Timely and Accurate Bitrate Switching in HTTP Adaptive Streaming with Date-Driven I-frame Prediction

Tongtong Feng, Qi Qi, *Senior Member, IEEE,* Jingyu Wang, *Senior Member, IEEE,* Jianxin Liao, *Member, IEEE,* Jiangchuan Liu, *Fellow, IEEE*

***Abstract*—In today's Internet, bandwidth dynamics are inevitable, and hence, the bitrate for live streaming applications should also be dynamically adjusted. However, in existing HTTP-based adaptive streaming (HAS), bitrate switching can only be performed at segment boundaries, making decisions unresponsive and often inaccurate. In this paper, we start from a close investigation on the impact of the segment length in HAS and accordingly present *VHAS*, an extension towards intelligent variable-length segmentation, which makes client-side decisions based on the massive amount of real-time information from the network and viewers. VHAS implements a smart trigger mechanism that balances accuracy and overhead for variable-length segmentation. We further develop an adaptive bitrate switching algorithm with data-driven I-frame prediction, which is tailored to individual viewers to minimize bitrate mismatches. We evaluate VHAS via extensive trace-driven simulations, and our results demonstrate that compared with state-of-the-art solutions, VHAS achieves 15% - 49% gains in QoE, with a noticeable bandwidth reduction of 37% - 57%.**

***Index Terms*—HTTP Adaptive Streaming, Segmentation, Bitrate Switching, I-frame Prediction, Reinforcement Learning**

I. Introduction

REAL-TIME live streaming applications, such as YouTube Live, Twitter's Periscope, Huya.tv and Douyu.tv, have been experiencing dramatic growth over the past decade, attracting millions of active users [1]. Bandwidth dynamics are inevitable in today's Internet, and hence, the bitrate for live streaming applications should be adaptive as well. Existing HTTP-based adaptive streaming (HAS) seeks to dynamically adjust the bitrate in real time yet being fully compatible with dominating HTTP protocol for Web content distribution [2]–[4]. It has seen great success in real-world deployment, e.g., Microsoft Smooth Streaming (MSS[1]), Apple's HTTP Live Streaming (HLS[2]), Adobe HDS[3], and Akamai HD[4], to name but a few examples.

In the standard HAS (see Fig. 1(a)), each video is encoded into streams of multiple discrete bitrates and each stream is partitioned into multiple segments. A client-side video player can dynamically choose a video bitrate that best matches the current network state. The switch to a different video bitrate, however, occurs only at the boundary of a segment, and the corresponding delay can be quite long, e.g., 2-8 seconds [5]. This coarse granularity makes switch decisions unresponsive and often inaccurate given the much quicker network bandwidth changes (typically an order of magnitude shorter) [6]. An enhancement is to divide a segment into shorter ones by inserting intra-coded frames (I-frames) [7] (see Fig. 1(b)), where each I-frame indicates the beginning of a group of pictures (GOP), i.e., a segment boundary. This effectively accelerates bitrate switching and the better responsiveness improves accuracy as well. However, since the size of an I-frame is 8-10 times greater than that of a P- or B-frame, the overhead can be significant if not well-controlled.

In this paper, we propose *VHAS* (see Fig. 1(c)), an extension towards fast and accurate bitrate switching with minimize bandwidth overhead. In VHAS, each video is encoded into streams of multiple discrete bitrates and each stream is partitioned into multiple *variable-length* segments. VHAS then implements a smart trigger mechanism that balances accuracy and overhead for variable-length segmentation based on an Adaptive Bitrate (ABR) algorithm. Existing ABR algorithms improve end users' quality of experience (QoE) by considering network bandwidth and client buffer information [8]–[15]. Recent works have also applied such advanced tools as *reinforcement learning* (RL) [5], [16]. However, these ABR algorithms are only applicable to the standard HAS, in which streams of different bitrates are of the same segment structure, i.e., their segment boundaries are well aligned, and hence a client can directly switch across streams if the next frame is an I-frame, without boundary check. For VHAS, the segments are of variable lengths and hence each stream will have a distinct segment structure. A decision to adjust the video bitrate, i.e., switch to another stream, cannot be simply triggered by observing the segment boundary of the

This work was supported in part by the National Natural Science Foundation of China under Grants 62071067, 62171057, 62101064, and 62001054, in part by the Ministry of Education and China Mobile Joint Fund MCM20200202, and in part by the BUPT Excellent Ph.D. Students Foundation CX2020230.

T. Feng, Q. Qi, J. Wang and J. Liao are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China and with EBUPT.COM, Beijing 100191, China (e-mail: ftt@bupt.edu.cn; qiqi8266@bupt.edu.cn; wangjingyu@bupt.edu.cn; liaojx@bupt.edu.cn). J. Liu is with Simon Fraser University, BC TASC I 9005, Canada (e-mail: jcliu@cs.sfu.ca).

The corresponding authors are J. Wang (wangjingyu@bupt.edu.cn), J. Liao (liaojx@bupt.edu.cn) and J. Liu (jcliu@cs.sfu.ca).

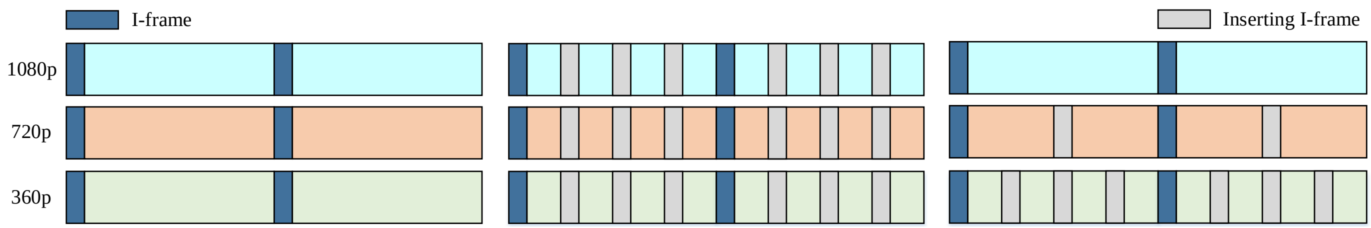
[1] http://go.microsoft.com/?linkid=9682896.

[2] https://developer.apple.com/resources/http-streaming.

[3] http://www.adobe.com/products/hds-dynamic-streaming.html.

[4] http://wwwns.akamai.com/hdnetwork/demo/index.html.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3165381, IEEE Transactions on Multimedia



(a) Standard HAS (b) Short-segment HAS (c) Variable-length HAS

Fig. 1. HTTP-based adaptive streaming.

current stream. Suspending the decision till the next I-frame in the target stream however would introduce a noticeable mismatch against the expected optimal bitrate. We address this challenge by a data-driven I-frame prediction that is based on the massive amount of real-time information from the network and viewers. Through cascading a deep neural network (DNN) [17] and RL network, our client-side decision is tailored to individual viewers to minimize the bitrate mismatches.

We evaluate VHAS via extensive trace-driven simulations with three real-world datasets. Our results demonstrate that compared with state-of-the-art solutions, VHAS achieves 15% - 49% gains in QoE, with a noticeable bandwidth reduction of 37% - 57%; meanwhile, VHAS also yields a good effect under various network conditions.

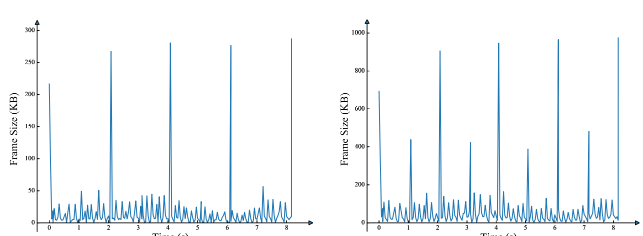
The remainder of the paper is organized as follows. Section II describes the design details of VHAS. Section III presents the implementation details and evaluation results of VHAS. Section IV introduces related work. Conclusions and future work are discussed in Section V.

II. Variable-Length HTTP Adaptive Streaming

In this section, we describe the design details of variable-length HTTP adaptive streaming (VHAS), including how to divide video streams into variable-length segments and how to use a trigger mechanism to achieve timely and accurate bitrate switching.

A. *Variable-Length Segmentation*

In the standard HAS (see Fig. 1(a)), inserting more I-frames can increase the number of segments and effectively improve the timeliness of bitrate switching, which is the essential strategy of the short-segment HAS (see Fig. 1(b)) [7]. Unfortunately, it will also significantly increase the bandwidth overhead, particularly for video streams of high bitrates (see Fig. 2). For example, inserting an I-frame for a 500 Kbps video only increases the video size by 250 KB; however, for an 1850 Kbps video, inserting an I-frame increases the bandwidth overhead by at least 800 KB. To manage the tradeoff between accurate bitrate switching and bandwidth overhead, we introduce variable-length segmentation in which a video stream of a lower bitrate is divided into shorter segments by inserting more I-frames (see Fig. 1(c)). We use function $y = K^x$ to determine the number of I-frames inserted for video streams at different bitrates.



(a) 500 Kbps. (b) 1850 Kbps.

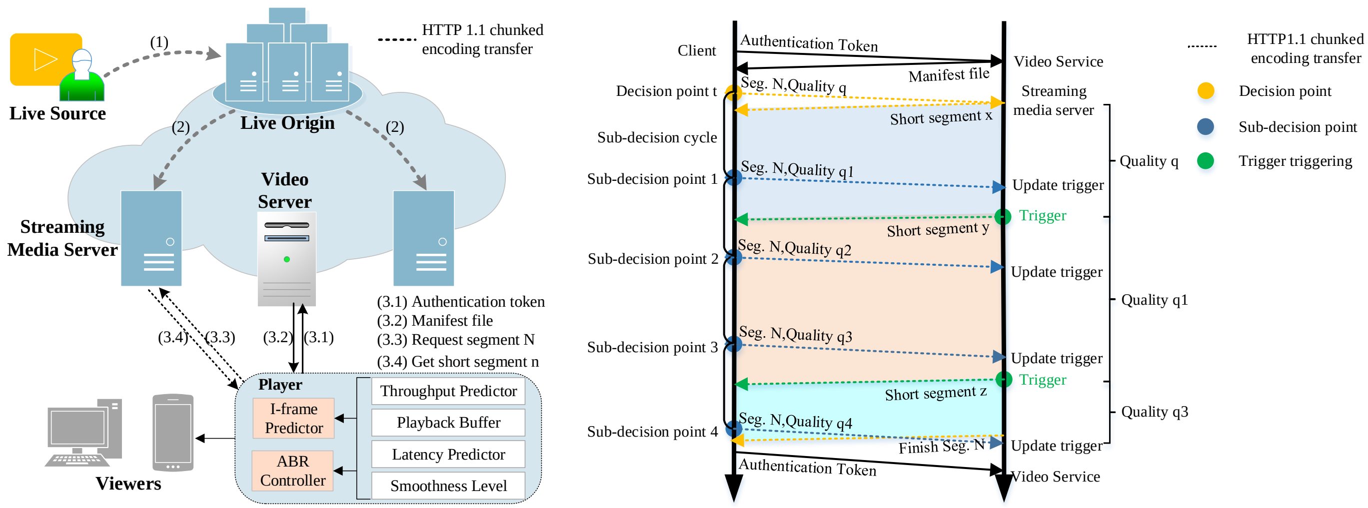
Fig. 2. The comparison of frame sizes in video at different bitrates.

Fig. 3 demonstrates VHAS's end-to-end data transmission process: (1) An encoder integrated into a live source partitions video streams into standard segments, which is the same as the segmentation operation in the standard HAS. Then the encoder constructs variable-length segments for each standard segment by inserted I-frames and pushes them to a live origin for redistribution; (2) The live origin pushes the variable-length segments to streaming media servers and updates the manifest files of this video in video servers (video authentication service providers); (3) Upon the request of a client, the video server responds with a manifest file that lists all available video bitrates and the position information of all downloadable variable-length segments. At decision point t (the standard segment boundary), the player of the client calculates video bitrate q by an ABR algorithm according to the current state information and requests the newest variable-length segment N of quality q from the streaming media server. In the download process for segment N, we start from decision point t to construct multiple sub-decision points t_i with equal intervals (referred to as a sub-decision period, e.g., 0.5s). At t_i, the player uses the ABR algorithm to make a bitrate decision q_i, sends the selected bitrate q_i to the streaming media server, and updates the bitrate switching trigger. Specifically, when the selected bitrate q_i video stream has an I-frame (located at short segment boundary) in the next sub-decision period, the bitrate switching triggered from bitrate q to q_i. The specific decision process is shown in Fig. 3(b).

B. *QoE Optimization*

We now consider the overall system optimization with our variable-length segmentation. Table I lists the notations used for our modelling.

We assume that in a city-level region the network service provider has one CDN server (the live origin) c and a set of



(a) Data transmission process. (b) Sub-decision process.

Fig. 3. Variable-length HTTP adaptive streaming.

streaming media servers $\mathbf{S} = \{1, 2, ..., S\}$ distributed across the city. All of them are connected via the backhaul network. Each streaming media server is capable of hosting all the video streams at bitrates $\mathbf{V} = \{1, 2, ..., V\}$. We assume that U viewers $\mathbf{U} = \{1, 2, ..., U\}$ have viewing requests at different video bitrates. Based on each viewer's individual bandwidth condition, the video bitrate requested by viewer u is denoted as $\phi^{(u,d)} = v^{'}$, where $d = 0$ (resp. 1) signifies at each (resp. sub) decision point. We use a binary variable X to denote whether the trigger in the streaming media server calls bitrate switching, where $X^{(u,d)}_{(v,v')} = 1$ (resp. 0) indicates that the trigger (resp. not) calls bitrate switching from v to $v^{'}$, where $d = 0$ (resp. 1) also signifies at each (resp. sub) decision point.

We consider four QoE metrics for each viewer, i.e., video quality [5], [14], video latency [18], bitrate smoothness level [5], bitrate mismatch level [18]. We can calculate the video quality $\Theta^{(u)}$ of viewer u as follows:

$$\begin{aligned}\Theta^{(u)} =& \sum_{n=1}^{N}\sum_{f=1}^{F}\phi^{(u,0)}_{(n,f)}D_{(n,f)}+\\ &\sum_{n=1}^{N}\sum_{f=1}^{F}X^{(u,1)}_{(v,v')}(\phi^{(u,1)}_{(n,f)}-\phi^{(u,0)}_{(n,f)})D_{(n,f)}\end{aligned} \tag{1}$$

where n represents the number of segments that viewer u requests, $n \in \{1, 2, ..., N\}$; (n, f) represents the f-th frame of the segment n, $f \in \{1, 2, ..., F\}$; and $D_{(n,f)}$ represents the duration time of the (n, f) frame. The first part of the equation represents the video quality if the trigger only performs bitrate switching at each decision point, which is equivalent to running only in the standard HAS, while the second part signifies how much the user experience can be improved by VHAS at each sub-decision point.

The video latency $\Gamma^{(u)}$ of viewer u can be calculated as:

$$\begin{aligned}\Gamma^{(u)} =& \sum_{n=1}^{N}\sum_{f=1}^{F}(T^{(u,0)}_{(v^*,v)}+C^{(u)}_{(n,f)}+L^{(u,s)}_{(n,f)}+C^{(s)}_{(n,f)})+\\ &\sum_{n=1}^{N}X^{(u,1)}_{(v,v')}(T^{(u,1)}_{(v^*,v')}-T^{(u,0)}_{(v^*,v)})\end{aligned} \tag{2}$$

where $T^{(u,e)}_{(v^*,v)}$ is the transcoding latency from v^* to v, where $e = 0$ (resp. 1) means transcoding to a fixed-length (variable-length) segment. Note that if $v = v^*$, there is no need to transcode ($T^{(u,e)}_{(v^*,v)} = 0$). $C^{(u)}_{(n,f)}$ indicates the player's cache size. $L^{(u,s)}_{(n,f)}$ is the transmission latency between viewer u and server s. $C^{(s)}_{(n,f)}$ is the cache size in streaming media server s. The first part of the equation represents the latency overhead if the transcoding server transcodes only at each decision point, while the second part incorporates the transcoding overhead added by transcoding to the variable-length segment.

The bitrate smoothness level $\Phi^{(u)}$ is calculated as:

$$\Phi^{(u)} = \sum_{n=1}^{N}\sum_{d=0}X^{(u,d)}_{(v,v')}\frac{1}{|I^{(v')}_{(n,d)}-I^{(v)}_{(n,d)}|+1} \tag{3}$$

where $I^{(v)}_{(n,d)}$ represents the level of video bitrate requested by viewer u at each decision point or sub-decision point. For instance, for a video that is encoded by an H.264/MPEG-4 codec at bitrates {300, 750, 1200, 1850, 2850, 4300} Kbps, we have $I^{(v)}$ = {0, 1, 2, 3, 4, 5}. The bitrate smoothness level $\Phi^{(u)}$ can avoid a decrease in the user's viewing experience that would result from the sharp jitter of video bitrate.

The bitrate mismatch level $\Psi^{(u)}$ is calculated as:

$$\Psi^{(u)} = \sum_{n=1}^{N}\sum_{f=1}^{F}(1-X^{(u,d)}_{(v,v')}) \tag{4}$$

It indicates the matching degree between the client's requested bitrate ($v' = \phi^{(u,d)}$) and received bitrate (v). At each decision point, since segments across bitrates are aligned and the required bitrate switching operation can always be performed, $\Psi^{(u)}$ is equal to 0. In a sub-decision point, it however depends on the accuracy of the I-frame prediction in the next sub-decision period. The lower $\Psi^{(u)}$ is, the better the matching of the video bitrate received by the viewer u to that requested.

In addition to the QoE metrics for each viewer, we also consider the overall the bandwidth overhead for the construction of variable-length segments may increase bandwidth demand.

$$\Lambda^{(u)} = \sum_{n=1}^{N}\sum_{f=1}^{F} B_{(n,f)}^{(u,0)} + \sum_{n=1}^{N}\sum_{f=1}^{F} X_{(v,v')}^{(u,1)} (B_{(n,f)}^{(u,1)} - B_{(n,f)}^{(u,0)}) \tag{5}$$

where $B_{(n,f)}^{(u,d)}$ is the bandwidth consumption during the (n, f) frame. The first part of the equation represents the bandwidth overhead if we just use the fixed-length segmentation for data transmission. The second part shows the bandwidth overhead added by the construction of variable-length segments, which indicates how much the user experience can be improved by our VHAS relative to the standard HAS.

Integrating viewers' QoE demands (Eq. 1, Eq. 2, Eq. 3, and Eq. 4) and the bandwidth overhead (Eq. 5), we obtain the following optimization objective (Ω) that minimizes the sum of the penalties, including the QoE penalty (the penalty in video quality, video latency, bitrate smoothness, and bitrate matching) and the penalty for the bandwidth overhead:

$$Min : \alpha(-\alpha_1^{(u)}\Theta^{(u)} + \alpha_2^{(u)}\Gamma^{(u)} - \alpha_3^{(u)}\Phi^{(u)} + \alpha_4^{(u)}\Psi^{(u)}) + \beta\Lambda^{(u)} \tag{6}$$

$s.t.$

$$0 < \phi^{(u,d)} \leq v^*, \forall d \tag{7}$$

$$X_{(v,v')}^{(u,d)} = 1, \exists v = v' \tag{8}$$

$$T_{(v^*,v)}^{(u,e)} = 0, \exists v = v^* \tag{9}$$

where α and β are weighting parameters to tune the QoE penalty and bandwidth overhead penalty, respectively, and $\alpha_1^{(u)}$, $\alpha_2^{(u)}$, $\alpha_3^{(u)}$ and $\alpha_4^{(u)}$ are personalized QoE preference factors for viewer u. Eq. 7 guarantees that the video bitrate requested by viewer u can be transcoded by a transcoding server. When the bitrate requested by viewer u is equal to the currently executed bitrate, Eq. 8 indicates that bitrate switch operations have been called. When the requested bitrate of viewer u is equal to the highest bitrate, Eq. 9 indicates that there is no need to transcode.

In summary, as the segment length becomes shorter, the timeliness of bitrate switching is improved, but the transcoding latency overhead (Eq. 2) and the bandwidth overhead (Eq. 5) also increase; the higher the bitrate mismatch level $\Psi^{(u)}$ (Eq. 4), the worse the performance of VHAS. The ABR algorithm must balance a variety of QoE goals (Eq. 6), such as maximizing video quality, minimizing video latency, maintaining video quality smoothness, minimizing bitrate mismatch level and minimizing bandwidth resource usage, and many of them are inherently conflicting. We accordingly design an adaptive bitrate switching algorithm with data-driven I-frame prediction to solve the challenges.

TABLE I
NOTATIONS USED IN THIS SUBSECTION

c	live origin
S	the streaming media server list
N	the segment list
F	the frame list in the segment
V	the video bitrate list
U	the set of all the viewers
d	whether current decision point is a sub-decision point
e	whether transcode to a variable-length segment
$\phi^{(u,d)}$	the request video bitrate of the viewer u
$X_{(v,v')}^{(u,d)}$	whether the trigger calls bitrate switching from v to v'
$\Theta^{(u)}$	the video quality for the viewer u
$D_{(n,f)}$	the duration time of the (n, f) frame
$\Gamma^{(u)}$	the video latency for viewer u
$T_{(v^*,v)}^{(u,e)}$	the transcoding latency from v^* to v
$L^{(u,s)}$	the transmission latency between viewer u and server s
$C^{(u)}$	the player cache size for the viewer u
$C^{(s)}$	the streaming media server's cache size for the viewer u
$\Phi^{(u)}$	the bitrate smoothness level for the viewer u
$I(v)$	the level of the video bitrate request of the viewer u
$\Psi^{(u)}$	the bitrate mismatch level for the viewer u
$\Lambda^{(u)}$	the total bandwidth overhead to the viewer u
$B^{(u,d)}$	the bandwidth consumption duration the decision period d
α,β	the weighted parameters to tune the QoE penalty and bandwidth overhead penalty
$\alpha_i^{(u)}$	the personalized QoE preference factors for viewer u

C. Adaptive Bitrate Switching Algorithm

The adaptive bitrate switching algorithm with data-driven I-frame prediction cascades two neural networks (see Fig. 4): one is an I-frame prediction network using a deep neural network (DNN) [17], and the other is a bitrate selection network using reinforcement learning (RL) [19].

I-frame prediction network. It predicts which video streams will appear in the I-frames in the next sub-decision period and assists the bitrate selection network to make an optimal bitrate decision. It is the core to reduce the bitrate mismatching ratio.

Input. The input of the I-frame prediction network M_t^1 consists of two parts: one is client's state information C_t^1, the other is streaming media server's state information S_t^1.

$$M_t^1 = (C_t^1|S_t^1) = (h_t^1, m_t^1, e_t^1, q_t^1, d_t^1, \vec{g}_t^1|k_t^1, o_t^1) \tag{10}$$

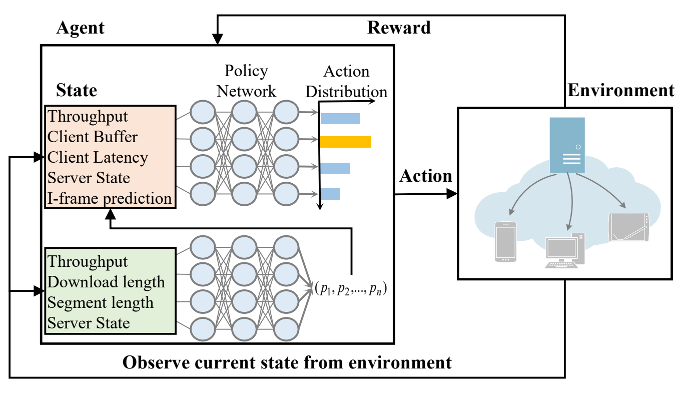


Fig. 4. Architectural overview of the adaptive bitrate switching algorithm with data-driven I-frame prediction.

where h_t^1 indicates the predicted network throughput; m_t^1 signifies the mean of the last K network throughput records; e_t^1 indicates the predicted probability of the network throughput increase at the next decision period; q_t^1 is the current video bitrate; d_t^1 indicates the remaining proportion of the current segment to be downloaded. $\vec{g}_t^1$ represents the sub-decision period; k_t^1 and o_t^1 indicate the video cache duration and the video rebuffering duration in the streaming media server, respectively. To calculate e_t^1, we define four network throughput change modes for the four consecutive network throughput records: up_up_up, up_down_up, down_up_up, down_down_up; we then calculate the probabilities of the four modes appearing in the past N historical network throughput records by a sliding window with window size of four; finally, according to the last three network throughput records, we predict the probability of the network throughput increase at the next decision period.

Output. The output $P = \{p_v\}$ is a probability vector, where $v \in V$ represents the video bitrates; p_v represents the probability that an I-frame will appear in the video stream with bitrate v in the next sub-decision period.

$$p_v = \begin{cases} \frac{1}{Q} & v \in Q \\ 0 & v \notin Q, v \in V \end{cases} \tag{11}$$

where Q represents the set of video bitrates that I-frames will appear in the next sub-decision period.

Policy. As in previous studies [16], we use a mean square error function as the loss function.

Bitrate selection network. Many RL algorithms have been used to train learning agents, including A2C [20], PPO [21], DQN [22], ACKTR [19], etc. In our work, we are particularly interested in ACKTR, which uses a very efficient Kronecker-factored approximate curvature (K-FAC) optimizer to estimate the natural gradient, and hence each update becomes much faster than others.

State. Similar to the I-frame prediction network, its state information M_t^2 comes from client C_t^2 and streaming media server S_t^2.

$$M_t^2 = (C_t^2|S_t^2) = (h_t^1, m_t^1, e_t^1, b_t^2, f_t^2, q_t^1, l_t^2, r_t^2|n_t^2, k_t^1, o_t^1) \tag{12}$$

where b_t^2 represents the client's playback buffer occupancy; f_t^2 indicates whether the client's playback buffer is empty; l_t^2 is the transmission latency between the client and the streaming media server; r_t^2 represents the client's rebuffering duration; n_t^2 represents the remaining number of segments in the streaming media server; and n_t^2 is given by $(l_t^2 - b_t^2)*$frame_rate.

Action. The action here is the optimal video bitrate adapted to the current environment.

Reward. The goal of the bitrate selection network is to achieve the maximum QoE, which is equivalent to minimizing optimization objective (Ω). As shown in subsection II-B, we have $QoE = -\Omega$.

$$QoE = \alpha(\alpha_1^{(u)}\Theta^{(u)} - \alpha_2^{(u)}\Gamma^{(u)} + \alpha_3^{(u)}\Phi^{(u)} - \alpha_4^{(u)}\Psi^{(u)}) - \beta\Lambda^{(u)} \tag{13}$$

where α and β are weighting parameters to tune the QoE penalty and bandwidth overhead penalty, respectively, and $\alpha_1^{(u)}$, $\alpha_2^{(u)}$, $\alpha_3^{(u)}$ and $\alpha_4^{(u)}$ are personalized QoE preference factors for viewer u.

TABLE II
PERFORMANCE COMPARISON OF THE EXPONENTIAL FUNCTION WITH K.

K	1	2	3	4
Average QoE	5.18	6.37	5.72	3.54

TABLE III
THE NUMBER OF HIDDEN LAYERS IN THE I-FRAME PREDICTION NETWORK.

Number of hidden layers	1	2	3	4	5
Average QoE	5.865	6.158	6.371	6.359	6.362

TABLE IV
THE NUMBER OF HIDDEN LAYERS IN THE BITRATE SELECTION NETWORK.

Number of hidden layers	1	2	3	4	5
Average QoE	6.033	6.217	6.371	6.370	6.365

III. PERFORMANCE EVALUATION

In this section, we delve in-depth into the implementation details of each component of VHAS and describe the hyper-parameter settings of the adaptive bitrate switching algorithm. We then evaluate its performance with comparison to other state-of-the-art solutions. To this end, we have implemented a public live streaming platform, which simulates the data interaction between clients and streaming media servers for the last-mile transmission. It uses Mahimahi's network emulation tools [5] to emulate many different link conditions (different round-trip times and network bandwidth) by taking the real network traces as input.

A. Implementation

Variable-length segmentation. With the increase of K value, the number of the I-frames inserted in the lowest bitrate video stream will increase exponentially. Since the adaptive bitrate switching algorithm needs to optimize the video smoothness level to avoid frequent bitrate switching, many I-frames inserted are invalid and dramatically increase the bandwidth overhead. By experimental comparison (see TABLE II), we have found that when K is equal to 2, the user experience is the best.

Time-slot. Based on trace analysis, we set the sub-decision period as 0.5s in our experiments.

I-frame prediction network. The network structure of the I-frame prediction network contains three fully connected layers (see TABLE III). The number of the hidden cells in the first fully-connected layer is 256, and in other fully-connected layers is 128. Its output includes V (V is equal to the number of actions) neurons. The I-frame prediction network applies a mean square error function to update model parameters and its learning rate is 0.001. Through experimental comparison, we found that $K = 3, N = 100$ can best characterize the dynamic changes of the network bandwidth.

Bitrate selection network. The actor-network contains three fully connected layers (see TABLE IV), and the number of the hidden cells per layer is 128. The outputs from the last layer are then aggregated in a hidden layer that uses V (V is equal to the number of actions) neurons and applies a softmax function. The critic network uses the same NN structure, but its final output is a linear neuron (without activation function). During training, we use a cumulative discount factor $\gamma = 0.99$, which reflects that the current action will be affected by the future time steps. The learning rate of the actor-network is 0.0001, and the learning rate of the critic network is 0.001. Cooperative enterprise builds an outstanding set of evaluation indicators through years of user feedbacks, accurately reflecting the users' preferences in various live streaming scenarios [23]. Based on those indicators and existing QoE models, including DeepQoE [24], DeepCast [18], MPC [14], Pensieve [5], and Vabis [23], the six optimal hyperparameters of reward can be configured as follows: $\alpha = 0.8$, $\alpha_1 = 1/frame\_rate$, $\alpha_2 = 0.01$, $\alpha_3 = 0.2$, $\alpha_4 = 0.8$, $\beta = 0.2$. Though fine-tuning would be useful, we have found that the bitrate selection network performs well for a wide range of hyperparameter settings and can converge stably.

B. Datasets and Baselines

Datasets. The video dataset used in our experiment is a competition set[5], which contains video traces of three application scenarios: game live streaming, indoor live streaming, and sports live streaming. Each trace includes four bitrates streams of {500, 850, 1200, 1850} Kbps, each with a length of 3358 seconds. We have implemented variable-length segmentation by artificially inserting I-frames.

We used three real-world datasets for background network traffic: an FCC broadband dataset[6] provided by the US Federal Communications Commission, a 3G/HSDPA mobile dataset collected from Norway [25], and a 4G/LTE mobile dataset collected from Belgium [6]. The FCC dataset contains over 1 million traces of throughput measurements, with each trace logging the average throughput over 2100 seconds, at 5-second granularity. We generated 1000 traces of 500 seconds each for our corpus by concatenating selected traces from the "Web browsing" category in February 2019 collection. The 3G/HSDPA dataset consists of 30 min traces of throughput measurements with 1-second granularity. The 4G/LTE dataset consists of 15 min traces with 1-second granularity. We randomly selected 1000 throughput traces from the 3G/HSDPA dataset and the 4G/LTE dataset for our corpus, each with a duration of 500 seconds. To avoid trivial cases for which choosing the maximum bitrate is always the optimal solution or where the network cannot support any available bitrate, we only considered original traces whose average throughput was less than 3 Mbps and whose minimum throughput exceeded 0.2 Mbps. Unless otherwise noted, we used a random sample of 80% our corpus as a training corpus and the remaining 20% as a test corpus. We used cross-validation to verify the performance of VHAS.

Baseline ABR algorithms. To understand the effectiveness of our algorithm, we considered four baseline ABR algorithms, which collectively represent the state-of-the-art solutions.

- Buffer-Based (BB) [8], which uses a reservoir of 5 seconds and a cushion of 10 seconds. It attempts to select bitrates by linear regulation such that buffer occupancy always maintains above 5 seconds and select the highest available bitrate if the buffer occupancy exceeds 15 seconds. In VHAS, we used a reservoir of 0.5 seconds and a cushion of 3 seconds;
- Festive [10], which selects the highest available bitrate that is below the predicted network throughput. This is calculated by using the harmonic mean of the network throughput values obtained during the past 5 decision periods;
- Pensieve [5], which trains a neural network by optimizing the QoE metrics and implements bitrate selection for future segments according to environmental state information. In VHAS, we retrain it by using the same configuration as that for Pensieve'
- Vabis [23], which uses an RL-based ABR algorithm to select the bitrate for future frames by observable state information.

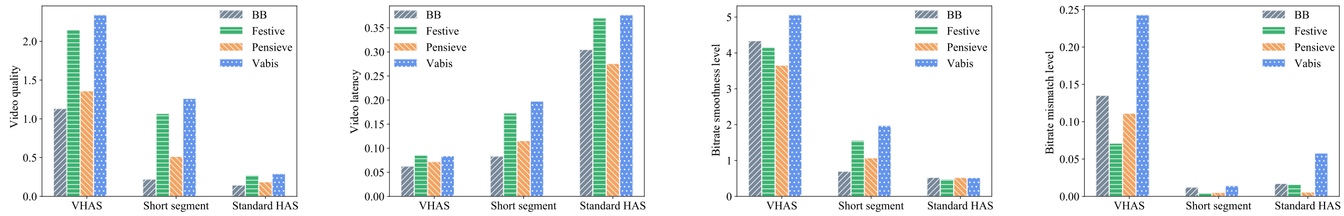
Our experiments cover a broad set of realistic network conditions and multiple video scenarios. We try to answer the following questions through the experiments:

1) How does VHAS compare to the standard HAS and the short segment HAS in terms of bitrate switching timeliness and bandwidth overhead by using the existing ABR algorithms? Fig. 5 and Fig. 6 suggest that VHAS is obviously better than the short segment HAS, with a decrease in the average bandwidth overhead of 37% - 57%. It also improves the timeliness of bitrate switching over the standard HAS. Given that existing ABR algorithms are hard to reduce the bitrate mismatch ratio, the bitrate mismatch penalty is much worse in VHAS.
2) Can the adaptive bitrate switching algorithm reduce the bitrate mismatch ratio and improve the overall QoE? Fig. 7 and Fig. 8 confirm that our algorithm can reduce the bitrate mismatch ratio, and the overall QoE will always be higher than that with the existing ABR algorithms.
3) Under different network conditions and video scenarios, how does the performance of our algorithm compare to that of existing ABR algorithms? As shown in Fig. 9 and Fig. 10, our algorithm performs significantly better than the existing algorithms with a noticeable QoE improvement of 15% - 49%.

C. VHAS with Existing ABR Algorithms

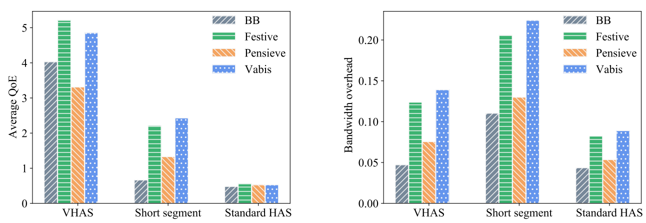
In this subsection, to verify that VHAS has a strong preponderance in terms of timely bitrate switching and efficient bandwidth usage, we compared it with the standard HAS[7] and the short segment HAS [7] on each QoE metric listed in

[5]https://www.aitrans.online/.
[6]https://www.fcc.gov/reports-research/reports/.
[7]https://github.com/Dash-Industry-Forum/dash.js/.



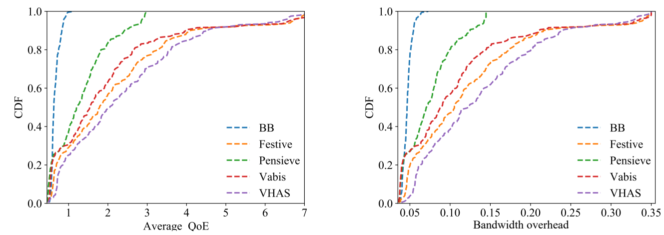
(a) Video quality (b) Video latency (c) Bitrate smoothness level (d) Bitrate mismatch level

Fig. 5. Compare VHAS, the short segment HAS, and the standard HAS by using the existing ABR algorithms in terms of the video quality, video latency, bitrate smoothness level and bitrate mismatch level.



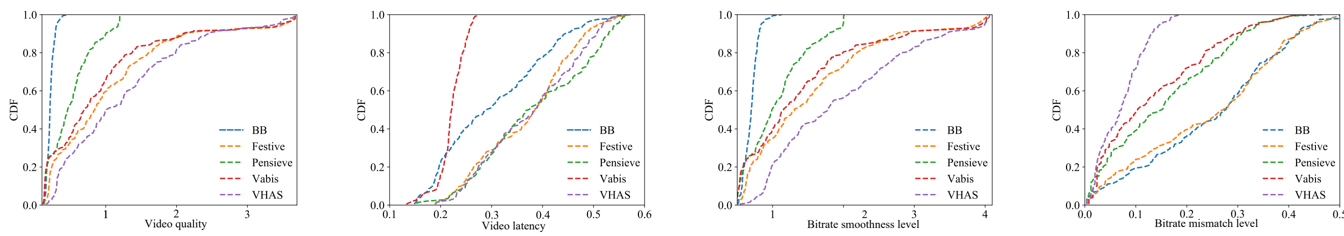
(a) Average QoE (b) Bandwidth overhead

Fig. 6. Compare VHAS, the short segment HAS, and the standard HAS by using the existing ABR algorithms in terms of the average QoE and bandwidth overhead.



(a) Average QoE (b) Bandwidth overhead

Fig. 7. Compare our algorithm and existing outstanding ABR algorithms in VHAS in terms of the average QoE and bandwidth overhead.



(a) Video quality (b) Video latency (c) Bitrate smoothness level (d) Bitrate mismatch level

Fig. 8. Compare our algorithm and existing outstanding ABR algorithms in VHAS in terms of the video quality, video latency, bitrate smoothness level and bitrate mismatch level.

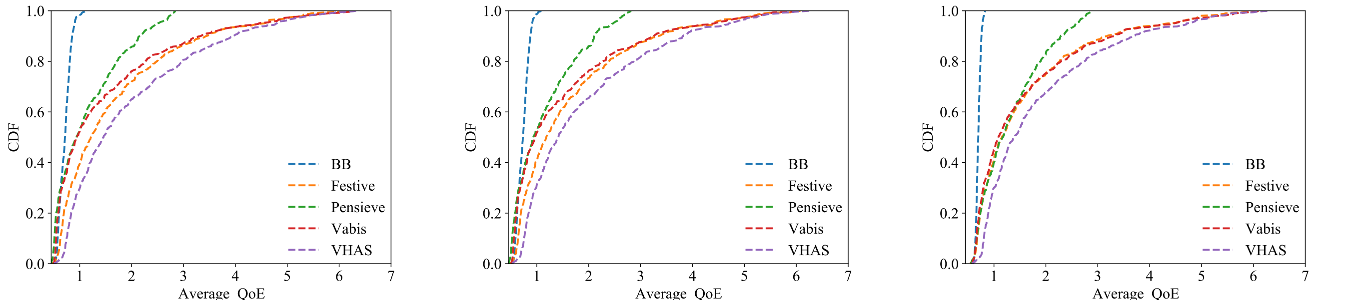
Subsection II-B. The segment duration in the standard HAS and the short segment HAS is 2s and 0.25s, respectively. We used the existing state-of-the-art ABR algorithms listed in Subsection III-B to make the bitrate decision and tested in the game live streaming dataset. The Pensieve and Vabis algorithms were trained in each experiment to optimize the considered metrics using the entire training corpus. Experimental results show six evaluation metrics obtained for the test corpus.

Fig. 5 and Fig. 6 show that VHAS is superior to baselines in many dimensions except for the bitrate mismatch level. Fig. 5(b) shows that the VHAS's video latency is lower than others. Fig. 5(c) demonstrates that VHAS can improve the timeliness of bitrate switching over the standard HAS. VHAS uses the trigger mechanism to transmit variable-length segments, which can significantly increase the times of bitrate switching. Fig. 5(d) shows that the VHAS's bitrate mismatch level is the worst, which is because the existing state-of-the-art ABR algorithms are hard to reduce the bitrate mismatch ratio with variable-length segments. Fig. 6(a) shows that the average QoE based on VHAS is the highest. Fig. 6(b) shows that VHAS outperforms the short segment HAS, with a decrease in the average bandwidth overhead of 37% -57%, which is because VHAS can significantly reduce the ineffective bandwidth overhead by inserting fewer I-frames in the higher bitrate video streams.

D. VHAS with the Adaptive Bitrate Switching Algorithm

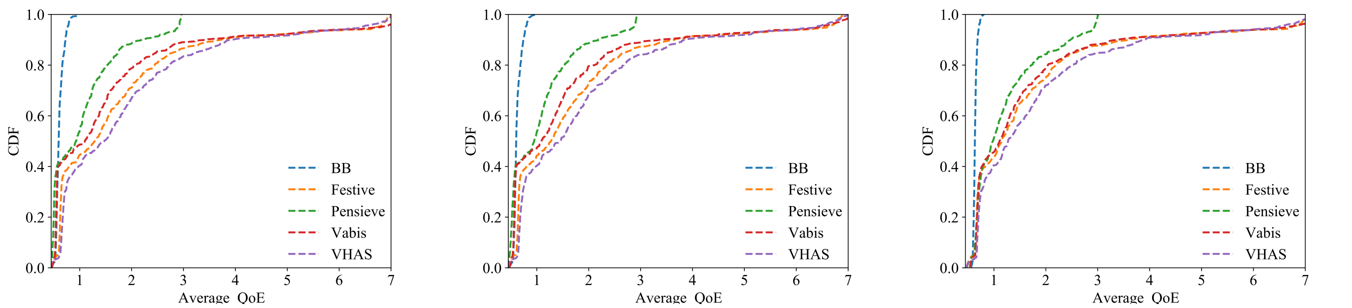
In this subsection, to verify that the adaptive bitrate switching algorithm can reduce the bitrate mismatch ratio and improve the overall QoE, we compared it with the existing outstanding ABR algorithms in terms of each QoE metric listed in Subsection II-B. Those outstanding ABR algorithms are listed in Subsection III-B. For comparison, we trained all ABR algorithms in VHAS. In each experience, we continued to iterate the Pensieve and Vabis algorithms until we obtained

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3165381, IEEE Transactions on Multimedia



(a) Game live streaming scenarios (b) Room live streaming scenarios (c) Sports live streaming scenarios

Fig. 9. Compare our algorithm and existing outstanding ABR algorithms in different video scenarios in terms of the average QoE. The results were collected on the 3G/HSDPA and 4G/LTE hybrid network datasets.



(a) Game live streaming scenarios (b) Room live streaming scenarios (c) Sports live streaming scenarios

Fig. 10. Compare our algorithm and existing outstanding ABR algorithms in different video scenarios in terms of the average QoE. The results were collected on the FCC broadband network datasets.

the optimal model under the QoE metrics considered. For the BB and Festive algorithms, we adjusted the parameters of each algorithm to obtain the optimal QoE. Experimental results for the game live streaming dataset and the hybrid network dataset.

Fig. 7 and Fig. 8 show that VHAS outperforms other ABR algorithms in all the dimensions considered. Fig. 7(a) shows that the average QoE of our algorithm exceeds that of other ABR algorithms. The reason is that the bitrate select network predicts the network bandwidth more accurately and considers the transmission latency more comprehensively. From Fig. 8(d), we determine that the bitrate mismatch level of our algorithm is far superior to that of other excellent ABR algorithms. The I-frame prediction network can significantly reduce the probability of the invalid bitrate decisions (the selected bitrate video stream does not have an I-frame in the next sub-decision period) and improve the hit ratio of the bitrate decisions.

E. Generalization

In this subsection, to verify that our algorithm has strong generalizability, we compared it with the existing outstanding ABR algorithms in the different video and network scenarios in terms of the average QoE. Those outstanding ABR algorithms are listed in Subsection III-B. In each experience, we trained all ABR algorithms in VHAS, and we iterated or adjusted their parameters to obtain the optimal QoE. We compared three video scenarios with the highest viewing frequencies, including room live streaming, game live streaming, and sports live streaming. We compared two network scenarios, namely the 3G/4G wireless network and broadband network scenarios. The experimental results are provided in the form of full CDFs for all combinations.

As shown in Fig. 9 and Fig. 10, our algorithm can maintain higher performance for the different video and network scenarios considered. Especially for the 3G/4G wireless network scenarios, it performs significantly better than those existing ABR algorithms with improvements in average QoE of 15% - 49%. Compared with the BB and Festive algorithms, the adaptive bitrate switching algorithm uses reinforcement learning to train a neural network and automatically adjust its parameters according to its input to obtain the optimal bitrate in complicated scenarios. Compared with the Pensieve algorithm, our algorithm has more accurate I-frame prediction and more comprehensive state information.

IV. Related Work

The hyper text transfer protocol (HTTP) is generally firewall-friendly, and HTTP server resources are also widely available; hence, supporting HTTP-based adaptive streaming (HAS) for a massive audience can be cost-effective using the existing web infrastructure [26]. In HAS, a video stream is divided into multiple segments, including initialization and media segments [2]–[4]. The initialization segments contain the required information for initializing a media decoder. The media segments contain media data and stream access points, indicating where the decoder can play. Each segment is encoded at several discrete bitrates, where a higher bitrate represents a higher video quality and a larger segment size, and the segment boundaries of different bitrates are aligned [5]. Using a series of HTTP's GET commands, a user can progressively download a segment with a specific bitrate while playing those already being downloaded. Any damaged or delayed segment will have a limited impact, thus ensuring continuous playback [27]. HAS is available on Apple's Safari, Google's Chrome, and Microsoft's Edge, and has been used by such primary streaming services as YouTube and Netflix. As compatible clients become available, it promises to be widely adopted in a wide range of devices and platforms [28], [29].

Adaptive bitrate (ABR) algorithms determine how a client selects different bitrates of segments to achieve the best QoE in HAS, e.g., request a lower video bitrate when network bandwidth is low or a higher video bitrate if enough network bandwidth is available. Generally, existing ABR algorithms can be organized into three broad categories, namely, buffer-based ABR algorithms, rate-based ABR algorithms, and quality-based ABR algorithms.

The most commonly used buffer-based ABR algorithms are the BBA [8] and BOLA [9] algorithm. The video bitrate requested by a player is determined only according to the current playback buffer occupancy, which aims to maintain the playback buffer occupancy within a suitable range that balances the video rebuffering (the playback buffer is empty) and video quality. For example, the BBA algorithm attempts to select bitrates by linear regulation such that the playback buffer occupancy always maintains above 5 seconds, especially selecting the highest available bitrate if the playback buffer occupancy exceeds 15 seconds. Since the playback buffer occupancy is very susceptible to the network bandwidth and the bitrate decision based on the current buffer has strong hysteresis, rate-based ABR algorithms are proposed. Rate-based ABR algorithms such as FESTIVE [10], CS2P [11] and pandas [12] have become popular in last years. By estimating available network bandwidth at the next decision period according to the historical records of network bandwidth, the highest available video bitrate is selected, which is below the predicted network bandwidth. Notable amongst the core of the rate-based ABR algorithms is the prediction of the future network bandwidth. For example, the Festive algorithm uses the harmonic mean of the network bandwidth records obtained during the past five decision periods to predict the network bandwidth of the next period. The combination of buffer-based and rate-based ABR algorithms is referred to as quality-based ABR algorithms, which are found to give good results in network bandwidth prediction and QoE optimization. These algorithms use the historical records of the network bandwidth and the current playback buffer occupancy to jointly decide the video bitrate, including MPC [14], ABMA+ [15], and BOLA-E [30].

Those algorithms are conventionally based on pre-defined rules with parameters observed from the server or client sides. Such rules are deterministic and tailored to specific network configurations. Recently, advanced learning tools [31]–[34], in particular, reinforcement learning (RL), have also been used to enhance the ABR algorithms. Those algorithms train a neural network according to the state information from the server, client, and transmission network, which can automatically adjust their parameters according to their input to obtain optimal video bitrates in complicated scenarios, including pensieve [5], QARC [16], Vabis [23], HotDASH [35] and Comyco [36]. Therefore, we also construct an RL-based ABR algorithm in this paper.

V. Conclusion

In this paper, we proposed VHAS, an extension for intelligent variable-length segmentation. Unlike the standard HAS, we divided a video stream into variable-length segments and implemented a data transmission process through a trigger mechanism, which could accommodate timely and accurate bitrate switching with the minimized bandwidth overhead. Unlike existing ABR algorithms, the adaptive bitrate switching algorithm with data-driven I-frame prediction could accurately predict the probability of I-frames appearing in the next subdecision period, improving the hit ratio of bitrate switching and minimizing the bitrate mismatch ratio. Over a broad set of network conditions and video scenarios, we found that VHAS outperformed existing algorithms with improvements in average QoE of 15% - 49% and a decrease in the average bandwidth overhead of 37% - 57%. In future work, we hope that VHAS can be deployed in large-scale network topology.

References

[1] Z. Lu, H. Xia, S. Heo, and D. Wigdor, "You Watch, You Give, and You Engage: A Study of Live Streaming Practices in China," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–13.

[2] Z. Jiang, C. Xu, J. Guan, Y. Liu, and G. Muntean, "Stochastic Analysis of DASH-Based Video Service in High-Speed Railway Networks," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1577–1592, 2019.

[3] K. Xiao, S. Mao, and J. K. Tugnait, "Robust QoE-Driven DASH Over OFDMA Networks," *IEEE Transactions on Multimedia*, vol. 22, no. 2, pp. 474–486, 2020.

[4] J. Song, F. Yang, W. Zhang, W. Zou, Y. Fan, and P. Di, "A Fast FoV-Switching DASH System Based on Tiling Mechanism for Practical Omnidirectional Video Services," *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2366–2381, 2020.

[5] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in *SIGCOMM*, 2017, pp. 197–210.

[6] J. Van Der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck, "HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.

[7] T. Zhang, F. Ren, W. Cheng, X. Luo, R. Shu, and X. Liu, "Towards Influence of Chunk Size Variation on Video Streaming in Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1715–1730, 2020.

[8] T. 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 *SIGCOMM*, 2014, pp. 187–198.

[9] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.

[10] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, 2014.

[11] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction," in *SIGCOMM*, 2016, pp. 272–285.

[12] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, 2014.

[13] J. V. D. Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. D. Turck, "HTTP/2-Based Adaptive Streaming of HEVC Video over 4G/LTE Networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.

[14] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *SIGCOMM*, 2015, pp. 325–338.

[15] A. Beben, P. Wiśniewski, J. M. Batalla, and P. Krawiec, "ABMA+: Lightweight and Efficient Algorithm for HTTP Adaptive Streaming," in *ACM MMSys*, 2016, pp. 2:1–2:11.

[16] T. Huang, R. Zhang, C. Zhou, and L. Sun, "QARC: Video Quality Aware Rate Control for Real-Time Video Streaming Based on Deep Reinforcement Learning," in *ACM MM*, 2018, pp. 1208–1216.

[17] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[18] F. Wang, C. Zhang, F. Wang, J. Liu, Y. Zhu, H. Pang, and L. Sun, "Intelligent Edge-Assisted Crowdcast with Deep Reinforcement Learning for Personalized QoE," in *INFOCOM*, 2019, pp. 910–918.

[19] Y. Wu, E. Mansimov, R. Grosse, S. Liao, and J. Ba, "Scalable Trust-Region Method for Deep Reinforcement Learning using Kronecker-Factored Approximation," in *NIPS*, 2017, pp. 5285–5294.

[20] V. Mnih *et al.*, "Asynchronous Methods for Deep Reinforcement Learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[22] M. Volodymyr *et al.*, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[23] T. Feng, H. Sun, Q. Qi, J. Wang, and J. Liao, "Vabis: Video Adaptation Bitrate System for Time-Critical Live Streaming," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2963–2976, 2020.

[24] M. Shen, J. Zhang, K. Xu, L. Zhu, J. Liu, and X. Du, "DeepQoE: Real-time Measurement of Video QoE from Encrypted Traffic with Deep Learning," in *IWQoS*, 2020, pp. 1–10.

[25] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," in *ACM MMSys*, 2013, pp. 114–118.

[26] Z.-N. Li, M. S. Drew, and J. Liu, *Fundamentals of multimedia*. Springer, 2004.

[27] C. Fan, S. Yen, C. Huang, and C. Hsu, "Optimizing Fixation Prediction Using Recurrent Neural Networks for 360 Video Streaming in Head-Mounted Virtual Reality," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 744–759, 2020.

[28] C. Zhan, H. Hu, Z. Wang, R. Fan, and D. Niyato, "Unmanned Aircraft System Aided Adaptive Video Streaming: A Joint Optimization Approach," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 795–807, 2020.

[29] X. Xiao, W. Wang, T. Chen, Y. Cao, T. Jiang, and Q. Zhang, "Sensor-Augmented Neural Adaptive Bitrate Video Streaming on UAVs," *IEEE Transactions on Multimedia*, vol. 22, no. 6, pp. 1567–1576, 2020.

[30] K. Spiteri, R. Sitaraman, and D. Sparacio, "From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player," in *ACM MMSys*, 2018, pp. 123–137.

[31] S. Kim and C. Kim, "XMAS: An Efficient Mobile Adaptive Streaming Scheme Based on Traffic Shaping," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 442–456, 2019.

[32] H. Chen, X. Zhang, Y. Xu, Z. Ma, and W. Zhang, "Efficient Mobile Video Streaming via Context-Aware RaptorQ-Based Unequal Error Protection," *IEEE Transactions on Multimedia*, vol. 22, no. 2, pp. 459–473, 2020.

[33] J. Guo, X. Gong, J. Liang, W. Wang, and X. Que, "EAAT: Environment-Aware Adaptive Transmission for Split-Screen Video Streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4355–4367, 2020.

[34] Z. Xu, X. Zhang, and Z. Guo, "QoE-Driven Adaptive K-Push for HTTP/2 Live Streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1781–1794, 2019.

[35] S. Sengupta, N. Ganguly, S. Chakraborty, and P. De, "HotDASH: Hotspot Aware Adaptive Video Streaming Using Deep Reinforcement Learning," in *ICNP*, 2018, pp. 165–175.

[36] T. Huang, C. Zhou, X. Yao, R. Zhang, C. Wu, B. Yu, and L. Sun, "Quality-Aware Neural Adaptive Video Streaming With Lifelong Imitation Learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2324–2342, 2020.

Tongtong Feng received his B.E. degree in communication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2017. He is currently working towards his Ph.D. degree in computer science and technology with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He has published 5 papers in international journals such as IEEE Transactions on Multimedia and presented at prestigious conferences, including ACM MM. His current research interests include video transmission optimization, adaptive multimedia streaming, and traffic engineering.

Qi Qi received her Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010. She is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She has authored or coauthored more than 30 papers in international journals and was the recipient of two awards from the National Natural Science Foundation of China. Her research interests include edge computing, cloud computing, Internet of Things, ubiquitous services, deep learning, and deep reinforcement learning.

Jingyu Wang received his Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He is senior member of IEEE/CIC and selected for Beijing Young Talents Program. He has published more than 100 papers in such as the JSAC, TMC, CVPR, ACL, MM, ICDE, AAAI and so on. His research interests include broad aspects of Intelligent Networks, Edge/Cloud Computing, Machine Learning, AIOps and Self-Driving Network, IoV/IoT, Knowledge-Defined Network and Intent-Driven Networking.

Jianxin Liao received his Ph.D. degree from the University of Electronics Science and Technology of China, Chengdu, China, in 1996. He is currently the Dean of the Network Intelligence Research Center and a Full Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He has authored or coauthored hundreds of research papers and several books. He has won a number of prizes in China for his research achievements, including the Premier Award for Distinguished Young Scientists from the National Natural Science Foundation of China in 2005 and a special invitation by the Ministry of Education to be Professor of the "Yangtse River Scholar Award Program" in 2009. His main research interests include cloud computing, mobile intelligent networks, intelligent network services, networking architectures and protocols, and multimedia communication.

Jiangchuan Liu (S'01-M'03-SM'08-F'17) is a University Professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada. He is a Fellow of the Canadian Academy of Engineering, an IEEE Fellow, and an NSERC E.W.R. Steacie Memorial Fellow. He was an EMC-Endowed Visiting Chair Professor at Tsinghua University (2013-2016). In the past he has served as an Assistant Professor at the Chinese University of Hong Kong and as a research fellow at Microsoft Research Asia. He received his BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and his PhD degree from the Hong Kong University of Science and Technology in 2003, both in computer science. He is a co-recipient of the inaugural Test of Time Paper Award of IEEE INFOCOM (2015), ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), and ACM Multimedia Best Paper Award (2012). His research interests include multimedia systems and networks, cloud and edge computing, social networking, online gaming, and Internet of things/RFID/backscatter. He has served on the editorial boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Big Data, IEEE Transactions on Multimedia, IEEE Communications Surveys and Tutorials, and IEEE Internet of Things Journal. He is a Steering Committee member for IEEE Transactions on Mobile Computing and Steering Committee Chair of IEEE/ACM IWQoS (2015-2017). He is TPC Co-Chair of IEEE INFOCOM'2021.