding when to perform quality version switching and which quality version to choose. It is very beneficial if HAS can predict the future network bandwidth accurately and can accordingly perform wise quality adaptations [15]. The machine-learning-based bandwidth prediction uses the prior knowledge of the bandwidth model to predict the future bandwidth capacity. Kanhere [16] showed that the Markov decision process is superior to the decision-making challenge when an HAS client needs to decide which quality version to choose. The history-based bandwidth prediction [17]–[21], [22] uses the real-time throughput measurement results from video chunk downloads during a viewing session to predict the future bandwidth capacity and performs TFRC/TCP-like (conservative step-wise up switching and aggressive down switching of representations) rate adaptation.

When moving onto an LTE cellular network, the history-based bandwidth predictions may fail when past bandwidth variations are not a good indicator for predicting the future [23]. Jiang *et al.* [17], Tian and Liu [18], [19], and Liu *et al.* [20] proposed conservative available bandwidth prediction. Thang *et al.* [21] and Juluri *et al.* [22] proposed aggressive available bandwidth prediction. The two different styles of functions have been proven to show inefficiencies in an LTE cellular network. The prediction results are either too aggressive when the available bandwidth suffers from constant fluctuations or too sluggish when the available bandwidth undergoes step-form shifts [24], [25]. Hao *et al.* [26] collected the available network bandwidth for some locations and uploaded them to an assistant server. Mobile devices with a GPS positioning sensor can send queries to the server to predict the near-future bandwidth availability. Essaili *et al.* [27] [27] proposed a cross-layer optimization by jointly optimizing the multiuser network resource allocation and the streaming rate of the DASH clients from the perspective of the mobile network operator. Xiao *et al.*[28], the authors proposed a rate adaptation schema over HTTP/2 instead of HTTP 1.1. Müller *et al.* [29] compared DASH systems using bandwidth traces that were captured under vehicular mobility. In this paper, the validation of the proposed function and comparison with existing works are carried out on an LTE cellular network testbed [30].

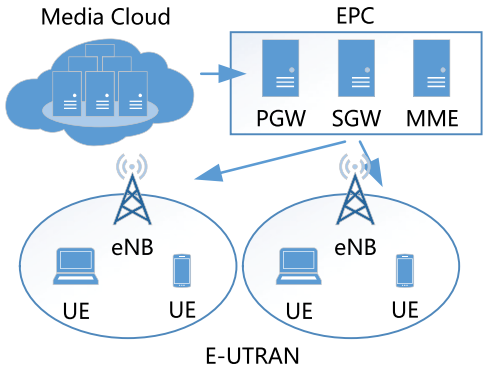

FIGURE 2. Simplified architecture of an LTE cellular network.

III. SYSTEM MODEL

The system model shows how HAS works. In HAS, the server keeps multiple versions of the same video with different levels of quality. Each video file is encoded at different bitrates and is segmented into chunks that are several seconds long. The HAS player sequentially downloads the fragments and plays them back to the user. During a streaming session, the rate adaptation algorithm of HAS allows the player to intelligently decide which bitrate to request for any chunk according to the underlying network conditions.

Network Model: A simplified LTE cellular network architecture, which consists of an evolved packet core (EPC) and an evolved universal terrestrial radio access network (E-UTRAN), is shown in Fig. 2. The EPC consists of a packet data network gateway (PGW), serving gateway (SGW) and mobility management entity (MME). The E-UTRAN includes eNodeBs (eNBs) or base stations and user equipment (UE).

The HAS service is deployed in the media cloud, and the HAS player runs on the UE. The end-to-end connectivity between the UE and HAS server is thus divided into two parts: the wired part and the wireless part. We assume that the wired path that connects the media cloud and the eNBs has adequate network bandwidth and stable propagation and queuing delay. The main concern in regards to the end-to-end bandwidth fluctuations is the wireless last hop between the UE and eNB. In an LTE cellular network, the bandwidth capacity fluctuations differ significantly from the common congestion-caused bandwidth variations in a wired network [31].

Video Model: On the HAS server, there are M different versions of the same video content encoded at different bitrates $R_1 < R_2 < \ldots < R_M$. Each version of the video is segmented into N chunks. Each chunk lasts Ω seconds. Let $Chunk_i^m$ denote the ith chunk of the mth version. Then, the size of

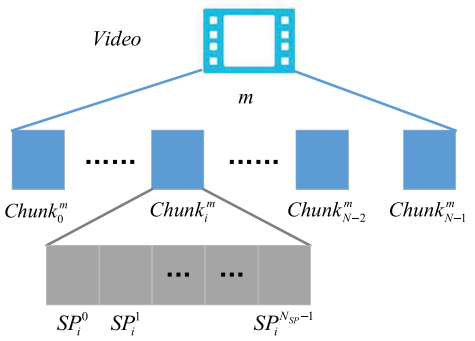

FIGURE 3. Chunk download process of HAS.

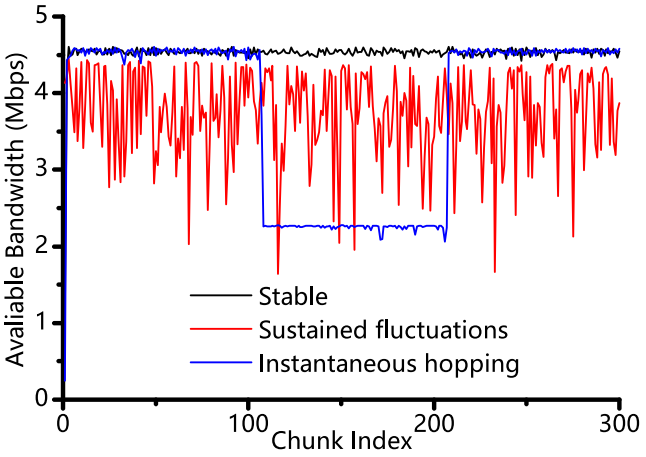

FIGURE 4. Available bandwidth variation in an LTE cellular network.

$Chunk_i^m$ is $L_i = \Omega R_m$. A chunk with a higher bitrate is larger in size and provides a better user viewing experience.

Player Model: The HAS player sequentially downloads chunks of the video content during a viewing session and plays them back to the user. In particular, we further divide the overall download process of $Chunk_i^m$ into N_{SP} sub-downloading processes $SP_i^j (j = 0, 1, ..., N_{SP} - 1)$ with a certain *Interval*, as shown in Fig. 3.

The HAS player maintains a limited player buffer in which to keep the to-be-played chunks. If the requested bitrate is higher than the actual available bandwidth, the player buffer decreases. The user will experience video playback stalling when the player buffer is in a state of underflow. If the requested bitrate is lower than the actual available bandwidth, the player buffer increases. The user will experience poor video quality even though the bandwidth capacity supports a higher video bitrate and quality.

IV. DESIGN OF BVPDRA

In this section, we first give an overview of BVPDRA. Following that, we present the bandwidth variation pattern differentiation function. Then, we introduce the bandwidth prediction function and the quality adaptation function of BVPDRA.

A. OVERVIEW

Commonly, the rate adaptation algorithm of HAS usually consists of two parts: bandwidth prediction and quality adaptation. The bandwidth prediction takes the throughput variation as its input and predicts the near-future bandwidth as its output. The quality adaptation determines the bitrate switch-up and switch-down operations based on the predicted bandwidth capacity and by taking other factors into consideration. When moving onto an LTE cellular network, there are two types of bandwidth variations:

1) SUSTAINED AVAILABLE BANDWIDTH FLUCTUATIONS

In LTE, using the channel quality indicator (CQI) reporting feature, an eNB is aware of the quality of the downlink channel and selects the proper modulation and coding scheme (MCS) to maximize the supported throughput with a given target block error rate (BLER) [32]. When the UE experiences poor channel conditions caused by weak coverage, noise, multipath fading, etc., the eNB that the UE is attached to adjusts the MCS according to the reported CQI. In this situation, the available bandwidth will undergo sustained fluctuations. Examples include when the UE is in an office building or moving along an urban road.

2) INSTANTANEOUS AVAILABLE BANDWIDTH HOPPING

In LTE, data are transmitted via the physical downlink shared channel (PDSCH). The smallest radio resource unit that can be assigned to a UE for data transmission is a resource block (RB). The total number of RBs is fixed according to the cell bandwidth, and RBs are shared among all the UE in a cell. The eNB usually sets an upper limit on the number of RBs that can be assigned to a single piece of UE. The upper bound decides the downlink peak rate of the UE. When the threshold varies due to the redistribution of the radio channel resources, the available bandwidth will undergo several step-form transitions. For example, the UE is handed over or redirected to an idle eNB or cell when the position of the UE is changing.

Fig. 4 shows the typical results of available bandwidth variations in different patterns. The available bandwidth is measured by continually downloading fixed-size files via the HTTP protocol. The existing works usually struck a balance between the stableness and responsiveness of video bitrate switching, which is hard to do for an LTE cellular network. These works may not be applicable. For example, the evaluation of the default rate adaptation algorithm of DASH.js [33] on the LTE cellular network testbed is shown in Fig. 5. The maximum channel bandwidth of the UE jumps from 0.75 Mbps to 2.30 Mbps (t=150 s) and to 1.244 Mbps (t=370 s). The duration is set to 120 s. The details regarding the experimental platform and scenario are discussed in the next section. DASH.js predicts the near-future bandwidth via the moving average method. The method uses the historical bandwidth variation in bandwidth prediction to filter out slight jitters to produce a smoothed prediction result.

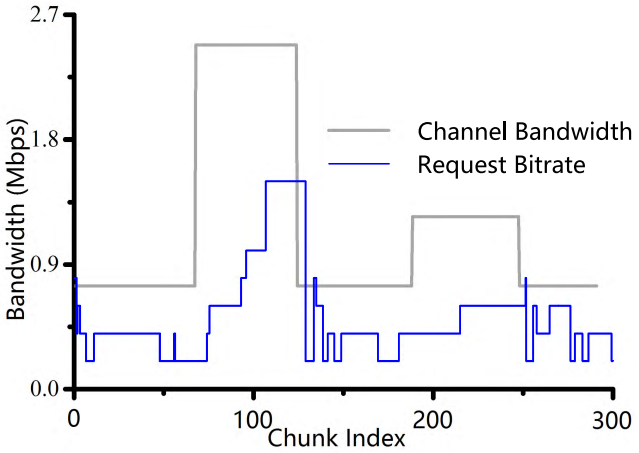

FIGURE 5. Evaluation of DASH.js in the case of bandwidth hopping.

However, DASH.js is unable to cope well with instantaneous step-form bandwidth changes. In this case, the historical bandwidth variation is no longer a good reference for predicting the future bandwidth. In this situation, there is a certain delay in performing necessary bitrate switchings. The defects include underutilized bandwidth capacity and stalling when the bandwidth capacity upshifts and downshifts. For example, when the channel bandwidth jumps from 2.30 Mbps to 0.75 Mbps, the HAS player keeps requesting the video encoded at 1548 kpbs due to the slowly decreasing predicted bandwidth. When the player buffer is nearly in a state of underflow, the requesting video version is forced to select the one with the lowest bitrate.

In light of this, considering the different patterns of bandwidth capacity variations, the goals of BVPDRA include the following:

1) Avoid video playback stalling and maximize the average video bitrate in the case of varying bandwidth capacity.
2) Increase the video bitrate stableness and the switching smoothness in the case of constant bandwidth fluctuations.
3) Increase the video bitrate switching responsiveness in the case of instantaneous bandwidth hopping.

The first one is the most basic goal of designing the HAS rate adaptation algorithm. The motivation of BVPDRA is to differentiate between bandwidth variation patterns and to do so with a dual characteristic to separately achieve the second and third conflicting goals rather than striking a balance between them.

BVPDRA consists of three parts: bandwidth variation pattern (BVP) differentiation, bandwidth prediction and quality adaptation (Fig. 6). The BVP differentiation first differentiates the patterns of bandwidth variations using the historical and most recent actually measured available bandwidths. Bandwidth prediction predicts the future bandwidth capacity. Quality adaptation chooses a version of a video at a "proper" bitrate that is adapted to the predicted bandwidth capacity. BVPDRA differentiates the bandwidth variation patterns of an LTE cellular network as constant bandwidth fluctuations and instantaneous bandwidth hopping. Accordingly, BVPDRA works with a dual character: for constant bandwidth fluctuation, BVPDRA performs smoothed bandwidth prediction and conservative rate switching; for instantaneous bandwidth hopping, BVPDRA performs positive bandwidth prediction and aggressive rate switching.

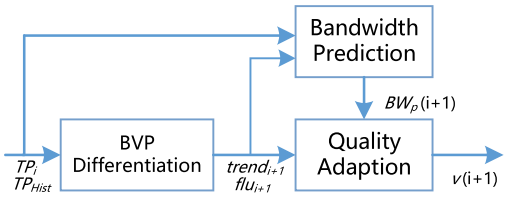

FIGURE 6. Overview of BVPDRA.

B. BANDWIDTH VARIATION PATTERN DIFFERENTIATION

Let $T_i^{(s)}$ be the time instant at which the player starts to download $Chunk_i^m$ and $T_i^{(e)}$ be the time instant at which the downloading of $Chunk_i^m$ ends. Let $D_i = T_i^{(e)} - T_i^{(s)}$ denote the download duration of $Chunk_i^m$.

Let TP_i denote the actual end-to-end available bandwidth measured by downloading $Chunk_i^m$, which is equal to:

$$TP_i = \frac{L_i}{D_i} \tag{1}$$

Let SP_i^j denote the jth sub-downloading process of $Chunk_i^m$. Let $Loaded_i^j$ denote the data transferred during SP_i^j and DSP_i^j denote the duration of SP_i^j. Then, we have:

$$N_{SP} = \lceil \frac{D_i}{Interval} \rceil \tag{2}$$

$$L_i = \sum_{j=0}^{N_{SP}-1} Loaded_i^j \tag{3}$$

$$DSP_i^j = \begin{cases} Interval & j \le N_{SP} - 2 \\ Actual\ duration & j = N_{SP} - 1 \end{cases} \tag{4}$$

Let STP_i^j denote the actual end-to-end available bandwidth measured during the sub-download process SP_i^j of $Chunk_i^m$, which is equal to:

$$STP_i^j = \frac{Loaded_i^j}{DSP_i^j} \tag{5}$$

After the ith chunk has been downloaded, we define $trend_{i+1}$, which denotes the ratio of relative throughput fluctuation, and flu_{i+1}, which denotes the ratio of absolute throughput fluctuation.

$trend_{i+1}$ is defined as:

$$trend_{i+1} = \left| \frac{TP_{i-1} - e^{TP^{jit}(i-1)}}{TP_{i-1} - e^{TP^{jit}(i)}} \right| \tag{6}$$

where $TP^{jit}(i) = \begin{cases} |TP_i - TP_{i-1}| & i \geq 1 \\ \varphi TP_0 & i = 0 \end{cases}$

flu_{i+1} is defined as:

$$flu_{i+1} = TP^{flu}(i) \times STP^{flu}(i) \tag{7}$$

where

$$TP^{flu}(i) = \frac{TP^{jit}(i)}{TP_i} \tag{8}$$

$$STP^{flu}(i) = \frac{STP^{ave}_{jit}(i)}{STP^{ave}(i)} \tag{9}$$

$$STP^{ave}(i) = \frac{\sum_{j=0}^{N_{SP}-1} STP_i^j}{N_{SP}-1} \tag{10}$$

$$STP^{ave}_{jit}(i) = \frac{\sum_{j=0}^{N_{SP}-1} STP_{jit}(i,j)}{N_{SP}-1} \tag{11}$$

$$STP_{jit}(i,j) = \begin{cases} \left| STP_i^j - STP_i^{j-1} \right| & j \geq 1 \\ \varphi TP_i^0 & j = 0 \end{cases} \tag{12}$$

φ is an empirical value and is set to 0.8 in this paper.

The key observation of BVP differentiation is that for the sustained bandwidth fluctuations, due to signal interference on the wireless link, the mean of the historical bandwidth capacity is lower than the maximum downlink bandwidth that the eNB allocated to the UE. The actual bandwidth "shakes" around the mean of the historical bandwidth capacity during chunk downloads. In this situation, the bandwidth fluctuation is in a permanent state. When the bandwidth hops, the maximum downlink channel bandwidth that the eNB allocates to the UE changes. The actual measured bandwidths values before and after the hop remain stable and equal to the channel bandwidth at that time instant. In this case, the bandwidth fluctuation is in an instantaneous state.

We use $trend_{i+1}$ as an indicator of the tendency of bandwidth variation for pattern differentiation. TP_{i-2}, TP_{i-1} and TP_i are used to calculate $trend_{i+1}$, as shown in Eq. (6). Suppose that there is a decrease in TP_{i-1} compared with TP_{i-2}. Upon sustained bandwidth fluctuations, TP_i is expected to increase again to the mean of the historical bandwidth capacity. We have $TP^{jit}(i-1) \approx TP^{jit}(i)$ and $trend_{i+1} \approx 1$. For instantaneous bandwidth hopping, TP_i is expected to stay around TP_{i-1}. We have $TP^{jit}(i-1) > TP^{jit}(i)$ and $trend_{i+1} < 1$.

Let $P(trend_{i+1})$ denote the outcome of BVP differentiation, where:

$$P(trend_{i+1}) = \begin{cases} 0 & trend_{i+1} \leq \tau \\ 1 & trend_{i+1} > \tau \end{cases} \tag{13}$$

$P(trend_{i+1}) = 1$ indicates sustained bandwidth fluctuation, and $P(trend_{i+1}) = 0$ indicates instantaneous bandwidth hopping. The parameter is designed to be sensitive to a larger bandwidth jump with a nearly exponential decrease. τ is an empirical value.

We use flu_{i+1} as an indicator of the severity of bandwidth variations. The parameter takes into consideration two factors: the variety measured based on adjacent chunk downloads and the sub-downloading processes within the latest single-chunk download. It has a positive correlation with the intensity of bandwidth variation.

C. BANDWIDTH PREDICTION

With regard to receiving the ith chunk, we denote the mean of the actual throughput of the most recent m chunks as $TP_{ave}(i, m)$, where:

$$TP_{ave}(i,K) = \begin{cases} \frac{\sum_{j=i-K+1}^{i} TP_j}{K} & i \geq K \\ \frac{\sum_{j=0}^{i} TP_j}{i+1} & i < K \end{cases} \tag{14}$$

Let $BW_p(i+1)$ denote the predicted bandwidth capacity for downloading the $i+1$th chunk, where:

$$BW_p(i+1) = \begin{cases} \lambda TP_{ave}(i-1,K) + (1-\lambda) TP_i & i \geq 1 \\ \varphi TP_0 & i = 0 \end{cases} \tag{15}$$

where λ denotes the weight of the historical and the current bandwidth capacity in predicting the near-future bandwidth and K denotes the length of historical actual measured bandwidth records.

Eq. (15) is a history-based weighted averaging bandwidth prediction method. The average bandwidth for each chunk is measured. To predict the bandwidth for the next downloading chunk, it takes an average of the bandwidths for the K previous chunks and the bandwidth capacity of the current chunk. Each is given a weight λ and $1-\lambda$. K and λ determine the smoothness and responsiveness of a prediction result, respectively. For example, at one extreme, when $\lambda = 0$, $BW_p(i+1) = TP(i)$. The predicted results are related only to the actual throughput of the last chunk and are the most sensitive to fluctuations but less smooth. If the parameters increase, the bandwidth prediction is capable of producing smoother results by taking longer historical records as a reference. However, the prediction reacts to sudden bandwidth hops slowly.

The bandwidth prediction of BVPDRA uses dynamic $\widetilde{k}_{i+1}$ and $\widetilde{\lambda}_{i+1}$ according to the outcomes of BVP differentiation instead of using fixed and carefully selected K and λ following Eq. (15).

$\widetilde{k}_{i+1}$ and $\widetilde{\lambda}_{i+1}$ can be determined as:

$$\widetilde{k}_{i+1} = P(trend_{i+1})\widetilde{k}_i + 1 < K? \quad P(trend_{i+1})\widetilde{k}_i + 1 : K \tag{16}$$

$$\widetilde{\lambda}_{i+1} = \frac{e^{flu_{i+1}}}{1 + e^{flu_{i+1}}} \tag{17}$$

The bandwidth prediction produces different results depending on the bandwidth variation patterns. For instantaneous bandwidth hopping, the historical average bandwidth

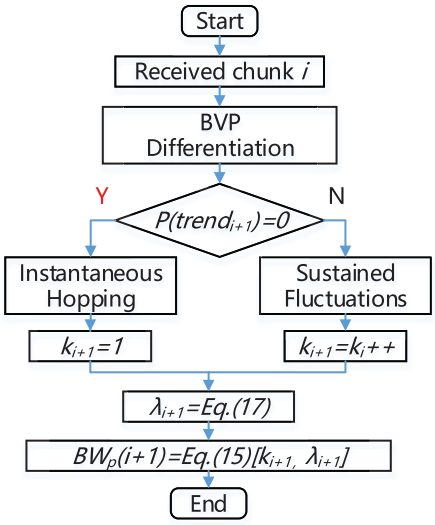

FIGURE 7. Flowchart of the bandwidth prediction function.

capacity is no longer a good indicator for predicting the future bandwidth. In this situation, $\widetilde{k}_{i+1} = 1$. The historical records are made to be forgotten in the bandwidth prediction. Only the most recent available bandwidths TP_i and TP_{i-1} are used as a reference to predict the bandwidth to increase the sensitivity of bandwidth prediction. For constant bandwidth fluctuations, $\widetilde{\lambda}_{i+1}$ increases/decreases exponentially as the bandwidth capacity varies. The lower bound is 0, and the upper bound is 1. As $\widetilde{k}_{i+1}$ increases chunk by chunk from 1 to an upper bound K after a bandwidth hop, use of a fixed λ may not allow achieving the excepted prediction smoothness when $\widetilde{k}_{i+1}$ is relatively small in cases of violent bandwidth variations. Instead, we use a dynamic $\widetilde{\lambda}_{i+1}$. The weight of the historical bandwidth capacity is related to the severity of bandwidth variations to keep the prediction result smooth for smaller $\widetilde{k}_{i+1}$. The flowchart of the bandwidth prediction is shown in Fig. 7.

D. QUALITY ADAPTATION

As the available bandwidth varies during the viewing session, the throughput-based rate adaptation scheme determines the bitrate $R(i+1)$ of the $i+1$th chunk according to the available bandwidth variations. The quality adaptation scheme can be modeled as:

$$R(i+1) = \underset{\{R_m, 1\le m\le M\}}{\arg\min} \left| R_m - BW_p(i+1) \right| \tag{18}$$

where $BW_p(i+1)$ is the predicted available bandwidth for the $i+1$th chunk. In fact, there is only a finite number of bitrate alternatives available for the same video content on the HAS server. The optimal solution can be easily obtained by traversing the M different video versions. If we fit the requested bitrate directly using the predicted bandwidth capacity, the quality adaptation always performs immediate bitrate switching to the video version with the nearest encoded bitrate compared with the predicted bandwidth capacity. The video quality may undergo frequent oscillations in a short time, which has been proven to cause significant QoE degradation [17]. To cope with the conflicting goal of increasing the average bitrate while reducing the bitrate switching frequency as the bandwidth fluctuates, the quality adaptation of BVPDRA is used to control the video bitrate switch, as shown in Algorithm 1.

Algorithm 1 Quality Adaptation Algorithm of BVPDRA

Input: The encoded bitrate of all M different quality versions and the video bitrates; The quality version, V_i, and the bitrate of the current requesting chunk, R_i; The predicted bandwidth, $BW_p(i+1)$; The outcomes of BVP differentiation, $P(trendi+1)$ and λ_{i+1}.

Output: The level index for the next chunk, V_{i+1}, at bitrate R_{i+1}.

$\widetilde{\xi}_{i+1} = 1 - \frac{65+25e^{flu_{i+1}}}{100e^{flu_{i+1}}}$
$\widetilde{BW}_p(i+1) = (1-\widetilde{\xi}_{i+1})BW_p(i+1)$
if $\widetilde{BW}_p(i+1) \ge R_i$ **then**
  if $\widetilde{BW}_p(i+1) < R_i^+$ **then**
    $Counter = 0$;
    return $V_{i+1} = V_i, R_{i+1} = R_i$;
  else
    $Counter++$
    if $Counter \ge \Theta_{i+1}$ **then**
      Switch up to V_i^+
      return $V_{i+1} = V_i^+, R_{i+1} = R_i^+$;
    else
      return $V_{i+1} = V_i, R_{i+1} = R_i$;
    end if
  end if
else
  $Counter++$
  if $Counter \ge \Theta_{i+1}$ **then**
    Switch down to V_i^-
    return $V_{i+1} = V_i^-, R_{i+1} = R_i^-$;
  else
    return $V_{i+1} = V_i, R_{i+1} = R_i$;
  end if
end if

When the ith chunk has been downloaded, the BVP differentiation outputs $P(trend_{i+1})$ and flu_{i+1} following Eq. (13) and Eq. (7). The predicted bandwidth is $BW_p(i+1)$ according to Eq. (15). Instead of directly matching $R(i+1)$ with $BW_p(i+1)$, a common practice is to insert a fixed margin $\xi (0 \le \xi \le 1)$ between the predicted bandwidth and video bitrate. The $BW_p(i+1)$ in Eq. (18) is replaced with $\widetilde{BW}_p(i+1) = (1-\xi)BW_p(i+1)$. When the bandwidth is stable, a fixed margin ξ is sufficient. In an LTE cellular work, due to bandwidth fluctuations, the margin needs to be able to dynamically adapt to the variability of the bandwidth capacity.

Let $\widetilde{\xi}_{i+1}$ denote the dynamic bitrate margin in deciding the bitrate of the $i+1$th chunk, where:

$$\widetilde{\xi}_{i+1} = 1 - \frac{65 + 25e^{flu_{i+1}}}{100e^{flu_{i+1}}} \tag{19}$$

flu_{i+1} is the outcome of BVP differentiation and indicates the severity of bandwidth variations according to Eq. (6). The upper bound of $\widetilde{\xi}_{i+1}$ is 0.25, and the lower bound is 0.1.

Let V_i denote the current quality version and R_i denote the current bitrate. Let V_{i+1} denote the next quality version and R_{i+1} denote its bitrate. If $\widetilde{BW}_p(i+1) \geq R_i$, the predicted bandwidth can keep requesting the current quality version or allow a higher-quality version. Thus, we choose the next higher-quality version index of V_i as a quality switching candidate. It is denoted as V_i^+, and its bitrate is R_i^+. If $\widetilde{BW}_p(i+1) \leq R_i^+$, the quality version remains unchanged, with $V_{i+1} = V_i$. If $\widetilde{BW}_p(i+1) > R_i^+$, it is possible to request a higher-quality version to increase the average bitrate. To avoid frequent bitrate switching, video quality version switch-up is triggered only if the predicted bandwidth $\widetilde{BW}_p(i+1)$ is larger than V_i^+ for Θ successive chunks. When the counter reaches Θ, the quality version switches up, with $V_{i+1} = V_i^+$. If not, the quality version remains unchanged, with $V_{i+1} = V_i$. Before the counter reaches Θ, if the predicted bandwidth is smaller than $\widetilde{BW}_p(i+1) < R_i^+$, the counter is reset to 0. If $\widetilde{BW}_p(i+1) < R_i$, the near-future bandwidth is considered to be insufficient to support the current quality version at index V_i. It is possible to request a lower-quality version to prevent buffer underflow and stalling. Let V_i^- be the first quality version of which $R_i^- \leq \widetilde{BW}_p(i+1)$. Similar to the quality switch-up logic, the switch-down is triggered only if the predicted bandwidth $\widetilde{BW}_p(i+1)$ is larger than R_i^- for Θ successive chunks.

The parameter Θ decides the smoothness of the quality adaptation, in other words, the responsiveness to instant available bandwidth hopping. The dynamic Θ_{i+1} in this paper is a simple two-value function with the output of BVP differentiation: if $P(trend_{i+1}) = 0$, $\Theta_{i+1} = 0$; if $P(trend_{i+1}) = 1$, $\Theta_{i+1} = 5$. The idea is that for constant bandwidth fluctuations, the quality adaptation has to wait for several more chunks before performing quality version switching to improve the smoothness. For instantaneous bandwidth hopping, the switching is performed immediately if necessary.

Note that when upshifting the quality version, BVPDRA chooses the next higher-quality version to avoid sudden quality transitions. When downshifting the quality version, BVPDRA chooses the highest-quality version that matches the predicted bandwidth.

V. PERFORMANCE EVALUATION

In this section, we first describe the LTE cellular network testbed. We implement the prototype of BVPDRA and the existing rate adaptation algorithm. Following that, we describe the experimental setup. Finally, we measure the effectiveness of BVPDRA and make comparisons with existing works.

A. LTE CELLULAR NETWORK TESTBED

In our previous work [30], we built a high-fidelity LTE cellular network testbed called LTE-EMU, as shown in Fig. 8. The testbed has been proven to have significantly higher fidelity than an LTE simulation environment when using the actual end-to-end transmission characteristics collected from a real LTE network (via a field test approach) as the bases of evaluating the fidelity. The topology of LTE-EMU is shown in Fig. 9. The web server and base station emulator (BSE) are connected via the Internet. The user equipment (UE) is a general 150 Mbps Cat 4 LTE-TDD pocket router with radio frequency (RF) interfaces. The UE is connected to the BSE via an RF transmission line. The BSE emulates a real LTE network service in the laboratory. The basic cell parameters of the base station emulator (BSE) and the configurations of the wireless connection between the UE and BSE follow the default values recommended by 3GPP (Table 1). The UE is attached to the BSE and creates an end-to-end IP connectivity with the web server.

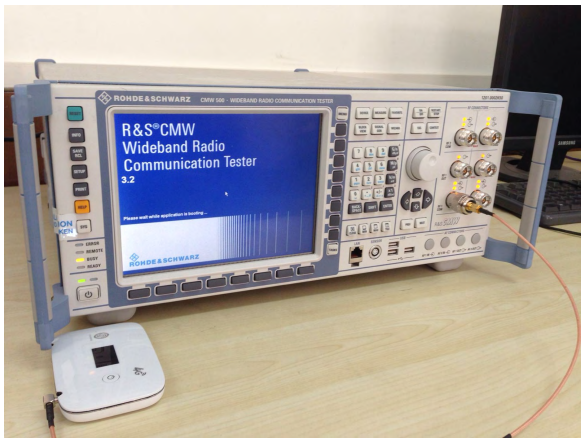

FIGURE 8. LTE cellular network testbed.

TABLE 1. BSE parameters. (a) Cell parameters. (b) Wireless connection parameters.

(a)

Parameter	Value
Duplexing Scheme	TDD
Operating Band	Band 41
Channel	40620 CH
Frequency	2593 MHz
Cell Bandwidth	20 MHz
Antenna Scheme	1x1 (SISO)

(b)

Parameter	Value
RB	50
RLC	UM
Scheduling	Follow wideband CQI
HARQ	Transmission: 2 Sequence: {0,1,2,3}/{0,0,1,2}

B. SYSTEM IMPLEMENTATION

The generic HAS system consists of three parts: server, player and delivery network, with varying available bandwidth.

On the server side, there is usually a web server and a pool of mobile media content. The HAS server in this

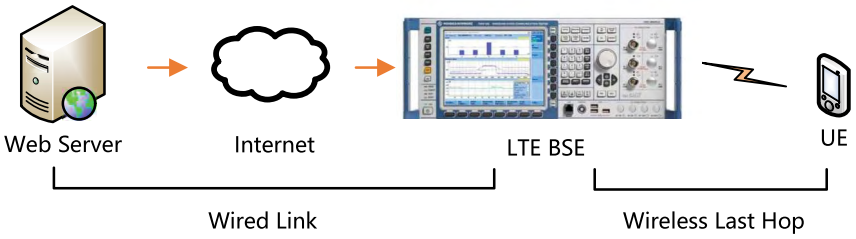

FIGURE 9. Topology of LTE-EMU.

TABLE 2. Multiversion sample video.

Version index	Bitrate (kbps)
1	265
2	462
3	661
4	858
5	1055
6	1548
7	2531
8	4006

paper runs on Windows with the Nginx HTTP Server. For the media content, several representations are created, with each corresponding to a different bitrate. The MPD declares the metadata, including the bitrates, the resolution and other information, for each representation. In this paper, the sample video content is "Big Buck Bunny" [34]. It has been encoded with H.264 video and AAC audio codecs in 8 different versions. The bitrate ranges from 265 kps to 4006 kbps, as listed in Table 2. The dataset has been widely used for the optimization of HAS.

The HAS player with the rate adaptation algorithm continually estimates the available bandwidth and adaptively schedules the target bitrate of each chunk to match the varying network capacity. In this paper, apart from BVPDRA, we also deploy other existing works on HAS, including FESTIVE [17], SARA [22] and the default rate adaptation algorithm of DASH.js [33] (denoted as DASH.js-DEF), to make comparisons with our proposed work.

In practice, the HAS server and the media content are usually deployed on the edge of the mobile cloud using a CDN. We assume that the network bandwidth between the content server and the base station is adequate and that the quality is stable. The bandwidth variations occur during the wireless last hop between the UE and base station due to the nature of the wireless channel. In this paper, the delivery network is an emulated LTE cellular network on LTE-EMU. Using the testbed in the laboratory offers significant advantages compared with the field test approach on a real LTE network or using artificial wireless channels in a simulation environment, such as authenticity and reproducible experimental results. By configuring the BSE, we build the experiment scenarios with different bandwidth variation patterns.

C. EXPERIMENTAL SETUP

To show the effectiveness of BVPDRA and make comparisons with existing works on LTE cellular networks, we define the following experimental scenarios according to the different patterns of the bandwidth variations.

1) STABLE AVAILABLE BANDWIDTH

In this scenario, the available bandwidth remains basically unchanged during the experiment. By default, the UE and BSE are connected via an RF line. The wireless channel is in an ideal state, and the bandwidth is regarded as stable. In real life, a similar scenario would be one in which the UE is in an open environment, for instance, a playground. The maximum channel bandwidth is set to 1.244 Mbps and 4.52 Mbps in this paper.

2) SUSTAINED AVAILABLE BANDWIDTH FLUCTUATIONS

In this scenario, the available bandwidth undergoes sustained fluctuation to test the stableness of BVPDRA. On the testbed, by activating the internal fading simulator (IFS) of the BSE using different multipath fading models, the actual available bandwidth undergoes sustained bandwidth fluctuations. In this paper, the fading models include EPA5Hz, ETU30Hz, EVA70Hz and ETU300Hz, which were defined by the 3GPP [35].

3) AVAILABLE BANDWIDTH VARIATION HOPPING

In this scenario, the available bandwidth undergoes several step-form transitions to test the responsiveness of BVPDRA. In this paper, by adjusting the number of RBs allocated to the UE, the maximum channel bandwidth jumps among 5 different levels. The time interval between adjacent bandwidth hops is 120 s. Before and after the level shifts, the bandwidth is in the stable state. If the IFS is still activated, the scenario is a hybridization of sustained available bandwidth fluctuations and available bandwidth variation hopping.

D. EVALUATION OF BANDWIDTH PREDICTION

In this subsection, we evaluate the bandwidth prediction function of BVPDRA and make comparisons with existing works, including FESTIVE [17], SARA [22] and DASH.js-DEF [33]. SARA uses the single-value method. The latest actual measured bandwidth is set as the predicted future bandwidth, with $BW_p(i+1) = TP_i$. FESTIVE uses the

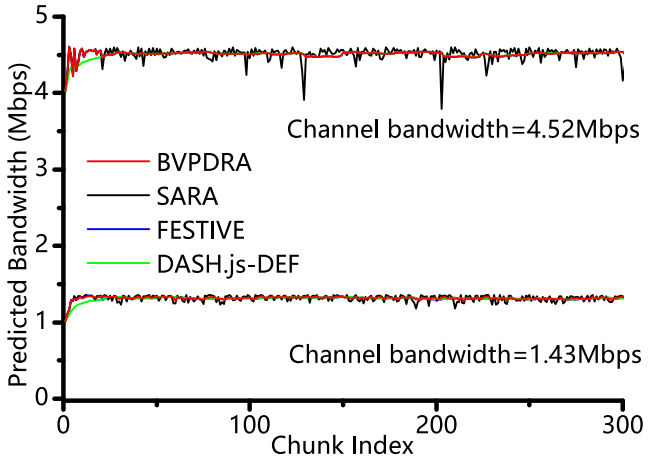

FIGURE 10. Bandwidth prediction in the case of a stable available bandwidth.

harmonic averaging method as follows:

$$BW_p(i+1) = \frac{K}{\sum_{j=i-K+1}^{i} \frac{1}{TP_j}} \tag{20}$$

where $j \geq K$ and $K = 20$.

DASH.js-DEF follows Eq. (15), with fixed $K = 20$ and $\lambda = 0.8$. FESTIVE and DASH.js-DEF tend to perform smoothed bandwidth prediction compared with SARA; on the other hand, both are less sensitive to bandwidth hopping. BVPDRA follows Eq. (15) with dynamic $\widetilde{k}_{i+1}$(Eq. (16)) and $\widetilde{\lambda}_{i+1}$(Eq. (17)).

We use the variance of the predicted bandwidth at each observation point as the quantitative description of the prediction smoothness and the average of the prediction error as the prediction accuracy. A lower value equates to a better prediction smoothness and accuracy.

- Stable Available Bandwidth

The results of bandwidth prediction in the case of a stable available bandwidth are shown in Fig. 10. We carry out two rounds of experiments in this scenario. The maximum channel bandwidth of the BSE allocated to the UE is 1.43 Mbps and 4.52 Mbps. In this scenario, the UE and BSE are connected via an RF line. The wireless connection is expected to be an ideal ratio channel, but the actual situation is that there are still slight bandwidth fluctuations. The prediction smoothness and accuracy are shown in Table 3. As expected, FESTIVE and DASH.js-DEF produce smoothed prediction outcomes. The slight upward and downward bandwidth variations are filtered out. For SARA, the prediction is not related to the history. The bandwidth fluctuations will immediately affect the predicted results. In both situations, the smoothness and accuracy of BVPDRA are similar to those of FESTIVE and DASH.js-DEF and better than those of SARA.

- Sustained Available Bandwidth Fluctuations

The results of bandwidth prediction in the case of sustained available bandwidth fluctuations are shown in Fig. 11. In this scenario, by activating the IFS using different multipath fading models, including EPA5Hz, ETU30Hz, EVA70Hz and ETU300Hz, the actual available bandwidth undergoes sustained bandwidth fluctuations. Similar to the stable available bandwidth scenario, SARA carries out aggressive bandwidth prediction. The predicted results instantly fluctuate according to the latest actual measured bandwidth. Also, FESTIVE and DASH.js-DEF work smoothly under all four fading models. There is a startup delay before the prediction of FESTIVE that flattens off as the function reaches the steady state. When $i \leq K$, FESTIVE uses the single-value method, as does SARA. This is motivated by the player buffer being quickly filled up in the initial stage with the greedy bandwidth prediction. This strategy is less effective when the bandwidth fluctuates intensively. The predicted results of SARA also show that the available bandwidth variations are affected by the different multipath fading models to some extent. For BVPDRA, the predicted results basically match those of DASH.js-DEF. The threshold τ of BVP differentiation is set to 0.61. The bandwidth variations are differentiated as constant bandwidth fluctuations, and the prediction follows a smooth trend.

Table 4 shows the prediction smoothness and accuracy of the four functions in this scenario. In the comparison, we can see that the smoothness of BVPDRA is obviously better than that of SARA and basically consistent with that of FESTIVE and DASH.js-DEF. The average of the actual available bandwidth under the different fading models is approximately 1.4 Mbps. The outcomes of SARA also indicate the severity of the bandwidth variations as a result of using different multipath fading models to some extent. The ETU300Hz fading model in Fig. 11(d) causes the most intense bandwidth variations compared with the other three models. In this situation, the dynamic $\widetilde{\lambda}_{i+1}$ in BVPDRA turns out to be greater than the fixed λ in DASH.js-DEF. This improves the prediction smoothness compared with that of DASH.js-DEF when dealing with more intense bandwidth variations.

TABLE 3. Prediction smoothness and accuracy in the case of a stable available bandwidth.

Channel Bandwidth	Smoothness/Accuracy BVPDRA	SARA	FESTIVE	DASH.js-DEF
1.43 Mbps	***24.7/2.49%***	46.4/2.54%	23.2/2.89%	22.9/2.81%
4.52 Mbps	***41.3/1.30%***	68.9/1.29%	47.1/1.74%	41.8/1.30%

TABLE 4. Prediction smoothness and accuracy in the case of sustained available bandwidth fluctuations.

Fading Model	Smoothness/Accuracy BVPDRA	SARA	FESTIVE	DASH.js-DEF
EPA5Hz	***52.8/3.80%***	79.6/4.50%	58.2/4.00%	53.4/4.00%
ETU30Hz	***56.9/1.44%***	98.2/1.44%	61.9/5.84%	57.2/1.78%
EVA70Hz	***59.2/8.25%***	118.3/9.54%	63.2/7.95%	60.1/8.10%
ETU300Hz	***71.3/13.42%***	251.4/18.32%	80.1/13.81%	73.2/16.10%

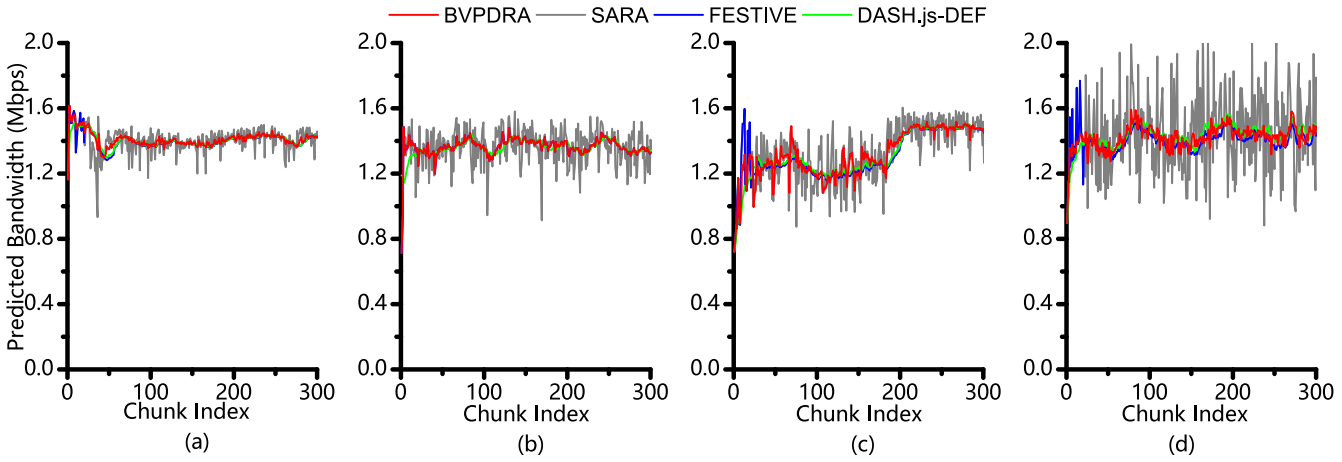

FIGURE 11. Bandwidth prediction in the case of sustained available bandwidth fluctuations. (a) EPA5Hz. (b) ETU30Hz. (c) EVA70Hz. (d) ETU300HZ.

- Instantaneous Available Bandwidth Hopping

In this scenario, by adjusting the maximum channel bandwidth allocated to the UE, the available bandwidth undergoes several upward and downward step-form hops. Two groups of experiments are carried out. In the first group, the IFS is turned off, while in the other group, the IFS is turned on, and the fading model is set as EVA70Hz. The available bandwidth also undergoes sustained fluctuations between adjacent bandwidth hops in the latter case. The bandwidth prediction should be smooth in the case of sustained available bandwidth fluctuations and aggressive to make the predicted bandwidth consistent with the actual bandwidth in the case of available bandwidth hopping.

The results of bandwidth prediction in the case of instantaneous bandwidth hopping are shown in Fig. 12. In Fig. 12, during this set of experiments, the fading model is turned off. The available bandwidth is expected to remain stable before and after the instantaneous bandwidth hopping. The results show that the predicted bandwidths of FESTIVE and DASH.js-DEF lag behind the actual bandwidth changes. Meanwhile, for BVPDRA and SARA, the predicted bandwidth follows the bandwidth hopping in a timely manner. In Fig. 12(b), the fading model is EVA70Hz. The available bandwidth also undergoes sustained fluctuations between adjacent bandwidth hops. In this situation, BVPDRA is more effective. When dealing with sustained bandwidth variations, the prediction smoothness of BVPDRA is similar to that of FESTIVE and SARA. Simultaneously, BVPDRA is also responsive to instantaneous bandwidth hopping, as is SARA. With BVP differentiation, BVPDRA simultaneously accomplishes prediction responsiveness for instantaneous bandwidth hopping and smoothness for sustained available bandwidth fluctuations.

The bandwidth prediction responsiveness is shown in Fig. 13. The point on the boundary line $Y = X$ indicates that the predicted bandwidth is exactly the same as the actual bandwidth. The points that lie on either side of the line indicate that the predicted bandwidth is either larger or smaller than the actual bandwidth. In Fig. 12(a), when the bandwidth shifts from a higher level to a lower level, the predicted results of FESTIVE and DASH.js-DEF are distributed far above the boundary. In contrast, when the bandwidth shifts from a lower level to a higher level, the predicted results of FESTIVE and DASH.js-DEF are distributed far below the line. It takes several iterations (more than 40 s) before the predicted results can reach the actual bandwidth. The processes are represented as a series of points that lie along the Y-axis. The predicted results of BVPDRA and SARA are distributed around the line, ignoring some singularities, and show the advantage of the responsiveness.

TABLE 5. Prediction smoothness and accuracy in the case of available bandwidth hopping. (a) No fading. (b) With fading (Fading Model: ETU30Hz).

(a)

Function	Smoothness	Accuracy
BVPDRA	***32.5***	***5.38%***
SARA	37.9	5.08%
FESTIVE	42.6	14.20%
DASH.js-DEF	43.8	15.61%

(b)

Function	Smoothness	Accuracy
BVPDRA	***87.4***	***9.24%***
SARA	146.3	18.96%
FESTIVE	172.8	18.37%
DASH.js-DEF	157.9	19.95%

Table 5 shows the prediction smoothness and accuracy of the four functions in this scenario. The prediction smoothness is calculated in sections according to the available bandwidth hops. In Table 5, it takes FESTIVE and DASH.js-DEF several iterations before the predicted results can reach the actual bandwidth when the available bandwidth moves up or down. The prediction results are less smooth and accurate. BVPDRA

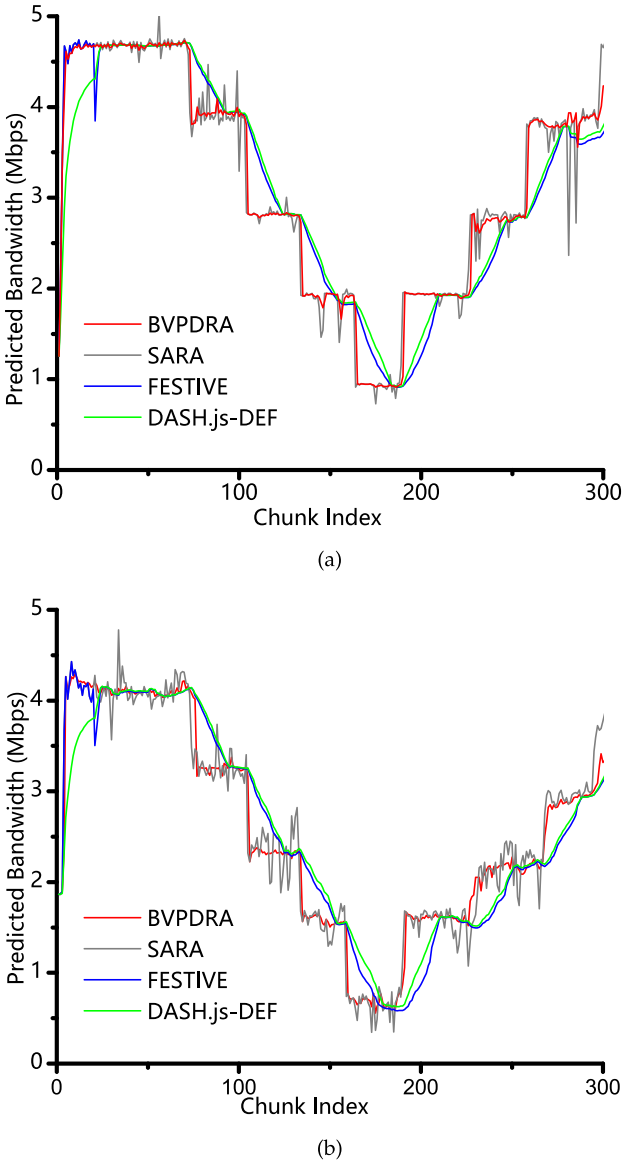

FIGURE 12. Bandwidth prediction in the case of available bandwidth hopping. (a) No fading. (b) With fading (Fading Model: ETU30Hz).

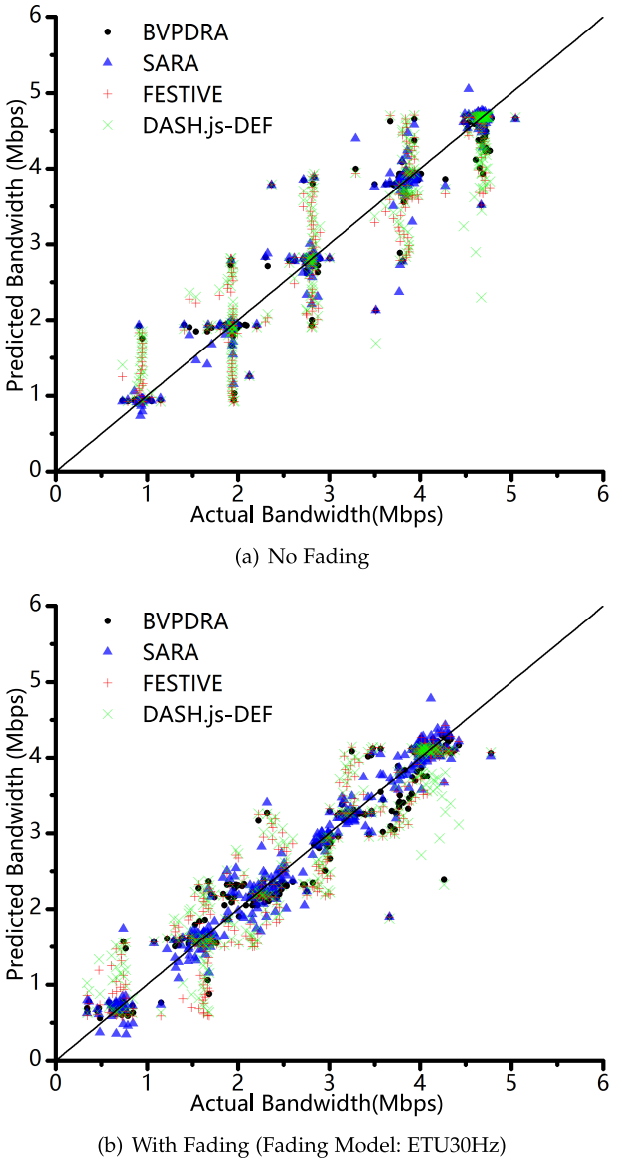

FIGURE 13. Prediction responsiveness. (a) No fading. (b) With fading (Fading Model: ETU30Hz).

achieves an approximately 10% higher prediction accuracy compared with FESTIVE and DASH.js-DEF due to its responsiveness in the case of available bandwidth hopping. In Table 5, SARA exhibits its disadvantage in the case of sustained available bandwidth fluctuations. There are decreases in both prediction smoothness and accuracy. Similar to the former case, the prediction accuracies of FESTIVE and DASH.js-DEF are lower than that of BVPDRA because they are too sluggish in the case of available bandwidth hopping. With BVP differentiation, the performance of BVPDRA is responsive in the case of instantaneous bandwidth hopping and smooth in the case of sustained available bandwidth fluctuations. BVPDRA achieves an approximately 10% higher prediction accuracy compared with that of the other functions.

E. EVALUATION OF QUALITY ADAPTATION

To measure the effectiveness of our proposed rate adaptation algorithm and make comparisons with existing works, we use a simplified QoE model [23]. The following quality metrics are considered: average bitrate, bitrate switching and rebuffering ratio.

The **average bitrate**, measured in kbps, is the average of all chunk bitrates during the entire playback session.

Bitrate switching is the number of times the bitrate changes between adjacent chunks.

The **rebuffering ratio** is the proportion of stalling time with respect to the total playback time.

A better QoE is characterized by a high average bitrate, low bitrate switchings and low rebuffering ratio.

- Stable Available Bandwidth

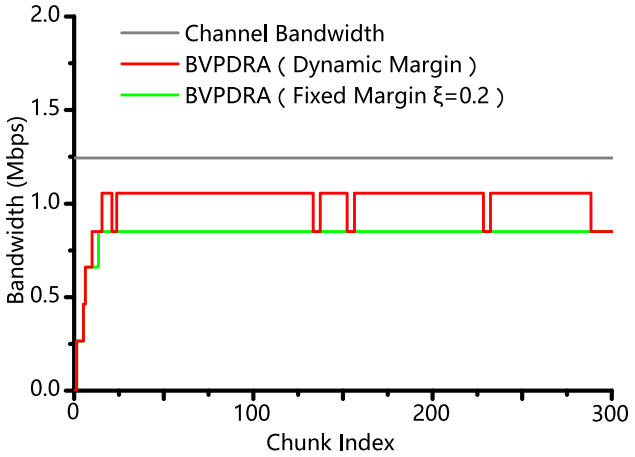

FIGURE 14. Quality adaptation in the case of a stable available bandwidth.

The evaluation of BVPDRA quality adaptation in stable available bandwidth scenarios is shown in Fig. 14. The bandwidth margin is set to a fixed value $\xi = 0.2$, and the dynamic value follows Eq. (19). During the experiments, the channel bandwidth allocated to the UE is set to 1.244 Mbps, and the IFS is turned off. The available bandwidth is expected to be stable. At the beginning, BVPDRA chooses the lowest quality level with the smallest video bitrate. As the predicted bandwidth smoothly increases, BVPDRA continuously switches the quality level to the next higher version step-by-step until reaching the maximum version after several transactions. With a fixed margin $\xi = 0.2$, the request bitrate stays at 858 kbps and remains unchanged until the experiment is finished. The slight bandwidth variations are filtered out and have no effect on the extra bandwidth switching. With a dynamic margin, when the request bandwidth is stable, following Eq. (19), the dynamic value turns out to be larger than 0.2 and approximately 0.11 during the experiments. The request bitrate stays at 1055 kbps. However, the bandwidth variations cause several extra quality version switchings.

In this scenario, the QoE results obtained using BVPDRA, FESTIVE, SARA and DASH.js-DEF are as shown in Table 6. There is no rebuffering for all four functions. With fixed ξ, the results of the four methods are very similar. The quality version switchings are concentrated at the beginning of the viewing session. During the rest of the viewing process, the quality version remains basically unchanged. The average video bitrates are approximately 835 kbps. With a dynamic margin, BVPDRA achieves an approximately 18.8% higher average bitrate with a risk of increased bitrate switching frequency. With different weights on the average bitrate and bitrate switching frequency, the effect of using a dynamic margin can vary. In the rest of the experiments, we use the dynamic margin for BVPDRA.

- Sustained Available Bandwidth Fluctuations

In this scenario, the maximum channel bandwidth of the BSE allocated to the UE is 1.43 Mbps. With the IFS turned on with different fading models, the actual available bandwidth undergoes sustained bandwidth fluctuations. The results of quality adaptation with the ETU30Hz fading model are shown in Fig. 15. With aggressive bandwidth prediction, SARA carries out aggressive quality adaptation. Some quality version switchings seem to be unnecessary because the request for a higher version or a lower version often lasts for several chunks and then switches back. FESTIVE and DASH.js-DEF carry out conservative quality adaptation. Most of the requests are for the quality version indexed with 3 and encoded at 661 kbps with the minimum rate switching frequency. BVPDRA exhibits a higher switching frequency than that of FESTIVE and DASH.js-DEF and is rewarded with a higher average video bitrate. Compared with SARA, the switching frequency is lower, as some useless quality version switchings are avoided. For the other fading models, the four methods yield similar outcomes.

The QoE results obtained using BVPDRA, FESTIVE, SARA and DASH.js-DEF for this scenario are shown in Table 7. During the viewing sessions, there are some rebufferings for all four functions. The competing goals of maximizing the average video bitrate and minimizing stalling are difficult to accomplish perfectly. BVPDRA exhibits a lower bitrate switching frequency compared with SARA but has a similar average bitrate. Some unnecessary bitrate upshifting and downshifting is ignored. BVPDRA achieves

TABLE 6. QoE of quality adaptation in the case of a stable available bandwidth.

Function	Average Bitrate	Bitrate Switching	Rebuffering Ratio
BVPDRA (Dyn-ξ)	***992***	***13***	***0***
BVPDRA (Fixed-ξ)	837	4	0
SARA	835	5	0
FESTIVE	832	4	0
DASH.js-DEF	838	4	0

TABLE 7. QoE of quality adaptation in the case of sustained available bandwidth fluctuations.

Fading Model	Function	Average Biterate	Bitrate Switching	Rebuffering Ratio
EPA5Hz	BVPDRA	***871***	***6***	***0.13***
	SARA	874	9	0.32
	FESTIVE	869	7	0.12
	DASH.js-DEF	862	4	0.11
ETU30Hz	BVPDRA	***690***	***10***	***0.12***
	SARA	705	20	0.33
	FESTIVE	664	10	0.18
	DASH.js-DEF	636	6	0.11
EVA70Hz	BVPDRA	***582***	***12***	***0.67***
	SARA	584	18	0.58
	FESTIVE	574	10	0.67
	DASH.js-DEF	562	8	0.67
ETU300Hz	BVPDRA	***542***	***18***	***0.19***
	SARA	537	26	0.68
	FESTIVE	535	12	0.20
	DASH.js-DEF	531	10	0.17

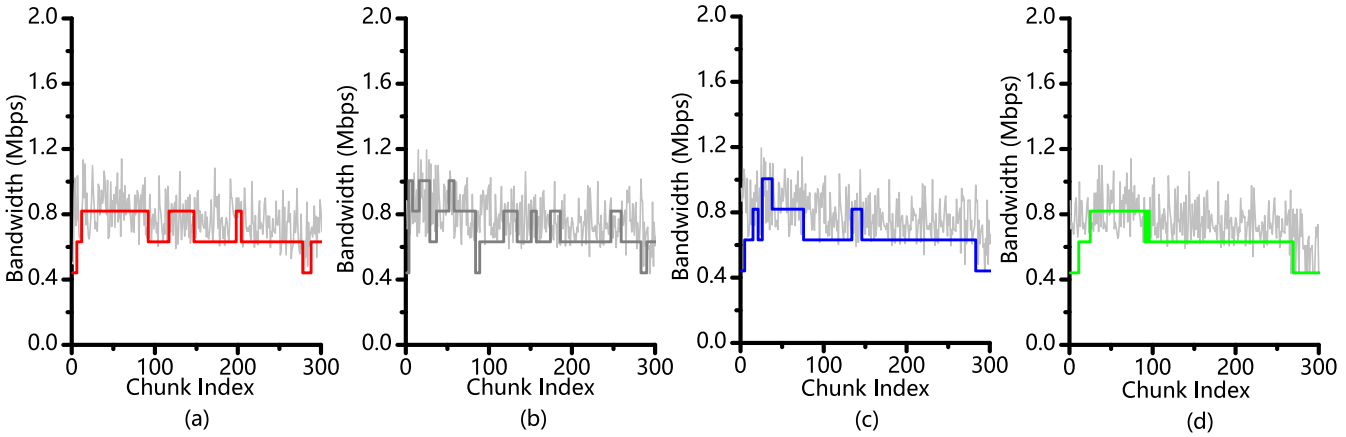

FIGURE 15. Quality adaptation in the case of sustained available bandwidth fluctuations (Fading Model: ETU30Hz). (a) BVPDRA. (b) SARA. (c) FESTIVE. (d) DASH.js-DEF.

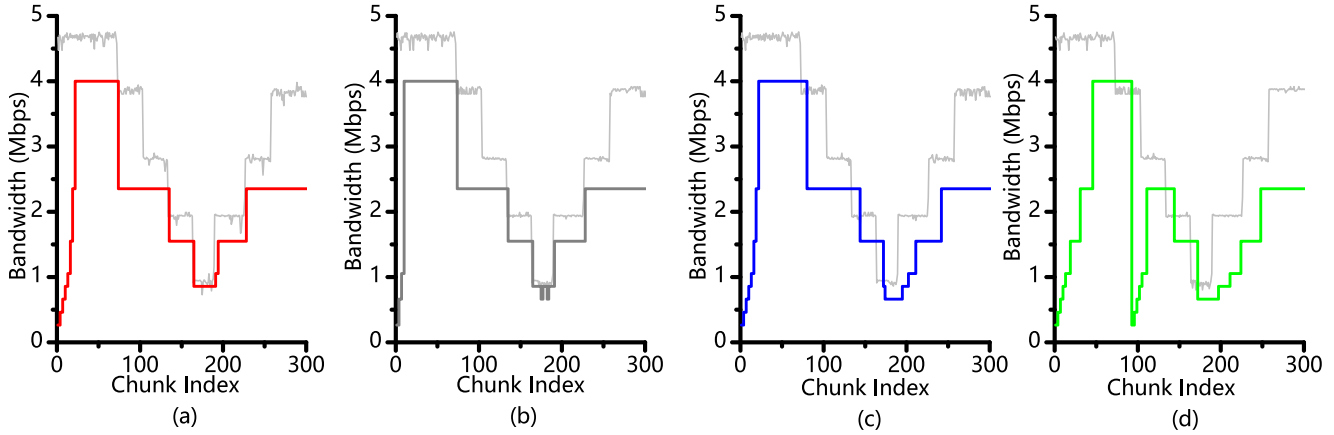

FIGURE 16. Quality adaptation in the case of instantaneous available bandwidth hopping. (a) BVPDRA. (b) SARA. (c) FESTIVE. (d) DASH.js-DEF.

an average bitrate up to 10% higher than that of FESTIVE and DASH.js-DEF. Although the quality version switching frequency is increased, BVPDRA performs the switchings in a smooth way: it prevents consecutive quality version shifts or the switching from crossing multiple quality versions.

- Instantaneous Available Bandwidth Hopping

The quality adaptation results in the case of available bandwidth hopping are shown in Fig. 16. In this scenario, the available bandwidth undergoes several upward and downward step-form hops as the channel bandwidth allocated to the UE is adjusted. As shown in Fig. 16(a), BVPDRA differentiates the bandwidth variation as bandwidth hopping. The prediction results reach the actual bandwidth in a timely manner after the shifts. The quality adaptation immediately performs quality version switch-up or switch-down. The responsiveness of BVPDRA is similar to that of SARA (Fig. 16(c)). FESTIVE (Fig. 16(b)) and DASH.js-DEF (Fig. 16(d)) both perform smoothed bandwidth prediction. It may take some time until the prediction results reach the actual bandwidth or the condition under which the quality version can be switched up or switched down. This causes either wasted bandwidth when the available bandwidth moves up or the risk of rebufferings when the available bandwidth moves down. Both cases lead to poor QoE. In Fig. 16(d), when the available bandwidth moves down from 4.7 Mbps to 3.8 Mbps, DASH.js-DEF continues to request the quality version encoded at a bitrate of 4,006 kbps. Before the prediction results reach the condition under which the quality version is switched down, the player buffer is in a state of underflow, and the playback stops. In this situation, DASH.js-DEF quickly switches to the quality level with the lowest bitrate. It takes several quality version switch-ups before the requested quality level can return to normal.

The QoE of quality adaptation in the case of available bandwidth hopping is shown in Table 8. Compared with SARA, without much degradation of the average bitrate, BVPDRA shows obvious advantages in terms of the reduced bitrate switching frequency and reduced rebuffering ratio. At the same time, compared with FESTIVE and DASH.js-DEF, BVPDRA has a similar rebuffering ratio with a higher average bitrate. In Table 8, when the IFS is enabled and with the ETU30Hz fading model affecting the wireless channel and causing constant bandwidth fluctuations, BVPDRA shows its advantage in stability compared with SARA and

TABLE 8. QoE of quality adaptation in the case of available bandwidth hopping. (a) No fading. (b) With fading (Fading Model: ETU30Hz).

(a)

Function	Average Biterate	Bitrate Switching	Rebuffering Ratio
BVPDRA	***2316***	***14***	***0.13***
SARA	2392	13	0.15
FESTIVE	2275	16	0.11
DASH.js-DEF	2067	20	0.91

(b)

Function	Average Biterate	Bitrate Switching	Rebuffering Ratio
BVPDRA	***2132***	***16***	***0.16***
SARA	2148	26	0.34
FESTIVE	2180	18	0.21
DASH.js-DEF	1926	26	1.32

advantage in responsiveness compared with FESTIVE and DASH.js-DEF. BVPDRA has a similar average bitrate with less bitrate switching frequency and a lower rebuffering ratio.

VI. CONCLUSIONS

In this paper, we propose BVPDRA, a novel rate adaptation algorithm for HAS in an LTE cellular network. BVPDRA has been designed to differentiate between the patterns of bandwidth variation as either constant bandwidth fluctuation or instantaneous bandwidth hopping. In the first case, BVPDRA performs smoothed bandwidth prediction and conservative rate switching to prevent frequent video quality oscillations; in the other case, BVPDRA performs positive bandwidth prediction and aggressive rate switching to prevent stalling. We evaluate the performance of BVPDRA on an LTE network testbed. The results demonstrate that BVPDRA achieves a higher average bitrate and lower rebuffering ratio with a reduced bitrate switching frequency.

REFERENCES

[1] (2017). *LTE Subscriptions Forecast to 2021*. [Online]. Available: https://gsacom.com/paper/lte-subscriptions-forecast-2021/
[2] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1076–1089, May 2017.
[3] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 1st Quart., 2015.
[4] T. Stockhammer, "Dynamic adaptive streaming over HTTP—Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. (MMSys)*, New York, NY, USA, 2011, pp. 133–144. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943572
[5] A. Begen, T. Akgul, and M. Baugher, "Watching video over the Web: Part 1: Streaming protocols," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 54–63, Mar./Apr. 2011.
[6] R. K. P. Mok, W. Li, and R. K. C. Chang, "IRate: Initial video bitrate selection system for HTTP streaming," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 6, pp. 1914–1928, Jun. 2016.
[7] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.
[8] A. Zambelli. *Smooth Streaming Technical Overview*. Accessed: Aug. 17, 2017. [Online]. Available: http://www.iis.net/learn/media/on-demand-smooth-streaming/smooth-streaming-technical-overview
[9] Apple. *Http Live Streaming Overview*. Accessed: Aug. 17, 2017. [Online]. Available: https://www.developer.apple.com/library/content/documentation/networkinginternet/conceptual/streamingmediaguide/introduction/introduction.html
[10] Adobe, *Http Dynamic Streaming*. Accessed: Aug. 17, 2017. [Online]. Available: http://www.adobe.com/cn/products/hds-dynamic-streaming.html
[11] J. Roettgers. *Don't Touch That Dial: How YouTube is Bringing Adaptive Streaming to Mobile, TVs*. accessed: Aug. 12, 2017. [Online]. Available: http://www.gigaom.com/2013/03/13/youtube-adaptive-streaming-mobile-tv
[12] V. K. Adhikari *et al.*, "Unreeling netflix: Understanding and improving multi-CDN movie delivery," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1620–1628.
[13] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. (MMSys)*, New York, NY, USA, 2011, pp. 157–168. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943574
[14] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1842–1866, 3rd Quart., 2017.
[15] V.-H. Vu and T.-Y. Chung, "Bandwidth estimation based on MACD for DASH," in *Proc. 10th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput. (IMIS)*, Jul. 2016, pp. 176–181.
[16] S. Kanhere, "On-move 2016 keynote: Optimizing HTTP-based adaptive streaming in vehicular environment using markov decision process," in *Proc. IEEE 41st Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Dubai, United Arab Emirates, 2016, p. 26, doi: 10.1109/LCN.2016.019.
[17] 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.
[18] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. 8th Int. Conf. Emerg. Netw. Experim. Technol. (CoNEXT)*, New York, NY, USA, Dec. 2012, pp. 109–120. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413190
[19] 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.
[20] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. (MMSys)*, New York, NY, USA, 2011, pp. 169–174. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943575
[21] 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.
[22] P. Juluri, V. Tamarapalli, and D. Medhi, "SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1765–1770.
[23] T. Mangla, N. Theera-Ampornpunt, M. Ammar, E. Zegura, and S. Bagchi, "Video through a crystal ball: Effect of bandwidth prediction quality on adaptive streaming in mobile environments," in *Proc. 8th Int. Workshop Mobile Video (MoVid)*, New York, NY, USA, 2016, pp. 1:1–1:6. [Online]. Available: http://doi.acm.org/10.1145/2910018.2910653
[24] L. R. Romero, "A dynamic adaptive HTTP streaming video service for Google Android," M.S. thesis, KTH, Stockholm, Oct. 2011.
[25] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 20–27, Apr. 2012.
[26] J. Hao, R. Zimmermann, and H. Ma, "GTube: Geo-predictive video streaming over HTTP in mobile environments," in *Proc. 5th ACM Multimedia Syst. Conf. (MMSys)*, New York, NY, USA, 2014, pp. 259–270. [Online]. Available: http://doi.acm.org/10.1145/2557642.2557647
[27] A. El 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.
[28] M. Xiao, V. Swaminathan, S. Wei, and S. Chen, "DASH2M: Exploring HTTP/2 for Internet streaming to mobile devices," in *Proc. ACM Multimedia Conf. MM*, New York, NY, USA, 2016, pp. 22–31. [Online]. Available: http://doi.acm.org/10.1145/2964284.2964313
[29] C. Müller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," in *Proc. 4th Workshop Mobile Video (MoVid)*, New York, NY, USA, 2012, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/2151677.2151686

[30] H. Du, Q. Zheng, W. Zhang, and Y. Huang, "LTE-EMU: A high fidelity LTE cellar network testbed for mobile video streaming," *Mobile Netw. Appl.*, vol. 22, no. 3, pp. 454–463, 2017. [Online]. Available: http://dx.doi.org/10.1007/s11036-017-0868-z
[31] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 703–717, Oct. 2003.
[32] E. Dahlman, Y. Jading, S. Parkvall, and H. Murai, "3G radio access evolution—HSPA and LTE for mobile broadband," *IEICE Trans. Commun.*, vol. E92-B, pp. 1432–1440, May 2009.
[33] dash.js. *Dash.js Home*. Accessed: Aug. 15, 2017. [Online]. Available: https://github.com/Dash-Industry-Forum/dash.js/wiki
[34] Blender. *Big Buck Bunny*. Accessed: Aug. 20, 2017. [Online]. Available: http://www.bigbuckbunny.org/
[35] 3GPP Technical Specification Group. *Base Station (BS) Radio Transmission and Reception*. Accessed: Aug. 16, 2017. [Online].Available: http://www.3gpp.org/ftp//Specs/archive/37_series/37.104/

HAIPENG DU received the B.S. and M.S. degrees in computer science and technology from Xi'an Jiaotong University, China, in 2006 and 2009, respectively, where he is currently pursuing the Ph.D. degree in computer science and technology. His research interests include wireless video streaming and mobile cloud computing.

QINGHUA ZHENG received the B.S. degree in computer software, the M.S. degree in computer organization and architecture, and the Ph.D. degree in system engineering from Xi'an Jiaotong University, China, in 1990, 1993, and 1997, respectively. He was a Post-Doctoral Researcher with Harvard University in 2002, and a Visiting Professor with The Hong Kong University from 2004 to 2005. He is currently a Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University, where he is also the Vice President. His research interests include intelligent e-learning theory and algorithms, computer networks, and trusted software. He received the first place regarding the National Teaching Achievement, State Education Ministry, in 2005, and the first place regarding the Scientific and Technological Development of the Shanghai City and Shaanxi Province, in 2004 and 2003, respectively.

WEIZHAN ZHANG received the B.S. degree in electronics engineering from Zhejiang University, China, in 1999, and the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, China, in 2010. He was a Software Engineer with Datang Telecom Corporation from 1999 to 2002. He is currently an Associate Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. His research interests include multimedia networking, cloud computing, and mobile computing.

XIANG GAO received the B.S. degree in computer science and technology from Lanzhou University, China, in 2016. She is currently pursuing the master's degree with Xian Jiaotong University, China. Her research interest is multimedia systems for e-learning.