Video Streaming Adaptation Strategy for Multiview Navigation Over DASH

Chao Yao, Jimin Xiao, *Member, IEEE*, Yao Zhao, *Senior Member, IEEE*, and Anlong Ming, *Member, IEEE*

***Abstract*—Video content delivery over Internet is receiving increasing attention from both industry and academia, especially for the multiview video contents, as it is the basis to support various applications, such as 3-D video, virtual reality, free view video, and so on. To cope with the dynamic nature of Internet throughput, dynamic adaptive streaming over HTTP (DASH) has been introduced to control the video streaming based on the network conditions. In this paper, we design a streaming framework to improve the user experience of the multiview video streaming over DASH, considering the user behavior of the viewpoint navigation during the streaming process. To eliminate the view switching delay, a multiple view navigation rule is introduced to pre-fetch the possible switching viewpoints. An optimal bitrate allocation scheme is proposed for the introduced rule, allowing the clients to maximize the video quality. Moreover, we found the video quality and the playback starvation probability are conflicting factors, while both are essential for the user's quality of experience (QoE). To tackle this issue, a QoE optimization solution is designed to maximize the overall performance in the proposed framework. Several experiments verify the effectiveness of the proposed framework, and the results demonstrate that the proposed framework outperforms two typical DASH methods.**

***Index Terms*—Multiview video, DASH, QoE, viewpoint navigation, rate adaption.**

I. Introduction

VIDEO streaming is gaining popularity due to the increasing Internet speed and the advancing cloud computing technologies. Users can request and play video content anywhere and at anytime as long as Internet service is available. Netflix, YouTube, Adobe OSFM and other video streaming services have been promoting adaptive streaming, which allows to deliver multiple versions of video contents (i.e., SD, HD and Ultra HD) in heterogeneous networks where network conditions are varying.

Among many adaptive video streaming techniques, the most popular one is the Dynamic Adaptive Streaming over HTTP (DASH). DASH is standardized by the ISO/IEC Moving Picture Experts Group (MPEG) [1], [2]. DASH is a client-driven streaming technology on top of TCP/HTTP without explicitly specifying the rate adaptation mechanism. In a classical DASH system, video contents are pre-encoded at different bitrates, which allows multiple stream representations stored in the server. One representation is divided into segments, each of which is corresponding to several seconds of video clip (such as $2s$). A manifest file called Media Presentation Description (MPD) file is also provided to describe available profiles of each segment. Both video segments and MPD file are stored at the server side. After receiving the MPD file, the DASH client continuously selects suitable representations during a streaming session, depending on the buffer status, available bandwidth and other factors.

In recent years, evolved from single video streaming, multiview video streaming has emerged where the client is able to interactively switch viewpoints among the provided viewpoints from different cameras [3]–[5]. In Fig. 1, a multiview DASH system is depicted where multiple cameras are used to capture different viewpoints, and each viewpoint is encoded into multiple representations with different bitrate levels. For single view DASH system, client is able to switch different representations to meet varying networking conditions, whereas in the multiview DASH system the client is able to switch among viewpoints as well as representations. Several studies have been conducted to support the user's interactive viewpoint navigation for multiview video streaming systems [6]–[11]. However, how to provide users with seamless view switching capability as well as smooth and high playback quality is a challenging task which has not been solved yet.

In DASH systems, in order to have smooth playback, video segments are buffered before being displayed. Generally, the longer the buffer is, the less buffer starvation happens [12]. The video buffer mechanism has a major problem for multiview interactive video streaming. During the streaming process, users may freely switch to other viewpoints, and long view switching delay is generally not acceptable. However, the video segments of newly switched viewpoint may not be available among the buffered videos. To cope with this problem, one possible solution is to download and buffer video segments

Manuscript received April 16, 2018; revised July 19, 2018; accepted September 12, 2018. Date of publication October 3, 2018; date of current version September 4, 2019. This work was supported in part by the National Key Research and Development of China under Grant 2016YFB0800404, in part by the National Natural Science Foundation of China under Grant 61532005 and Grant 61501379, in part by the Jiangsu Science and Technology Programme under Grant BK20150375, and in part by the Specialized Fund for the Basic Research Operating expenses Program of Central College under Grant 2018JBZ001. *(Corresponding author: Anlong Ming.)*

C. Yao and A. Ming are with the Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yaochao1986@gmail.com; mal@bupt.edu.cn).

J. Xiao is with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: jimin.xiao@xjtlu.edu.cn).

Y. Zhao is with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, and also with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing Jiaotong University, Beijing 100044, China (e-mail: yzhao@bjtu.edu.cn).

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

Digital Object Identifier 10.1109/TBC.2018.2871370

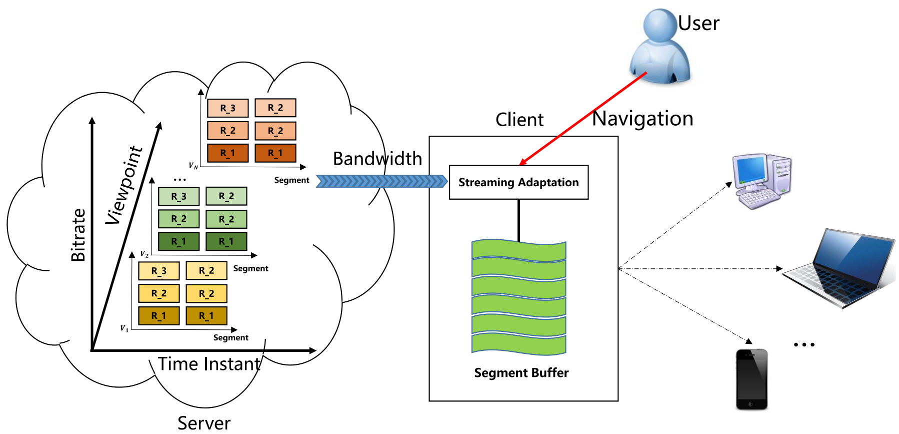


Fig. 1. The diagram of multiview video streaming over DASH. During the streaming process, the client is able to switch among different viewpoints as well as different representations.

of all viewpoints. In such a way, the limited bandwidth is shared among many viewpoints, and the displayed video quality is compromised significantly. Alternatively, another solution is to display the newly switched viewpoint once it becomes available, and in the meantime the buffered segments have been played out. In this case, the long switching delay will sacrifice users' quality of experience.

To tackle this problem, in this paper, we design a multiview video interactive streaming framework with seamless viewpoint switching enabled to improve the user's quality of experience. To eliminate the view switching delay, a multiview navigation rule is introduced to pre-fetch all the alternative viewpoints. A quality level optimization mechanism for the pre-fetch segments is designed to maximize the quality of the displayed segments. Finally, a QoE optimization solution is designed to maximize the overall performance in the playback session.

In summary, our main contributions are as follows:

1) Based on users' multiview switching behavior, we introduce a multiple viewpoint navigation rule for the DASH system. A view switching estimation model is designed to predict the possible switching viewpoints and the corresponding probabilities.
2) We formulate an optimal quality selection strategy for the pre-fetched multiview segments, which allow us to take full advantage of the estimated bandwidth. Moreover, the unequal quality distribution among the possible viewpoints is also discussed in the optimization model, which avoids to waste too much bits on rarely watched viewpoints.
3) We designed a rate adaptation strategy for the multiview interactive streaming framework, where it optimizes the QoE that integrates three factors: *average view quality*, *quality variations* and *playback continuity*. We found that it is of significant importance to maintain a proper buffer length for the multiview streaming framework. To the best of our knowledge, it has never been discussed in literature before.

The remainder of the paper is organized as follows. In Section II, some related works are briefly discussed. In Section III, we introduce the multiple viewpoint navigation model and present the bitrate allocation algorithm among multiple view segments. Section IV introduce the QoE model and the designed multiview rate adaptation algorithm based QoE optimization. Section VI describes our experimental settings and results. Our results are evaluated and compared with state-of-the-arts methods. Finally, We conclude this paper in Section VII.

II. Related Work

In the past decade, many works have been published on interactive multiview systems and adaptive streaming. Interactive multiview video system aims to provide interactions between client and server, in particular for the interactive free view navigation. Liu *et al.* [13] designed a distributed source coding based video coding method to support the interactive multiview video streaming. However, the designed framework focuses on the coding structure to restrain the error propagation caused by view switching within one GOP. However, standard video encoder and decoder are applied in a DASH system as each DASH video segment includes multiple GOPs. Hence, new video streaming strategies are essential to support the interactive multiview video streaming over DASH. For example, Zhao *et al.* [5] designed a cloud-assisted interactive multiview video streaming solution, in which scalable video coding is used to generate the selected viewpoints based on the network condition and the cost of the cloud. Similarly, Su *et al.* [7] proposed a multiview video streaming framework over DASH, the designed framework transmits multiview videos with the assistant of depth map. Fujihashi *et al.* [14] proposed user dependent multi-view video streaming for multi-users (UMSM), but the designed method was mainly for reducing transmission bitrates when multiple users request overlapping frames.

Pre-fetching segments of multiview videos in parallel has been one popular approach for the multiview DASH system. Based on this framework, [15] applied virtual view generation to represent the selected viewpoints, and a rate adaptation logic was designed to maximize the quality of rendered virtual views. Reference [16] also proposed an adaptation strategy to balance the number of views and the quality of views. Unfortunately, none of these works considered viewpoint switching behavior during the streaming process. Recently, [17] proposed a server-based optimal representations for interactive multiview videos, and they introduced the concept of multiview navigation segment to allow users to freely select viewpoints. Several recent works such as [8] and [9] started to optimize the multiview video streaming for navigation. However, it is still a challenging task to design an interactive multiview streaming system with optimizing user experience.

The objective of adaptive streaming is to improve users' quality of experience, which focuses on video playback quality and smoothness [18]–[21]. In [22], several rate adaptation algorithms are evaluated, such as Netflix client, Microsoft Smooth Streaming and Adobe OSMF. The evaluation claims that none of these algorithms can provide smooth quality adaptation with stable playback buffer. The two major factors influencing the users' experience are bandwidth and playback buffer. Thus, rate adaption methods are correspondingly divided into bandwidth-based solutions [23], [24] and buffer-based solutions [25]–[27]. However, solutions solely relying on one single factor fails to bring improvement on the users' quality of experience. For example, the key of the bandwidth-based solutions is to adapt the video bitrate to ensure that the selected video bitrate is lower than the available bandwidth. Thus, it usually leads to a low bandwidth utilization or video quality degradation; buffer-based methods aim to improve the utilization ratio of client buffer, avoiding buffer occupancy or re-buffering. However, frequent bitrate switching largely decreases the visual quality of video streaming. Therefore, multiple factors which could affect users' experience should be jointly considered in DASH system.

QoE model which jointly considers multiple factors for video streaming system is one available approach to evaluate users' experience. Various QoE models have been designed for the streaming system [28]–[30]. Gong *et al.* [31] proposed a five scale model for QoE calculation, which includes integrity, retainability, availability, usability and instantaneousness. Perkis *et al.* [32] describes a QoE model with technology and user related factors. In fact, many factors could be considered in designing QoE models for some specific environments. Based on the pre-designed QoE model, [33] proposed a QoE adaptation scheme for video applications that maximizes content provisioning and network resources according to user's QoE requirement over resource constrained wireless and mobile networks. Refrerence [34] presented a QoE model based on the distortion between the required bitrate and the actual streaming rate. In conclusion, several general QoE-related factors have been considered in the published works, including the start-up delay, starvation probability, average video quality, utilization ratio of bandwidth and receiving buffer, and the frequency of bitrate switching [12]. Nevertheless, to the best of our knowledge, interactive multiview system evaluated by QoE performance is rarely discussed in the published works.

TABLE I
SUMMARY OF NOTATION

Notation	Definition
N	Total number of viewpoints
V_n	Viewpoint n
ψ	Segment index
QP	Quantization Parameters in the video coding
k^{ψ}	Segment with index ψ in the switch unit k
R	Bitrate in video coding
$R_{i,j}^{\psi}$	Bitrate of segment ψ in Viewpoint i, quality level j
$P(k,n)$	Probability of watching Viewpoint n in switch unit k
$P_l(k,n)$	Probability of left switch to Viewpoint n in switch unit k
$P_m(k,n)$	Probability of keep Viewpoint n in switch unit k
$P_r(k,n)$	Probability of right switch to Viewpoint n in switch unit k
Q	Encoding quality of the segment, in term of PSNR
d	Duration of segment
$C(\psi)$	Estimated bandwidth for requesting segment ψ
$B(\psi)$	Buffer length status after finishing downloading segment ψ
R_{total}^{ψ}	Total bitrate for requesting multiview segments with index ψ
$R_{total,min}^{\psi}$	Total bitrate with all requesting multiview segments ψ is the lowest bitrate level
$\bar{Q}^K$	Average video quality for the playing K segments
$\bar{\xi}^K$	Average quality variations for the playing K segments
ϕ^K	Buffer starvation probability for the playing K segments
Ω^*	Calculated optimal average buffer length

III. MULTIPLE VIEWPOINT NAVIGATION MODEL

The base configuration for DASH system with multiview video is shown in Fig. 1. At the server side, let us assume that there are totally N views, each of which is encoded into several different versions with different quality levels and stored in the form of segments. The set of viewpoints is represented as $\{V_1, V_2, \ldots, V_N\}$, and the current segment is represented as ψ, each view is a texture video, and coded with different QP levels. The coding quality level is Q, the corresponding bitrate is R. Thus, for the segment ψ in view i and with quality level j, the rate is denoted as $R_{i,j}^{\psi}$ ($R_{i,j}^{\psi} \in \mathbb{R}$).

At the client side, users are able to smoothly select and navigate among the available viewpoints set. Consequently, the task of the streaming adaptation strategy is to select the segments of different viewpoints and the suitable quality of segments, so as to ensure two requirements for multiview DASH system. One is that users are able to switch to different viewpoints without noticing delay, the other is that good quality of experience is maintained.

The DASH client maintains certain length of buffered video to make the video playback smooth. Generally, long buffer length means small probability of video playback starvation, but it also means that the buffered video data will be stored in the buffer for long period before being displayed to the user, and vice versa. For example, if the current watching view is view $n (n \in N)$, the video buffer contains k *switch units*,[1] the user may switch k view distances by the period that these k different views are played out. Therefore, if the video buffer contains k switch units, the newly request video segments should include video segments of views $[n-k, n+k]$. This

[1]The client can change viewpoint if and only after one whole switch unit is played. One switch unit may include several consecutive segments. For example, if one switch unit has 2 segments, it means that clients can change viewpoints after finishing watching the current 2 segments.

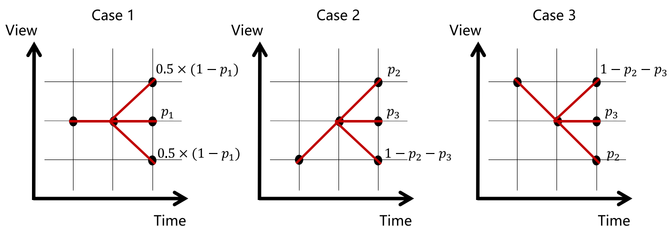


Fig. 2. The switching model. We take 3 cases into account in this model, include keep watching (Case 1), left switch (Case 2) and right switch (Case 3).

means that for long video buffer length, the network throughput will be dedicated for large number of views. Eventually, for a fixed throughput, the video quality for each view degrades with the length of buffered video segments. Therefore, it is important for the DASH client to maintain a proper length of buffer to trade-off the buffer starvation probability and the video quality of each requested view.

A. Viewpoint Switching Model

In this work, we apply the view switch model in [3] to predict the following unit, as shown in Fig. 2. In general, the knowledge of current watching view is not sufficient to predict the switching probabilities of the next unit. However, in the statistical sense, it is largely possible to predict the client behavior when the system knows both the current and previous watching units. For example, if the user is switching from one view to another one, the user will be more likely to continue switching to the same direction, or keep watching the current view, and the probability go back to the previous view is low. On the other hand, if the user has been statically watching a particular view, with high probability he will continue to watch the same view rather than switching to another view. Therefore, the view switch model is given as below:

- The user is watching View n in one switch unit, which is the same as the previous switch unit (View n), the probability to continue watching View n in the following switch unit should be large (denoted as p_1), and the probability of watching View $n-1$ and View $n+1$ should be the same, being $0.5 \times (1-p_1)$.
- If the user is watching View n in the current view switch unit, which is not the same as the previous switch unit, $n-1$ (or $n+1$), the probability to switch to View $n+1$ in the following switch unit is p_2, and the probability of watching View n is p_3, the probability of watching $n-1$ (or $n+1$) is $1-p_2-p_3$.

Based on this view switching model, let us assume the client is watching View n_0 in switch unit k_0. After m view switch units, the user's possible view range is $[n_0-m, n_0+m]$. Let us use $P(k,n)$ $(0 \leq k \leq m)$, $n_0-m \leq n \leq n_0+m$ to denote the probability of watching View n after k view switch units. And that, $P_l(k,n)$ denotes the probability of having the user watching View $n-1$ in switch unit $k-1$, and left switch to View n in the switch unit k. Similarly, $P_r(k,n)$ and $P_m(k,n)$ are used to represent right switching and keeping the same view, respectively. Therefore, we can have

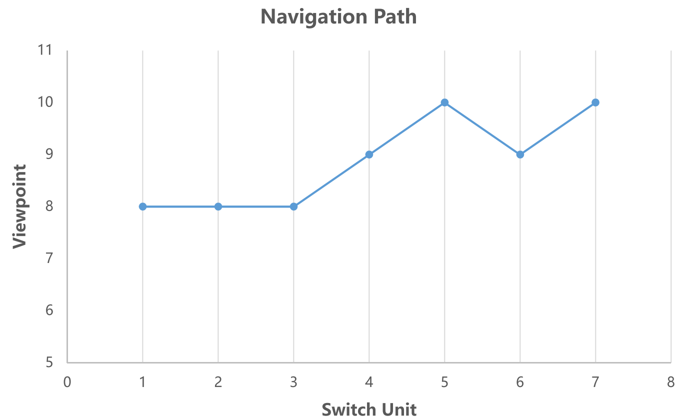


Fig. 3. Navigation Path Example, the initial state is $P_m(k_1, 8) = 1$ and $P(k_1, 8) = 1$.

$$P(k,n) = P_l(k,n) + P_r(k,n) + P_m(k,n) \tag{1}$$

For term $P_l(k,n)$, $P_r(k,n)$ and $P_m(k,n)$, they could be evaluated recursively

$$\begin{cases} P_l(k,n) = p_2 \cdot P_l(k-1,n-1) \\ \qquad\qquad + \frac{1-p_1}{2} \cdot P_m(k-1,n-1) \\ \qquad\qquad + (1-p_2-p_3) \cdot P_r(k-1,n-1) \\ P_r(k,n) = p_2 \cdot P_r(k-1,n+1) \\ \qquad\qquad + \frac{1-p_1}{2} \cdot P_m(k-1,n+1) \\ \qquad\qquad + (1-p_2-p_3) \cdot P_l(k-1,n+1) \\ P_m(k,n) = p_1 \cdot P_m(k-1,n) \\ \qquad\qquad + p_3 \cdot P_l(k-1,n) \\ \qquad\qquad + p_3 \cdot P_r(k-1,n) \end{cases} \tag{2}$$

Fig. 3 shows an example of view switching case for 6 view switch units. In this example, the user is watching viewpoint 8 in switch unit k_1, $P(k_1, 8) = 1$. Meanwhile, based on the known state of switch unit k_0, $P_l(k_1, 8)$, $P_r(k_1, 8)$ and $P_m(k_1, 8)$ can be initialized as $P_m(k_1, 8) = 1$, $P_l(k_1, 8) = P_r(k_1, 8) = 0$. In the following view switch units, the watching probability of each view could be evaluated recursively using (2). It should be noted that the watching probability estimation of each view is based on the same initial state ($P_m(k_1, 8) = 1$). The estimated probabilities are reported in Fig. 4, where the horizontal axis denotes the viewpoint index, and the vertical axis represents the switch probability. We assume that the number of viewpoint is infinite, and we set View 8 as the central viewpoint due to the initial state. For the switch units $k_1 \sim k_6$, the watching probability of each view in each switch unit is shown in the sub-figure of Fig. 4, where $p_1 = 0.7$, $p_2 = 0.6$ and $p_3 = 0.3$.

B. View-Switching Quality Selection

In this section, we will discuss optimal bitrate allocation among multiple viewpoints for interactive streaming. We consider a general case that one single client is requesting multiview video stream over DASH. In the published works [8], [9], [15], [35], parallel viewpoint streaming is a

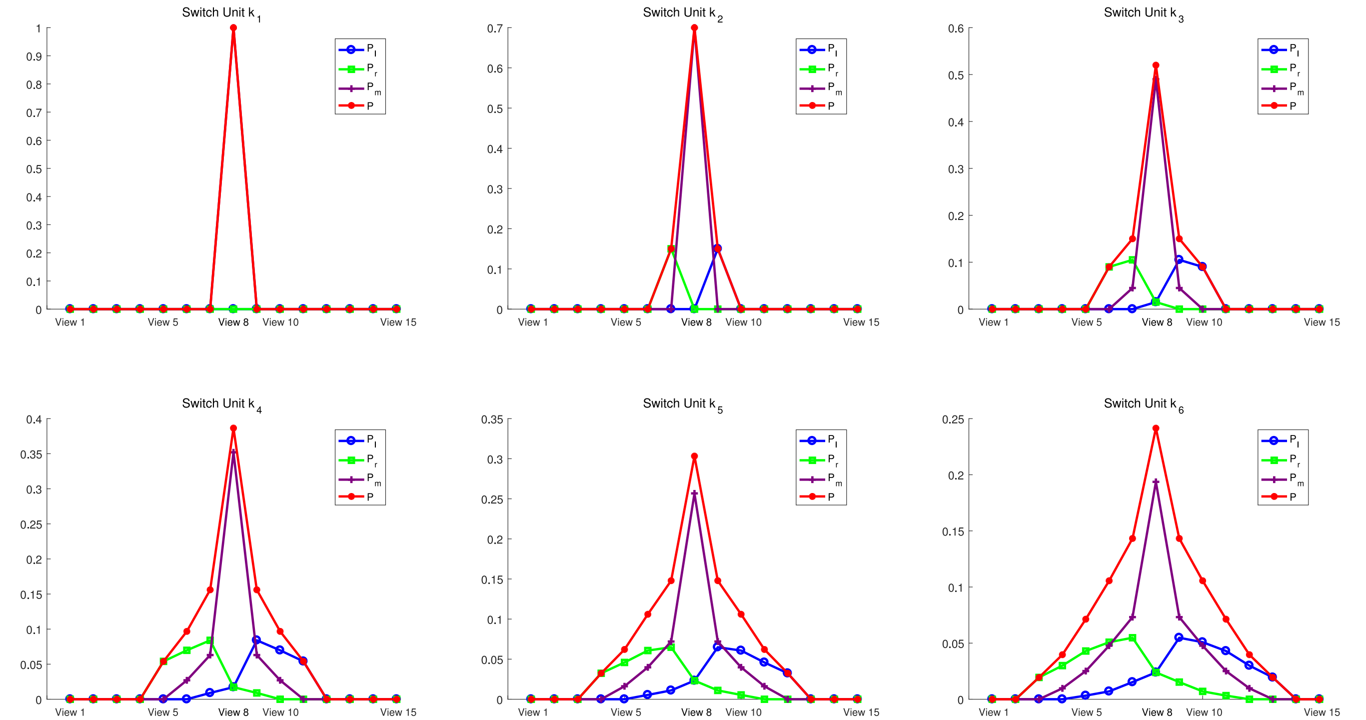


Fig. 4. Estimated probability of each view in 6 switch units, $p_1 = 0.7$, $p_2 = 0.6$ and $p_3 = 0.3$: in the sub-figure, the horizontal axis denotes viewpoints, where the central view is View 8, the vertical axis represents the switch probability.

generic solution which enables the client to request segments of multiple viewpoints in parallel. However, to download the segments of multiple views, round-robin ordering is commonly used, which is non-optimal approach. For example, in most scenarios, only several viewpoints will be watched with a high possibility. It is unnecessary to download the segments of all viewpoints. In this part, the above viewpoint switch model is used to estimate the possible switching views, in addition, an optimal bitrate allocation strategy is proposed.

Based on the viewpoint switching model introduced in Section III-A, a set of segments for several viewpoints have already been loaded in the buffer.[2] Meanwhile, the bandwidth (C) to download the next segment is known using the bandwidth estimation method, the bitrate for each viewpoint should be properly turned. Let us assume there are N views, with different probabilities to be watched, where the probability of segment ψ in View n is denoted as $P^{\psi}(k^{\psi}, n)$. Here, k^{ψ} denotes the switch unit index for segment ψ. $P^{\psi}(k^{\psi}, n)$ could be evaluated using (1), with $\sum_{n=1}^{N} P^{\psi}(k^{\psi}, n) = 1$. It is known that for segment ψ, view n, and quality level j, its rate is denoted as $R_{n,j}^{\psi}$. The video quality in terms of PSNR is $Q^{\psi}(R_{n,j}^{\psi})$. We assume the rate-distortion properties for N views of segment ψ are the same, denoted as $Q^{\psi}(\bullet)$, which is based on the reason that video contents of all views and the corresponding coding properties are similar among the multiple viewpoints. Hence, $Q^{\psi}(R_{n,j}^{\psi}) = 10 \times \log_2 \frac{255^2}{D^{\psi}(R_{n,j}^{\psi})}$. Moreover, for the bandwidth C, the expected video quality for segment ψ is denoted as $Q^{\psi}(C)$, which could be evaluated as:

$$
\begin{aligned}
Q^{\psi}(C) &= \sum_{n=1}^{N} P^{\psi}\left(k^{\psi}, n\right) Q^{\psi}\left(R_{n,j}^{\psi}\right) \\
&= \sum_{n=1}^{N} 10 \cdot \log_2 \frac{255^2}{D^{\psi}\left(R_{n,j}^{\psi}\right)} P^{\psi}\left(k^{\psi}, n\right), \\
& s.t. \sum_{n=1}^{N} R_{n,j}^{\psi} \leq C
\end{aligned}
\tag{3}
$$

where $D^{\psi}(R_{n,j}^{\psi})$ is the Mean Squared Error (MSE) of view n, with its bitrate being $R_{n,j}^{\psi}$.

The decision-making for the segment quality can be formulated to maximize the quality based on the viewpoint switching model, which is also subject to the bandwidth C,

$$
\begin{cases}
\max_{R^{\psi}=\left(R_1^{\psi}, R_2^{\psi}, \ldots, R_N^{\psi}\right)} & Q^{\psi}(C) \\
\text{subject to} & \sum_{n=1}^{N} R_{n,j}^{\psi} \leq C
\end{cases}
\tag{4}
$$

Here, $Q^{\psi}(C)$ is the expected quality of the downloaded segments in all possible views, and segments are encoded with the rate R_n^{ψ}, $n \in 1, 2, \ldots, N$. As the switch probabilities $P^{\psi}(k^{\psi}, n)$ do not depend on the Rate-Distortion properties, this problem can be seen as a classical constrained optimization problem,

$$
J = \arg\max_{R_{n,j}^{\psi}} Q^{\psi}(C) + \ell\left(\sum_{n=1}^{N} R_{n,j}^{\psi} - C\right) \tag{5}
$$

where ℓ is the Lagrangian Multiplier. Moreover, there is no dependencies between the distortions of different viewpoints.

[2]To simplify description, one switch unit has one segment in this part.

This constrained optimization problem can be solved by means of the standard Lagrangian approach, and we get

$$\begin{aligned} \frac{\partial Q^{\psi}\left(R_{1,j}^{\psi}\right)}{\partial R_{1,j}^{\psi}} P^{\psi}\left(k^{\psi}, 1\right) &= \frac{\partial Q^{\psi}\left(R_{2,j}^{\psi}\right)}{\partial R_{2,j}^{\psi}} P^{\psi}\left(k^{\psi}, 2\right) \\ &= \dots \\ &= \frac{\partial Q^{\psi}\left(R_{N,j}^{\psi}\right)}{\partial R_{N,j}^{\psi}} P^{\psi}\left(k^{\psi}, N\right) \end{aligned} \quad (6)$$

It can be also written as

$$\begin{aligned} \frac{P^{\psi}\left(k^{\psi}, 1\right)}{D^{\psi}\left(R_{1,j}^{\psi}\right)} \frac{\partial D^{\psi}\left(R_{1,j}^{\psi}\right)}{\partial R_{1,j}^{\psi}} &= \frac{P^{\psi}\left(k^{\psi}, 2\right)}{D^{\psi}\left(R_{2,j}^{\psi}\right)} \frac{\partial D^{\psi}\left(R_{2,j}^{\psi}\right)}{\partial R_{2,j}^{\psi}} \\ &= \cdots \\ &= \frac{P^{\psi}\left(k^{\psi}, N\right)}{D^{\psi}\left(R_{N,j}^{\psi}\right)} \frac{\partial D^{\psi}\left(R_{N,j}^{\psi}\right)}{\partial R_{N,j}^{\psi}} \end{aligned} \quad (7)$$

Since $\partial D(R)/\partial R$ stands for how MSE changes with bitrate for video compression. In H.264/AVC, $\partial D(R)/\partial R = -0.85 \times 2^{(QP-12)/3.0}$ [36]. In the latest video coding standard HEVC, $\partial D(R)/\partial R = \alpha \times W_k \times 2^{(QP-12)/3.0}$, where α and W_k are affected by the GOP structure [37]. We could assume that they are constant for different views because GOP structure is generally the same among all the views. Depending on that the coding property of each segment does not relate to the other views, Therefore (refer to [38] and [39]), for any two views u and v, their bitrate $R_{u,j}^{\psi}$ (with QP being QP_u) and $R_{v,j}^{\psi}$ (with QP being QP_v) allocation should follow the relation:

$$\frac{P^{\psi}\left(k^{\psi}, u\right)}{P^{\psi}\left(k^{\psi}, v\right)} \frac{D^{\psi}\left(R_{v,j}^{\psi}\right)}{D^{\psi}\left(R_{u,j}^{\psi}\right)} = 2^{(QP_v - QP_u)/3.0} \quad (8)$$

It is equivalent as

$$\frac{P^{\psi}\left(k^{\psi}, u\right)}{P^{\psi}\left(k^{\psi}, v\right)} \frac{D(QP_v)}{D(QP_u)} = 2^{(QP_v - QP_u)/3.0} \quad (9)$$

Let us use $\beta(QP_v, QP_u)$ to replace $\frac{D(QP_v)}{D(QP_u)}$. Thus, we have

$$\frac{P^{\psi}\left(k^{\psi}, u\right)}{P^{\psi}\left(k^{\psi}, v\right)} \beta(QP_v, QP_u) = 2^{(QP_v - QP_u)/3.0} \quad (10)$$

then

$$QP_v - QP_u = 3 * \log_2 \frac{\beta(QP_v, QP_u) P^{\psi}\left(k^{\psi}, u\right)}{P^{\psi}\left(k^{\psi}, v\right)} \quad (11)$$

Here the information of $D(QP_v)$ and $D(QP_u)$ is available during the coding process, and are stored in the DASH sever, so $\beta(QP_v, QP_u)$ could be obtained. The value of QP_v and QP_u can be solved using an iterative approach with a constraint on the total bandwidth $\sum_{n=1}^{N} R_{n,j}^{\psi} \leq C$.

IV. QoE Optimization Model

In this section, we optimize the overall QoE performance of the DASH system which integrates the proposed view switch model. Here, we apply a common definition of QoE model, the key elements are the following:

- *Average View Quality:* The average quality over the entire session is one key factor for QoE. Let us assume the selected watching view for segment ψ is view i^*, and the selected quality level is j^*. The average video quality $\bar{Q}^K$, which is in term of PSNR, for K segments could be evaluated as follows:

$$\bar{Q}^K = \frac{1}{K} \sum_{\psi=1}^{K} Q^{\psi}\left(R_{i^*,j^*}^{\psi}\right) \quad (12)$$

- *Average Quality Variations:* The quality fluctuations between segments will also affect QoE. Let us define that the average quality variations $\bar{\xi}^K$ could be evaluated as follows:

$$\bar{\xi}^K = \frac{1}{K-1} \sum_{\psi=1}^{K-1} |Q^{\psi+1}\left(R^{\psi+1}\right) - Q^{\psi}\left(R^{\psi}\right)| \quad (13)$$

- *Playback Continuity:* The instances that a streaming user sees freezing images should be avoided. This is especially true for the interactive system with free viewpoint switch. We use $\bar{\phi}^K$ to denote buffer starvation probability for K segments.

The QoE of a video shot including segments 1 to K can be denoted by a weighted sum of the aforementioned components:

$$QoE = \bar{Q}^K - \lambda \bar{\xi}^K - \mu \bar{\phi}^K. \quad (14)$$

Here, λ, μ are positive weighting parameters corresponding to *average quality variations* and *playback continuity*, respectively. In common, λ should be set a relatively small value, and μ should be given a large value. This common setting meets the practical requirement because users are deeply annoyed with the display freezing.

To maximize QoE, we characterize (14) as a buffer-related rate adaptation problem. As we know, the longer the DASH buffer length is, the number of buffered possible watching views is larger. Since the total bandwidth C is limited, larger number of possible switching views means that the average bitrate for each view is less. Consequently, the *average view quality* of the whole video session decreases correspondingly. To validate this, we conduct a simple experiment. Pre-encoded multivie video segments are stored in the sever side. To generate different representations for each view, each viewpoint of the test video is encoded with different QPs. To evaluate the relation between the buffer length and the average quality, the network bandwidth is set as a fixed value, including 3000kbps, 5000kbps, 7000kbps and 10000kbps. As (2) and (11), we can obtain the possible switching viewpoints and the corresponding quality levels based on the buffered switch units. For example, when the buffer stores 1 switch unit, the number of the possible switching viewpoints in the next switch unit should be 3. In Fig. 5, how $\bar{Q}^K$ changes with the number of possible switching views is reported. Interestingly, under the a fixed bandwidth, $\bar{Q}^K$ changes linearly with the number of possible switching viewpoints. Therefore, we could have the

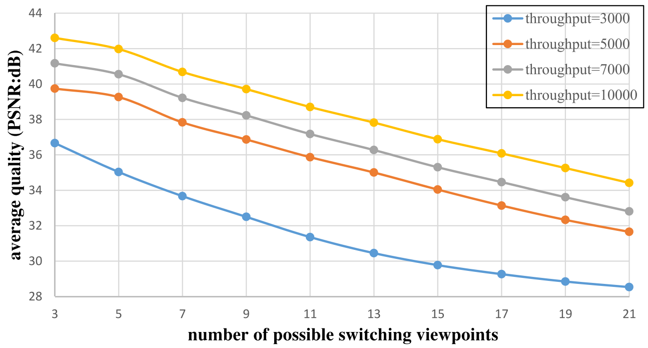


Fig. 5. How $\bar{Q}$ changes with the number of the possible switching viewpoints in different network condition.

following formula:

$$\bar{Q}^K = a_1\Omega + b_1, s.t., \sum R^K \leq C^K \tag{15}$$

where Ω is the average video buffer length, and $a_1 < 0$. The parameters a_1 and b_1 could be obtained by linear curve fitting technique.

Quality variation is usually caused by bandwidth and buffer occupancy change during the streaming process. In practical DASH system, the bandwidth and buffer occupancy randomly vary in the playback session. Moreover, users' behaviors are uncontrollable which add to uncertainty of quality variation in the whole playback session. Therefore, it is very difficult to evaluate quality variation one segment by one segment. Meanwhile, several published buffer management methods [25], [26], [34] have proved that "buffer-based" method can decrease the video quality variation.

Playback continuity largely depends on the DASH client buffer which is used to store unplayed video content. Once there is no video data in the buffer, playback will stop. Certain buffer level must be maintained to avoid this. However, a long buffer length is also undesired as it decreases the average playback quality. As reported in [12], the buffer starvation probability is an exponential function of the average buffer length,

$$\bar{\phi}^K = a_2 e^{-b_2\Omega} + c_2 \tag{16}$$

where parameters a_2, b_2 and c_2 can be obtained using the analytical method in [12] or simple fitting methods.

Finally, the problem becomes to find an optimal average buffer length Ω^* that could maximize the QoE value by integrating (14), (15), (16). The QoE metric can be represented as

$$QoE(\Omega) = \bar{Q}^K(\Omega) - \mu\bar{\phi}^K(\Omega) - \lambda\bar{\xi}^K \tag{17}$$

where μ is predefined weight, which can be assigned according to user preference. Here, $\bar{\xi}^K$ is not a function of Ω because Ω will not affect $\bar{\xi}^K$ too much once it is in a reasonable range. It is a linear relationship between the view quality and the buffer length as proved by (15), we can have

$$\frac{\partial\bar{Q}^K}{\partial\Omega} - \mu\frac{\partial\bar{\phi}^K}{\partial\Omega} = 0 \tag{18}$$

That means the optimal average buffer length should be

$$\Omega^* = -\frac{1}{b_2} ln\frac{-a_1}{\mu a_2 b_2}. \tag{19}$$

V. Streaming Adaptation in DASH System

In this section, a QoE-driven streaming adaptation strategy is integrated into the multiview DASH system. It selects to download segments of certain viewpoints as well as the quality level of these segments.

Let us take segment ψ as an example, the estimated available bandwidth $C(\psi)$ is firstly calculated according to [22] in prior to requesting segment ψ. Via (1), all possible watching viewpoints and the corresponding probabilities can be predicted based on the historical watching record and current buffer length $B(\psi-1)$, the request segment set with index ψ is denoted as $Seg(\psi)$, with the total bitrate being R^{ψ}_{total}. To balance the QoE performance of the multiview DASH system, an ideal buffer length Ω^* is pre-calculated according to (19). As described in Section IV, Ω^* is derived to maximize the QoE performance in the designed framework. Hence, the core of our proposed streaming adaptation is to maintain the buffer length near Ω^*. Ideally, the total bitrate R^{ψ}_{total} should be adapted depending on the current buffer occupancy, which is formulated as

$$R^{\psi *}_{total} = \arg\min_{R^{\psi *}_{total}} |B(\psi-1) + d - \frac{R^{\psi}_{total}}{C(\psi)} d - \Omega^*| \tag{20}$$

Here, $\frac{R^{\psi}_{total}}{C(\psi)} d$ represents the expected consumed time to download $Seg(\psi)$, and d is the duration of one segment.

In practical implementation, we set a constraint condition $R^{\psi}_{total} = \omega \cdot C(\psi)$, where ω is a regulation factor used to adapt the level of R^{ψ}_{total}. When the buffer length is shorter than Ω^*, it means that the probability of buffer starvation is high. ω should be set to a small value to download segments with a fast speed, otherwise ω should be set a large value. Finally, by solving (11), we can select the best bitrate levels of all requested segments.

In our proposed strategy, a buffer-based approach is applied to set the value of ω. Three conditions are considered as follows:

1) The current buffer length $B(\psi-1)$ is larger than Ω^*, denoted as $B(\psi-1) > \Omega^*$. Hence, to ensure that the buffer length $B(\psi)$ after downloading the requested segments is near to Ω^*, ω can be set as

$$\omega = \frac{B(\psi-1) + d - \Omega^*}{d} \tag{21}$$

2) The current buffer length $B(\psi-1)$ is equal to Ω^*, denoted as $B(\psi-1) = \Omega^*$. In this case, the buffer length $B(\psi)$ can be kept as Ω^* after downloading the requested segments, so

$$\omega = 1 \tag{22}$$

3) The current buffer length $B(\psi-1)$ is less than Ω^*, denoted as $B(\psi-1) < \Omega^*$. To adapt the bitrate levels of the requested segments, two conditions should be followed: i) the

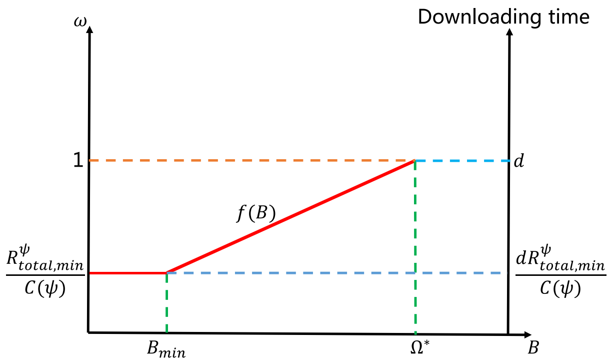


Fig. 6. ω selection as a function of buffer occupancy.

buffer length should increase, $B(\psi) > B(\psi - 1)$; ii) the speed to fill the buffer should be as fast as possible. In this case, the extreme value is that all selected quality level is the lowest bitrate level of the requested segments, the total lowest birate is denoted as $R^{\psi}_{total,min}$. The conditions can be formulated as

$$\begin{cases} d > \frac{R^{\psi}_{total}}{C(\psi)} d \\ R_{total} \geq R_{total,min} \end{cases} \tag{23}$$

Thereby, we design a piecewise linear function $f(B)$ to pick ω, which is based on the current buffer occupancy, as shown in Fig. 6.

Note that we have two key nodes in the designed piecewise linear function. Firstly, we set a minimal value for the buffer length, which is used to avoid the starvation event. The minimum buffer length can be represented as

$$B_{min} = \frac{R^{\psi}_{total,min}}{C(\psi)} d \tag{24}$$

Then, another key node can be identified by Ω^*. The piecewise linear function is represented as

$$\omega = \begin{cases} \frac{R^{\psi}_{total,min}}{C(\psi)}, & B(\psi - 1) \leq B_{min} \\ f(B), & B_{min} < B(\psi - 1) < \Omega^* \end{cases} \tag{25}$$

According to the above guidelines, the complete policy of the proposed rate adaptation is described as the pseudo-code in Algorithm 1.

VI. Experimental Results

A. Experiment Setups

To verify the performance of the proposed framework, we implement all the components of a DASH system based on the *DASH Industry Forum* [40]. The sever is installed with the *Apache* HTTP sever of version 2.4.1, which is used for media delivery. *DummyNet* [41] is used to control the bandwidth capacity. In the experiments, "*Newspaper*" sequence with 9 viewpoints is used, and each view lasts 10 seconds. Specifically, each view of the multiview video is independently encoded by JM 19.0 [42], with different quality levels by setting different QPs. The properties of the test sequence is listed in Table II. Moreover, to obtain the segments compatible with MPEG-DASH standard [43], MP4Box is used to divide the encoded stream into several segments, where the duration of each segment equals to 2 second.[3]

Algorithm 1 QoE-Optimal Rate Adaptation Algorithm

Input: $C(\psi)$: the estimated available bandwidth for segment ψ
$B(\psi - 1)$: the current buffer length before downloading segment ψ
Ω^*: the ideal buffer length
Output: $Seg(\psi)$: all possible segments with index ψ
$R_i^{\psi^*}$ $(i \in [1, \ldots, m])$: rate levels of all segments in $Seg(\psi)$

Identify $Seg(\psi)$ based on the buffered segments in $B(\psi - 1)$, using (1)
Calculate the total bitrate of $Seg(\psi)$, R^{ψ}_{total}:
if $B(\psi - 1) > \Omega^*$ **then**
  $\omega = (B(\psi - 1) + d - \Omega^*)/d$
else
  if $B(\psi - 1) < B_{min}$ **then**
    $\omega = \frac{R^{\psi}_{total,min}}{C(\psi)}$
  else
    if $B(\psi - 1) == \Omega^*$ **then**
      $\omega = 1$
    else
      $\omega = f(B)$
    end if
  end if
end if
Set the constrain bandwidth as $\omega \cdot C(\psi)$ in (5)
Solve (5) and (11), get $R_i^{\psi^*}$ of $Seg(\psi)$
return $R_i^{\psi^*}$, $Seg(\psi)$

TABLE II
Properties of the Test Sequence

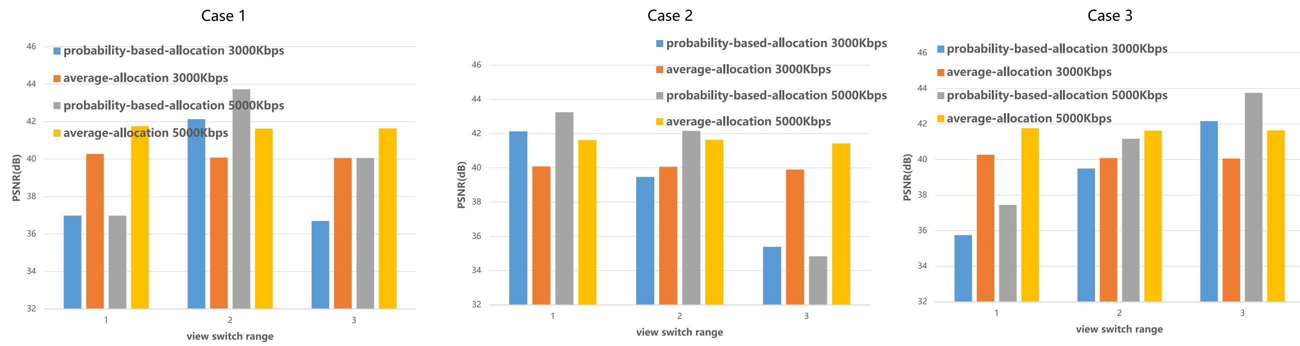
Sequence Name	Frame Rate(fps)	Resolution	Views	Length(s)
Newspaper	30	1024×768	$0 \sim 8$	300

In our implemented platform, the parameters λ and μ in the designed QoE model (17) are pre-defined. Especially for μ, it will be discussed in the following experiments. Note that both of these parameters are different from that in [44], because the first term in (14) represents the PSNR value in our QoE model. In addition, the deployed bandwidth estimation algorithm is the same as that in [22] for fairness. To improve the accuracy of bandwidth estimation, other bandwidth estimation algorithms could be used in our system.

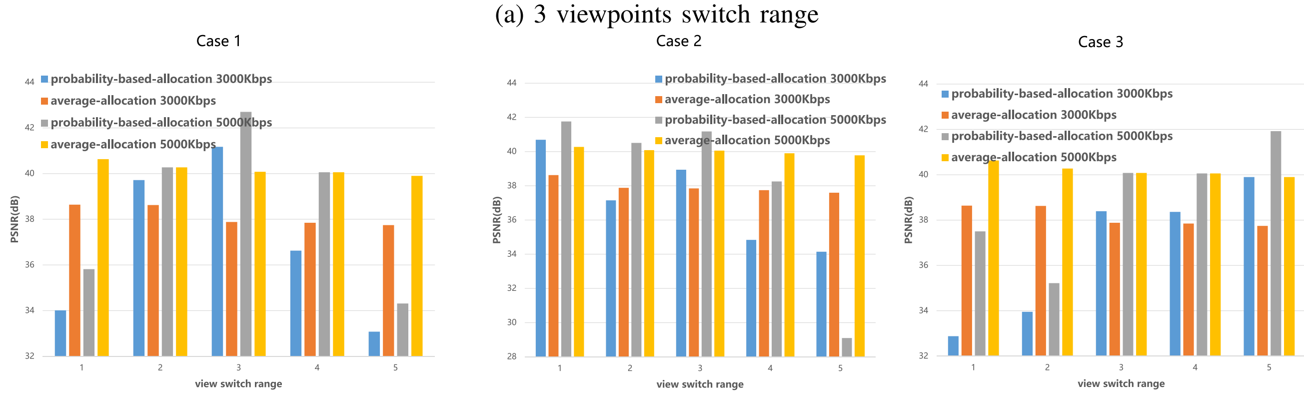
To demonstrate the effectiveness of the proposed method, we have two categories of baseline methods. On one hand, we conduct experiments in which all possible switching viewpoints are equally selected, so that in this scenario all possible segments equally share the estimated bandwidth (called "average-allocation"). It is similar as the pre-fetch approach in [8]. Correspondingly, we called the proposed approach as "probability-based-allocation".[4] On the other hand, we compare the proposed approach with the classical buffer-based

[3]Here, the video content is played in cycle, so that enough segments can be obtained.

[4]All quality levels of possible viewpoints are selected as in Section III-B.



(a) 3 viewpoints switch range



(b) 5 viewpoints switch range

Fig. 7. Average playback quality for several settings, including the number of possible switching viewpoints being 3 and 5, and the total bandwidth being 3000 kbps and 5000 kbps.

TABLE III
IMPLEMENTED ALGORITHMS FOR COMPARISON

Approach	Bandwidth Sharing	Buffer Management
S1	average-allocation	Buffer-Based
S2	probability-based-allocation	Buffer-Based
S3	average-allocation	QoE-Based
proposed	probability-based-allocation	QoE-Based

management scheme. For the classical buffer-based management scheme, two important parameters are defined, including T_{min} and T_{max}. It works as follows: 1) when buffer occupancy is lower than T_{min}, the selected quality levels of all possible switching viewpoints are one level lower than that of the previous downloaded segments; 2) when buffer occupancy is larger than T_{max}, the selected quality levels of all possible switching viewpoints are one level higher than that of the previous downloaded segments; 3) in other cases, the quality levels of the downloading segments should be the same as the previous ones. In our experiment, we set “Buffer-Based” method as the baseline of buffer management, and the proposed buffer management approach in Section V is called “QoE-Based”. All implemented algorithms are listed in Table III.

B. Playback Quality Comparison

In this section, we prove the effectiveness of the quality selection algorithm introduced in Section III-B by comparing the obtained video quality of the “average-allocation” method and the “probability-based-allocation” method. The benchmark method “average-allocation” equally shares the estimated bandwidth among all the possible watched viewpoints, while the proposed “probability-based-allocation” method unequally shares the bandwidth with high-probability-viewed viewpoints allocated with more bits, and vice versa. For the proposed “probability-based-allocation” method, the bitrates of the requested segments for different viewpoints are evaluated using the algorithm described in Section III-B.

In this experiment, we consider the 3 introduced switch cases shown in Fig. 2, with $p_1 = 0.7$, $p_2 = 0.6$ and $p_3 = 0.3$. We consider several cases, where the number of possible switching viewpoints are 3 and 5, and the total bandwidth are 3000 kbps and 5000 kbps. Fig. 7 shows the statistical playback quality of the possible switching viewpoints in the experiment. Note that any switch probabilities can be chosen depending on the user’s switching behavior. As reported in Fig. 7, the “probability-based-allocation” method can always provide the highest playback quality in the case that the viewpoint with the highest probability is selected as the watching viewpoint; the quality difference among the different viewpoints is the lowest by using “average-allocation” method. However, in terms of the average quality for all possible switching viewpoints, the “probability-based-allocation” method outperforms the “average-allocation” method, as shown in Table IV, which are evaluated in different bandwidth conditions and switching cases.

C. Buffer Occupancy Evaluation

In this part, some experimental results are presented to illustrate the buffer occupancy during streaming process. The

TABLE IV
AVERAGE QUALITY FOR ALL POSSIBLE WATCHING VIEWPOINTS

View Range	Case	average-allocation		probability-based-allocation	
		$3000Kbps$	$5000Kbps$	$3000Kbps$	$5000Kbps$
3	1	40.11	41.65	**40.54**	**42.17**
	2	40.06	41.61	**40.66**	**42.08**
	3	40.09	41.65	**40.72**	**42.35**
5	1	38.04	40.14	**38.90**	**40.57**
	2	38.11	40.11	**38.57**	**40.29**
	3	37.93	40.06	**38.26**	**40.22**

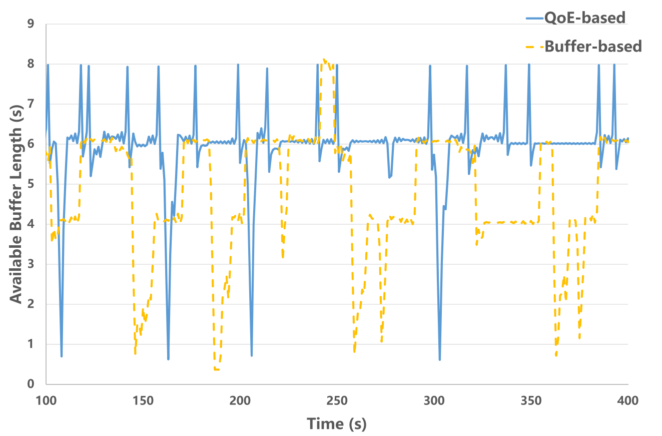


Fig. 8. Comparison of the available buffer length between “QoE-based” and “Buffer-based”.

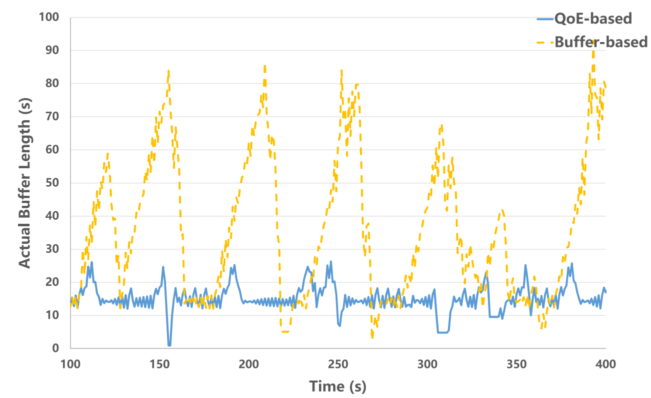


Fig. 9. Comparison of the actual buffer length between “QoE-based” and “Buffer-based”.

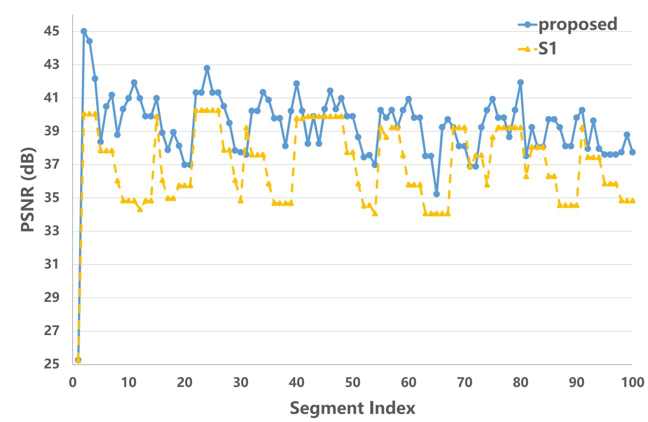


Fig. 10. Comparison of the playback quality between “proposed” and “S1”.

buffer is controlled by the proposed “QoE-based” method and the classical “Buffer-based” method, respectively. It is noted that there are two kinds of buffer occupancy in our designed system: one is total segment duration of watched viewpoints (“available buffer length”); another is total segment duration of all downloaded viewpoints (“actual buffer length”).[5]

Fig. 8 and Fig. 9 show the comparison of the buffer occupancy change under bandwidth varying network, where the ideal buffer length Ω^* in “QoE-based” method is set as $6s$ and $T_{min} = 4s$, $T_{max} = 8s$ in the classical “Buffer-based” method. In the whole play session, the view switch event occur after finishing playing the current switch unit,[6] which follows the defined switch model. As described in Section IV, the quality of the pre-fetched segment largely depends on the available buffer length. The longer the buffered switch units, the larger the number of possible switching viewpoints in the next switch unit. Correspondingly, the quality of fetched segment decreases. Therefore, stable buffer occupancy leads to less quality variation. As shown in Fig. 8, our proposed “QoE-based” method can well control the buffer, so the buffer occupancy stays near to Ω^* for more time. By contrast, “Buffer-based” method tries to have buffer occupancy between T_{min} and T_{max}, which means that the number of pre-fetched segments could always change. Fig. 9 proves that the number of pre-fetched segments by “Buffer-based” method varies more frequently than that by our proposed “QoE-based” method.

[5]Because one segment index could be corresponding to several segments from different viewpoints.

[6]the switch unit has 3 segments in this experiments

D. QoE Performance Evaluation

In this part, the QoE performance of the designed system is evaluated and reported in Table V. In the framework settings, the parameter λ is 0.3, μ is 200 in (17). For the “QoE-based” method, the ideal buffer length Ω^* is $6s$. The classical “Buffer-based” method sets the available buffer range is $T_{min} = 4s$, $T_{max} = 8s$.

Fig. 10, Fig. 11 and Fig. 12 compare the playback quality performance of the first 100 segments of the “proposed” with that of “S1”, “S2” and “S3”, respectively. On one hand, Fig. 10 and Fig. 12 prove that the proposed “probability-based-allocation” method can improve the playback quality of the viewpoint based on the estimated possibilities. This is demonstrated by the fact that the playback quality of the selected viewpoints using “probability-based-allocation” is higher than that of “average-allocation” method. Note that,

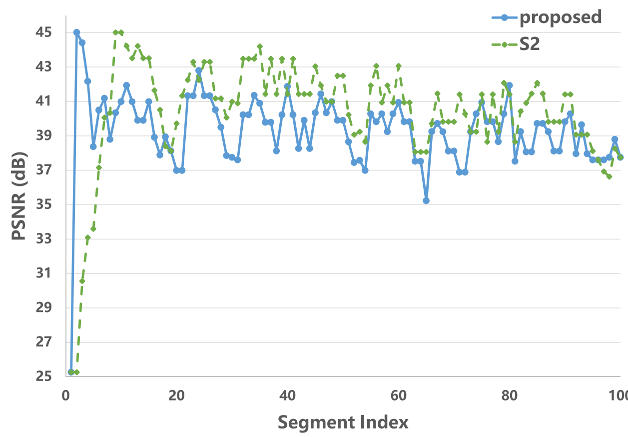


Fig. 11. Comparison of the playback quality between "proposed" and "S2".

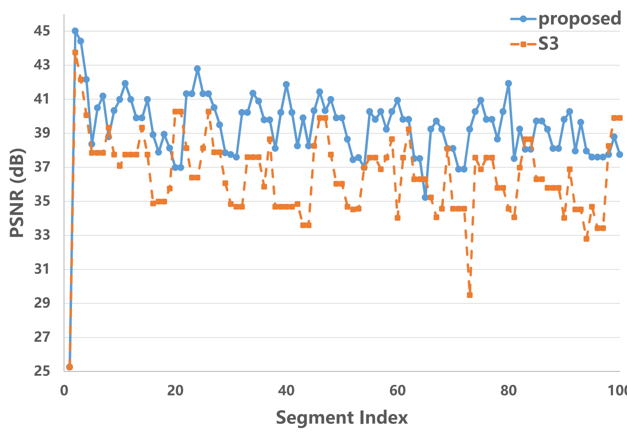


Fig. 12. Comparison of the playback quality between "proposed" and "S3".

Fig. 11 shows that the playback quality of "S2" is higher than "proposed". In fact, it depends on the available buffer length. As shown in Fig. 8, the average available buffer length in "S2" is lower than "proposed", because it has more states with the available buffer length lower than Ω^*. Nevertheless, it cannot prove that "S2" is a better solution, since shorter available buffer length also lead to more frequent starvation event. As shown in the statistical results in Table V, the average playback quality is 39.24dB in "proposed", and it is 40.80dB in "S2". The average playback quality is 36.71dB and 36.74dB for "S1" and "S3", respectively. On the other hand, the starvation probability demonstrates the advantage of "proposed" over "S2". The starvation probability for "proposed" is 0.74%, which is much higher for "S2", being 2.86%. Here, we can see that the "S2" method maintains shorter buffer in most cases, but accordingly more frequent starvation events occur. Based on the reported results, it can be found that "Buffer-based" method is better than the proposed "QoE-based" method in terms of quality variation, but "S1" cannot provide a good playback quality and "S2" leads to frequency starvation events. The root cause is that buffer-based method tries to control the buffer in a small range while "QoE-based" method tries to limit the buffer near to the calculated ideal buffer length. The ideal buffer length calculated using (19) is to balance the three factors, e.g., the average

TABLE V
STATISTICAL QOE PERFORMANCE OF IMPLEMENTED SOLUTIONS

Approach	Average Quality(dB)	Quality Variation(dB)	Starvation Probability	QoE Value
proposed	39.24	0.78	0.74%	**37.50**
S1	36.71	0.64	0.44%	35.62
S2	**40.80**	**0.47**	2.86%	34.93
S3	36.32	0.94	0.60%	34.81

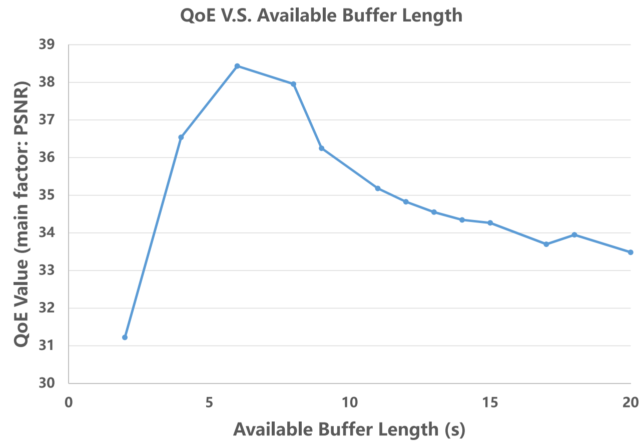


Fig. 13. QoE performance under different available buffer length settings.

playback quality, the average quality variation and starvation probability.

Fig. 13 provides the QoE performance under different Ω^*, with μ being 200. The results show that the calculated ideal buffer length, which is 6s, gives the best QoE performance in the "proposed" approach. It matches the expected trend of the analytical framework. When the available buffer length is short, we can obtain high playback quality for the pre-fetched segments. However, frequent starvation events would occur. With the increase of available buffer length, the playback quality decreases, but the starvation probability also reduces. Therefore, the ideal buffer length in our proposed framework is very important to balance the playback quality and the starvation probability.

We also discuss the correlation among μ, Ω^* and QoE value in the following experiments. μ generally denotes the significance of starvation in the QoE model. Ω^* is the ideal buffer length which could ensure the best QoE performance. To prove the equation (19), we tried to find the ideal buffer length which leads to the best QoE value by experiments. As shown in Fig. 14, the blue line is the found Ω^* by experiments with different μ and red line is calculated as (19). It proves that the correlation between μ and Ω^* is close to the distribution of (19). Correspondingly, several QoE values are also provided in Fig. 15, where μ is set as 50, 100 and 300, respectively. Similar as Fig. 13, the results in Fig. 15 prove that by changing the weight of the starvation term μ in QoE model, the Ω^* that leads to the best QoE value will also change. With different settings of μ (50, 100 and 300), the QoE performance comparison between "proposed" and "S1", "S2", "S3" are reported in Table VI. It proves the effectiveness of the "proposed" method, even for different μ.

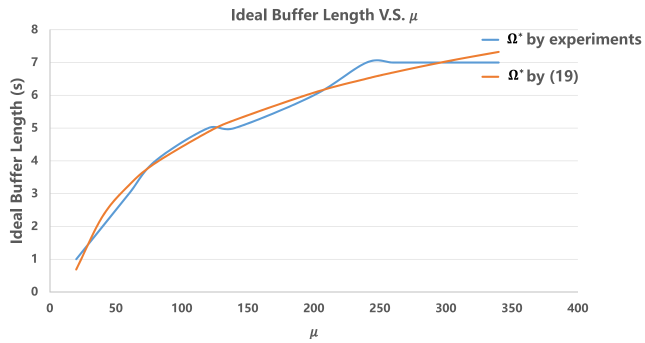


Fig. 14. Ideal buffer length Ω^* with different μ: the blue line represent the actual settings of Ω^* corresponding to μ, and the red line is calculated as (19).

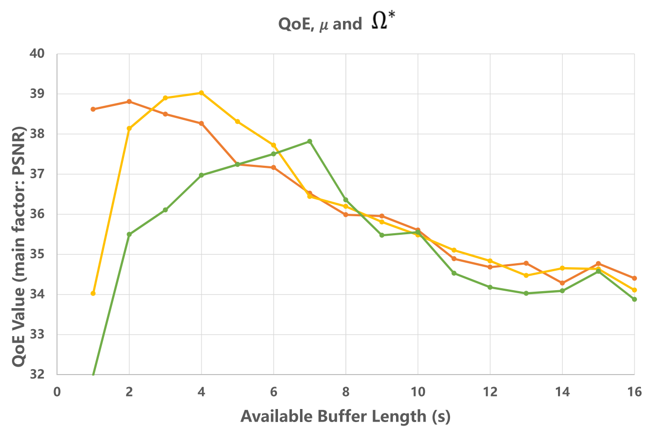


Fig. 15. QoE, Ideal buffer length with different μ, where red line is with $\mu = 50$, yellow line is with $\mu = 100$, and green line is with $\mu = 300$.

TABLE VI
QoE PERFORMANCE OF IMPLEMENTED SOLUTIONS WITH DIFFERENT μ

μ	Approach	Average Quality(dB)	Quality Variation(dB)	Starvation Probability	QoE
50	proposed	39.48	0.91	0.73%	**38.81**
	S1	36.94	0.43	0.40%	36.55
	S2	42.03	0.52	9.01%	36.45
	S3	36.45	1.25	1.06%	35.51
100	proposed	40.24	0.87	0.92%	**39.03**
	S1	38.17	0.40	0.74%	37.15
	S2	41.33	0.47	7.82%	32.79
	S3	38.47	1.00	1.32%	36.81
300	proposed	39.53	0.80	0.48%	**37.81**
	S1	36.80	0.44	0.49%	35.18
	S2	41.03	0.52	2.03%	34.77
	S3	36.37	1.18	0.32%	35.01

VII. CONCLUSION

In this paper, we propose and implement a novel multiview video stream adaptation strategy for interactive video playback which allows client to navigate among multiple viewpoints. To simplify the user's navigation behavior, our solution firstly introduce a constraint viewpoint switching model. Then, a switching probability estimation model is designed to guide the bandwidth sharing of all possible switching video streaming. Finally, the implemented framework balances the average quality, average quality variation and starvation probability as a QoE model, which is used to adapt the multiview stream for improving the performance of the whole playback session. Experimental results show that the proposed framework is able to pre-fetch the multiview stream at high qualities, moreover, it can effectively reduce the starvation event in case of frequent viewpoint switching.

ACKNOWLEDGMENT

The authors would like to thank Prof. Tammam Tillo in the Faculty of Computer Science, Free University of Bozen-Bolzano, Italy.

REFERENCES

[1] "Information technology—Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation description and segment formats," MPEG, Denver, CO, USA, Rep. ISO/IEC 23009-1:2012, 2012.

[2] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 133–144.

[3] T. Maugey and P. Frossard, "Interactive multiview video system with low complexity 2D look around at decoder," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1070–1082, Aug. 2013.

[4] Z. Liu, G. Cheung, and Y. Ji, "Unified distributed source coding frames for interactive multiview video streaming," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, 2012, pp. 2048–2053.

[5] M. Zhao *et al.*, "A cloud-assisted DASH-based scalable interactive multiview video streaming framework," in *Proc. Picture Coding Symp. (PCS)*, 2015, pp. 221–226.

[6] J. Xiao *et al.*, "Scalable bit allocation between texture and depth views for 3-D video streaming over heterogeneous networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 139–152, Jan. 2015.

[7] T. Su, A. Sobhani, A. Yassine, S. Shirmohammadi, and A. Javadtalab, "A DASH-based HEVC multi-view video streaming system," *J. Real Time Image Process.*, vol. 12, no. 2, pp. 329–342, 2016.

[8] D. Yun and K. Chung, "DASH-based multi-view video streaming system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1974–1980, Aug. 2018.

[9] N. Carlsson, D. Eager, V. Krishnamoorthi, and T. Polishchuk, "Optimized adaptive streaming of multi-video stream bundles," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1637–1653, Jul. 2017.

[10] L. Yu, T. Tillo, J. Xiao, and M. Grangetto, "Convolutional neural network for intermediate view enhancement in multiview streaming," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 15–28, Jan. 2018.

[11] X. Zhang, L. Toni, P. Frossard, Y. Zhao, and C. Lin, "Adaptive streaming in interactive multiview video systems," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[12] Y. Xu, Y. Zhou, and D.-M. Chiu, "Analytical QoE models for bit-rate switching in dynamic adaptive streaming systems," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2734–2748, Dec. 2014.

[13] Z. Liu, G. Cheung, and Y. Ji, "Optimizing distributed source coding for interactive multiview video streaming over lossy networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1781–1794, Oct. 2013.

[14] T. Fujihashi, Z. Pan, and T. Watanabe, "UMSM: A traffic reduction method on multi-view video streaming for multiple users," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 228–241, Jan. 2014.

[15] A. Hamza and M. Hefeeda, "A DASH-based free viewpoint video streaming system," in *Proc. Netw. Oper. Syst. Support Digit. Audio Video Workshop*, Mar. 2014, pp. 55–60.

[16] B. Oztas, M. T. Pourazad, P. Nasiopoulos, I. Sodagar, and V. C. M. Leung, "A rate adaptation approach for streaming multiview plus depth content," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Honolulu, HI, USA, 2014, pp. 1006–1010.

[17] L. Toni and P. Frossard, "Optimal representations for adaptive streaming in interactive multiview video systems," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2775–2787, Dec. 2017.

[18] A. Bentaleb, A. C. Begen, and R. Zimmermann, "QoE-aware bandwidth broker for HTTP adaptive streaming flows in an SDN-enabled HFC network," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 575–589, Jun. 2018.

[19] V. P. K. M and S. Mahapatra, "Quality of experience driven rate adaptation for adaptive HTTP streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 602–620, Jun. 2018.

[20] H. Yuan, H. Fu, J. Liu, J. Hou, and S. Kwong, "Non-cooperative game theory based rate adaptation for dynamic video streaming over HTTP," *IEEE Trans. Mobile Comput.*, vol. 17, no. 10, pp. 2334–2348, Oct. 2018.

[21] H. Yuan, X. Wei, F. Yang, J. Xiao, and S. Kwong, "Cooperative bargaining game-based multiuser bandwidth allocation for dynamic adaptive streaming over HTTP," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 183–197, Jan. 2018.

[22] 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.*, 2011, pp. 157–168.

[23] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 169–174.

[24] B. Zhou, J. Wang, Z. Zou, and J. Wen, "Bandwidth estimation and rate adaptation in HTTP streaming," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Maui, HI, USA, Jan. 2012, pp. 734–738.

[25] T.-Y. Huang, R. Johari, and N. McKeown, "Downton abbey without the hiccups: Buffer-based rate adaptation for HTTP video streaming," in *Proc. ACM SIGCOMM Workshop Future Human Centric Multimedia Netw. (FhMN)*, 2013, pp. 9–14.

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

[27] W. Huang *et al.*, "Buffer state is enough: Simplifying the design of QoE-aware HTTP adaptive video streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 590–601, Jun. 2018.

[28] D. Z. Rodríguez, R. L. Rosa, E. C. Alfaia, J. I. Abrahão, and G. Bressan, "Video quality metric for streaming service using DASH standard," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 628–639, Sep. 2016.

[29] L. Qian *et al.*, "A QoE-driven encoder adaptation scheme for multi-user video streaming in wireless networks," *IEEE Trans. Broadcast.*, vol. 63, no. 1, pp. 20–31, Mar. 2017.

[30] L. Yu, T. Tillo, and J. Xiao, "QoE-driven dynamic adaptive video streaming strategy with future information," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 523–534, Sep. 2017.

[31] Y. Gong, F. Yang, L. Huang, and S. Su, "Model-based approach to measuring quality of experience," in *Proc. 1st Int. Conf. Emerg. Netw. Intell.*, 2009, pp. 29–32.

[32] A. Perkis, S. Munkeby, and O. I. Hillestad, "A model for measuring quality of experience," in *Proc. 7th Nordic Signal Process. Symp. (NORSIG)*, 2006, pp. 198–201.

[33] A. Khan, L. Sun, E. Jammeh, and E. Ifeachor, "Quality of experience-driven adaptation scheme for video applications over wireless networks," *IET Commun.*, vol. 4, no. 11, pp. 1337–1347, Jul. 2010.

[34] 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.

[35] H. Kimata, K. Fukazawa, A. Kameda, Y. Yamaguchi, and N. Matsuura, "Interactive 3D multi-angle live streaming system," in *Proc. IEEE 15th Int. Symp. Consum. Electron. (ISCE)*, Jun. 2011, pp. 576–579.

[36] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[37] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[38] S. Wang, A. Rehman, K. Zeng, J. Wang, and Z. Wang, "SSIM-motivated two-pass VBR coding for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2189–2203, Oct. 2017.

[39] A. Ortega, "Optimization techniques for adaptive quantization of image and video under delay constraints," Ph.D. dissertation, Grad. School Arts Sci., Columbia Univ., New York, NY, USA, 1994.

[40] *DASH Industry Forum*. Accessed Apr. 2013. [Online]. Available: www.dashif.org

[41] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.

[42] A. M. Tourapis, A. Leontaris, K. Shring, and G. Sullivan, *H.264/14496-10 AVC Reference Software Manual*, document JVT-AE010, JVT, Jul. 2009.

[43] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[44] X. Yin, V. Sekar, and B. Sinopoli, "Toward a principled framework to design dynamic adaptive streaming algorithms over HTTP," in *Proc. 13th ACM Workshop Hot Topics Netw. (HotNets-XIII)*, 2014, pp. 1–9.

Chao Yao received the B.S. degree in computer science from Beijing Jiaotong University (BJTU), Beijing, China, in 2009 and the Ph.D. degree from the Institute of Information Science, BJTU, in 2016. From 2014 to 2015, he served as a visiting Ph.D. student with the Ecole Polytechnique Federale de Lausanne, Switzerland. From 2016 to 2018, he served as a Post-Doctoral with the Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Beijing. His current research interests include image and video processing, computer vision, and deep learning.

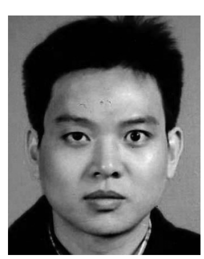
Jimin Xiao (M'15) received the B.S. and M.E. degrees in telecommunication engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2007, respectively, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K., in 2013.

From 2013 to 2014, he was a Senior Researcher with the Department of Signal Processing, Tampere University of Technology, Tampere, Finland, and an External Researcher with the Nokia Research Center, Tampere. Since 2014, he has been a Faculty Member with Xian Jiaotong-Liverpool University, Suzhou, China. His research interests include image and video processing, computer vision, and deep learning.

Yao Zhao (M'06–SM'12) received the B.S. degree from the Radio Engineering Department, Fuzhou University, Fuzhou, China, in 1989, the M.E. degree from the Radio Engineering Department, Southeast University, Nanjing, China, in 1992, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996.

He became an Associate Professor with BJTU in 1998 and then became a Professor in 2001. From 2001 to 2002, he was a Senior Research Fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. In 2015, he visited the Swiss Federal Institute of Technology, Lausanne, Switzerland. From 2017 to 2018, he visited the University of Southern California. He is currently the Director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding.

Dr. Zhao serves on the Editorial Boards of several international journals, including as an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and IEEE SIGNAL PROCESSING LETTERS, and an Area Editor of *Signal Processing: Image Communication*. He was named as a Distinguished Young Scholar by the National Science Foundation of China in 2010, and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013. He is a fellow of the IET.



Anlong Ming received the Ph.D. degree from the School of Computer Science, Beijing University of Posts and Telecommunications, where he is currently a Professor. His research interests include robot vision, pattern recognition, and visual arts.