IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 10, 2019, accepted September 24, 2019, date of publication September 30, 2019,
date of current version October 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944437

Network-Assisted Neural Adaptive Naked-Eye 3D Video Streaming Over Wireless Networks

YITONG LIU, (Member, IEEE), WEI HE, YIDI WANG, AND HONGWEN YANG, (Member, IEEE)
School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
Corresponding author: Yitong Liu (liuyitong@bupt.edu.cn)

ABSTRACT High quality transmission of Virtual Reality (VR) video depends strictly on bandwidth and delay requirements. It becomes possible with the maturity and commercialization of 5G technology. However, to transmit VR video streaming over wireless communication system, we must cope with the challenge that fluctuation exists in the wireless channel conditions. For example, available channel bandwidth, packet loss rate, and interference level may vary with time due to channel fading and user's mobility. The problem is more prominent when transmitting naked-eye 3D video which generally consists of multiple viewpoints with different resolutions. In order to optimize the quality of experience (QoE) of watching naked-eye VR video over wireless networks, this paper proposes a network-assisted neural adaptive video streaming algorithm (NAVSA). Specifically, we present a modified QoE function to describe the quality of naked-eye 3D streaming quantitatively, which not only considers the quality of naked-eye 3D video itself, but also considers the phenomenon of rebuffering and video fluctuation that occurs during video transmission. Next, with the network-assisted feedback, the physical layer information, the buffer occupancy of the video client, and the size of the next video chunk are collected to train a reinforcement learning model. Based on dynamic adaptive streaming over HTTP (DASH), the model can automatically choose appropriate viewpoints and resolutions corresponding to the current condition of the wireless networks such that the network capacity can be fully explored. To verify the performance of our proposed NAVSA, we simulate 3 naked-eye 3D video application scenarios on the NS3 platform. The results show that the performance of NAVSA is about 5∼8% better than some state-of-the-art algorithms in wireless networks.

INDEX TERMS Reinforcement learning, adaptive transmission, naked-eye 3D, network assistance, wireless networks.

I. INTRODUCTION

With the development of mobile terminals and the enrichment of online video content, the demand for streaming video service applications is increasing, and it is becoming the main type of traffic in wireless networks. Currently, streaming video services account for a considerable proportion of Internet traffic [1]. According to the Cisco report [2], it is predicted that the network video traffic will account for 82% of all consumer Internet traffic by 2021. In the past decade, mobile users' demand for high-definition images and high-quality video streams has been continuously upgrading.

With the enhancement of 5G technologies, such as network slice [3], D2D [4], [5], physical layer coding [6], [7] and other IoT technologies [8], [9], many streaming media application services emerge [10]–[13]. Among them, VR applications become the new hits due to the unique immersive experience they brought to the public [14]–[16]. The rapid development of wearable devices makes VR video streaming service applications quickly enter people's sights and lives, and is widely used in video live broadcast, immersive games, online exhibition halls, panoramic street scenes and other fields. In particular, the naked-eye 3D video enables each user to view objects from a freely switchable viewpoint, and the viewer can place himself in an immersive scene without any additional accessories, making it ideal for commercial use [17]–[19].

However, with the widespread application of streaming video services and the explosive development of content, the tradeoff between business needs and wireless network transmission capabilities has become increasingly prominent, such as greedier bandwidth and stricter delay.

The associate editor coordinating the review of this manuscript and approving it for publication was Dapeng Wu.

The existing wireless networks shouldn't transmit these applications directly, otherwise it will sacrifice the picture quality, or there will incur serious stalls and scene switching delay, causing the viewer's dizziness. However, vivid image, immersive sound and rich video information acquire sufficient bandwidth. To solve these problems, there has been mickle works focusing on the encoding/decoding schemes to reduce video traffic [20]–[27]. For examples, to reduce the high bandwidth consumption in cloud gaming, [20] proposed a hybrid streaming framework which jointly applies video streaming and graphics streaming to provide a high-quality gaming experience. Reference [21] proposed the adaptive multi-view video streaming system for real-time streaming which can achieve low encoding/decoding computation and high video quality while satisfying bandwidth limitations. To reduce the huge amount of video content redundancy, a content-slimming system framework based on the convergence of computing, communication and cache was proposed in [22]. However, the existing methods can not effectively adapt to the wireless channel fluctuations. In a wireless network, the bandwidth, channel quality, packet loss, transmission delay and many other metrics typically vary from time to time, ignoring these variances may lead to lower efficiency or degraded QoE.

The current immersive video transmission research on the wireless networks is mainly based on the MPEG-DASH technology [28]–[32], in which the quality of the video chunk to be downloaded is determined by observing the throughput of the network and the length of the client buffer. These adaptive bitrate (ABR) algorithms run on the client video player, dynamically selecting the video bitrate for each video chunk to maximize the user's QoE. Most of the existing ABR algorithms are based on a fixed rule, or consider the estimated network throughput [33], [34], or consider the player buffer size [35], or comprehensively take these two quantities [36] into account. These scenarios need to be adjusted for different network conditions and QoE requirements. Reference [33] developed a suite of techniques that systematically guide the trade-offs between stability, fairness, and efficiency, and it proposed a general framework for robust video adaptation, which was found to improve fairness by 40%, stability by 50% and efficiency by at least 10%. The most advanced model predictive control (MPC) algorithm [37] is aimed to solve the QoE maximization problem in the future of several video blocks, which is particularly sensitive to the prediction of future network throughput. In order to provide users with better QoE through wireless network transmission, [38] proposed a framework named as QoE-driven and network-assisted naked-eye 3D adaptive video streaming over wireless networks (QNANA) to adaptively transmit naked-eye 3D video streaming over wireless networks through network-assisted methods.

Recently, machine learning based methods have been widely used in video streaming. Due to the limitation of traditional ABR algorithms that their rules are deterministic and inevitably fail to achieve optimal performance across a broad set of network conditions and QoE metrics. For this problem, [39] proposed a reinforcement learning based ABR algorithm Pensieve, which can adapt to different user preferences directly by modifying the reward module, and learn the best ABR algorithm to adapt to various network conditions. Reference [40] applied the deep reinforcement learning approach to dynamic resource optimization for wireless buffer-aware video streaming under unknown channel state and video rate. These reinforcement learning methods don't know any information at the beginning of training, but they can gradually learn to make a good ABR decision under the incentive of reward signals. After training, the performance of reinforcement learning methods is comparable to or better than the state-of-the-art ABR algorithms. However, the input of these reinforcement learning methods still adopts the bandwidth estimation of the application layer, which is incompatible with the wireless channel condition, resulting in inadequate utilization of bandwidth resources.

In this paper, we modify the QoE function in order to accurately describe the viewing experience of naked-eye 3D video in wireless network transmission and propose a network-assisted adaptive bitrate algorithm based on reinforcement learning, namely NAVSA, to meet the best QoE. In NAVSA, a neural network, which uses collected network physical layer information, buffer occupancy and next video chunk size as inputs replaces the traditional DASH policy. We take the QoE function of naked-eye 3D video streaming as the objective function, and update the neural network parameters under different network traces through reinforcement learning, in which the QoE is continuously improved. By tracking the changes of the physical layer of the wireless channels, NAVSA can better adapt to the network environment than other reinforcement learning adaptive bitrate algorithms such as Pensieve, which uses past chunk throughput estimation, past chunk download time, next chunk size, current buffer size, number of chunks left and last chunk bitrate as inputs. To verify it, we use the same network traces to train a NAVSA model and a Pensieve model, and compare them using the same testing set, which shows that NAVSA is a more adaptive bitrate algorithm than Pensieve. Then, we put the two trained models into 3 specific scenario simulations, and analyze the performance with other algorithms in details. The results show that the outcome of NAVSA is about 5∼8% better than some state-of-the-art algorithms.

The reminder of this paper is organized as follows. Section II presents the details of our proposed NAVSA algorithm, including QoE function, network-assisted function and reinforcement learning model. The model is trained with simulated data based on A3C framework, and evaluated the accuracy compared with the Pensieve. In section III, we verify the NAVSA algorithm with naked-eye 3D video streaming under different scenarios, and compare our proposed algorithm with other adaptive algorithms. The section IV gives the conclusion.

TABLE 1. Viewpoints and resolution combinations.

Bitrate	Resolution	Viewpoints
100Mbps	1920*1080	16
90Mbps	1920*1080	9
80Mbps	1280*720	16
70Mbps	1280*720	9
60Mbps	1280*720	4
50Mbps	960*540	16
40Mbps	960*540	9
30Mbps	960*540	4
20Mbps	480*360	9
10Mbps	480*360	4

II. ALGORITHM

A. QOE FUNCTION

The naked-eye 3D video stream realizes stereoscopic perception based on the principle that the left eye and the right eye simultaneously acquire different image information from the same object. By synthesizing 3D images using multiple viewpoints, the autostereoscopic 3D effect can be well achieved. More views mean greater viewers' stereoscopic freedom. For a given degree of freedom, more viewpoints can also produce more detailed and continuous images, thus creating even more impressive depth and robustness.The quality of naked-eye 3D video is mainly affected by two factors: stereo effect and picture quality. To quantitatively measure the quality of naked-eye 3D video, [41] proposed a multi-view video quality evaluation method based on the structural similarity image measure (SSIM). In order to simplify the model and improve the operability of the simulation, we characterize the stereoscopic effect with the number of viewpoints and characterizes the image quality with resolution. It should be noted that the viewpoints and resolution only describe the quality of the video itself, while not involve the impact of video rebuffering and fluctuation on users' QoE in the transmission process.

As shown in Table.1, it is assumed that there are $K = 10$ kinds of viewpoints and resolution combinations for the video source of the same content, and each combination has a specific bitrate level configuration.

For the k-th combination of viewpoints and resolution in Table.1, the quality of the video $q(k)$ is defined by

$$q(k) = \rho \cdot SQP(k) + (1-\rho) \cdot IQ(k), \tag{1}$$

where $k = 1, 2, \cdots, K$, ρ denotes a weighting factor, $SQP(k)$ and $IQ(k)$ denote, respectively, the stereoscopic influence factor and the picture quality influence factor of the k-th combination. In addition, the QoE is related to the video quality, as well as the rebuffering time and smoothness in the process of watching naked-eye 3D video in wireless transmission, which is shown in the following equations (2) and (3):

$$qoe_{rebuff} = -\sum_{n=1}^{N} B_n, \tag{2}$$

$$qoe_{smooth} = -\sum_{n=1}^{N-1} |rep(k_{n+1}) - rep(k_n)|, \tag{3}$$

where B_n represents the rebuffering time in the process of downloading the n-th video chunk, and $rep(k_n)$ represents the bitrate level corresponding to viewpoints and resolution combination k_n.

With qoe_{rebuff} and qoe_{smooth} defined in (2) and (3), the QoE metric of the naked-eye 3D video is defined as equation (4).

$$QoE = \alpha \sum_{n=1}^{N-1} q(k_n) + \beta \cdot qoe_{rebuff} + \gamma \cdot qoe_{smooth}, \tag{4}$$

where $q(k_n)$ represents the video quality of the n-th video chunk corresponding to viewpoints and resolution combinations $k_n \in \{1, 2, \cdots, K\}$. α is the influence factor of the naked-eye 3D video chunk quality, β is the influence factor of the rebuffering time, and γ is the influence factor of smoothness. The magnitude of these three weighting coefficients determines whether the user prefers high quality video or prefers low rebuffering time under the QoE metric. In this paper, QoE_{hd} metric is selected to assume that users prefer high quality video.

B. NETWORK-ASSISTED FUNCTION

In the traditional DASH policy, the client calculates the available bandwidth at the next video chunk by using the terminal network port throughput or the rate of downloading chunks within a certain period of time. This method has no data interaction with the underlying network, leading to hysteresis, which causes the bandwidth carrying capacity to be often overestimated or underestimated. The network assisted policy can directly obtain the bandwidth estimation value from the network side and transmit the bandwidth estimation value to the client by establishing the information interaction interface between the network side and the application side, thereby improving the accuracy and stability of the bandwidth measurement, and better tracking the channel condition changes.

In order to follow the channel fluctuation, an algorithm is proposed in [38] which samples the physical layer information at each time intervals, taking the time $t - \Delta t$ as the starting center sampling point, Δt as the step sampling interval, and $\Delta t/2$ as the symmetric sampling range. The discrete layer symmetric sampling is performed on the physical layer information for the time ranging from $t - \Delta t$ to $t - m\Delta t$, and the estimated available bandwidth carrying capacity at the sampling time t is calculated as

$$BW_t(i) = \frac{1}{\Delta t} \sum_{j=t-\frac{(1+2i)\Delta t}{2}}^{t+\frac{(1-2i)\Delta t}{2}} TBSize(j), \tag{5}$$

$$\overline{BW_t} = \frac{1}{m}\sum_{i=1}^{m} BW_t(i), \tag{6}$$

where $BW_t(i)$ represents the physical layer throughput sample value of the i-th equal time interval at the sampling time t, $\overline{BW_t}$ represents the arithmetic mean of the throughput sampling time series at the sampling time t, which is used to estimate the available bandwidth carrying capacity at that time. $TBSize(j)$ represents the transport block size received by the receiving end at time j. In the simulation experiment, Δt is set to 25 milliseconds, and the total number of samples is set to 20, that is, the channel condition in the time range of about 0.5 seconds is covered.

C. REINFORCEMENT LEARNING MODEL

In this section, we introduce the design ideas and training methods of the proposed network-assisted neural adaptive video streaming algorithm (NAVSA), and detail the architecture and functions of each module in the algorithm.

1) DESIGN

We use the A3C (asynchronous advantage actor-critic) framework [43] to train the NAVSA, as shown in Fig. 1. This is a relatively advanced reinforcement learning algorithm that introduces a multi-threaded asynchronous architecture based on Actor-Critic. That is, by creating multiple threads, each thread has an agent running on the copy of the environment, and the parameter gradients of the multiple threads are added to the neural network of the global agent after a certain number of training steps. The neural networks of all threads then update the parameters of the neural network sharing the global agent together to achieve parallel and asynchronous execution and learning in multiple environment instances. Each thread has two neural networks, one called *actor network*, which is responsible for the policy gradient to learn strategy; the other is called *critic network*, which is responsible for the value evaluation function. The structure of these two neural networks is shown in Fig. 2.

When the algorithm has been completely trained, only the parameters of the global neural network need to be saved as a policy model. Then, enter the state into the model to get the probability of each action, and then select the action with the highest probability as the model output.

In the following, the various modules in NAVSA are described.

- **State:** After downloading a video block n, the resulting $s_n = (\overrightarrow{x}_n, \overrightarrow{c}_n, b_t)$ is used as input to the two neural networks. $\overrightarrow{x}_n$ is a vector representing the estimated network capacity of the past 5 seconds. In the process of model trains, we directly replace collected network physical layer information with the throughput value of the past 5 seconds in the network trace document. $\overrightarrow{c}_n$ is also a vector indicating the size of the next video chunk at each viewpoints and resolution combination. b_t indicates the buffer size of the client player at current time t. Specifically, $\overrightarrow{x}_n$ and $\overrightarrow{c}_n$ are given by

$$\overrightarrow{x}_n = \left(BW_{t-MT}, BW_{t-(M-1)T}, \cdots, BW_t\right) \tag{7}$$

$$\overrightarrow{c}_n = (chunk_{n+1}(1), chunk_{n+1}(2), \cdots, chunk_{n+1}(K)) \tag{8}$$

where BW_{t-MT} represents the estimated available bandwidth carrying capacity at time $t-MT$, and M is set to 9, T is set to 500 milliseconds. $chunk_{n+1}(k_{n+1})$ represents the size (in bytes) of the $(n+1)$-th video chunk with viewpoints and resolution combination k_{n+1}.

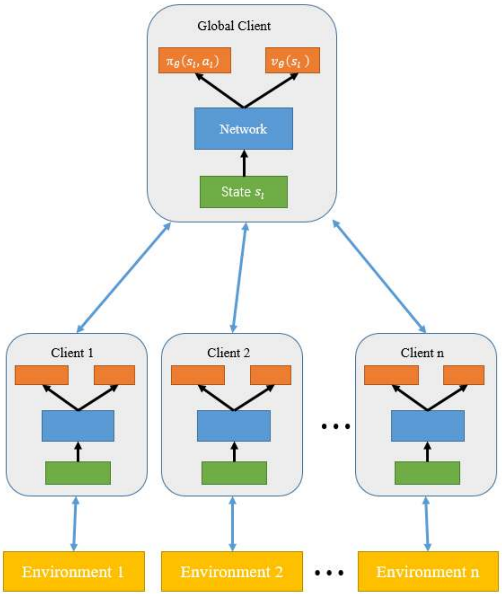


FIGURE 1. A3C framework.

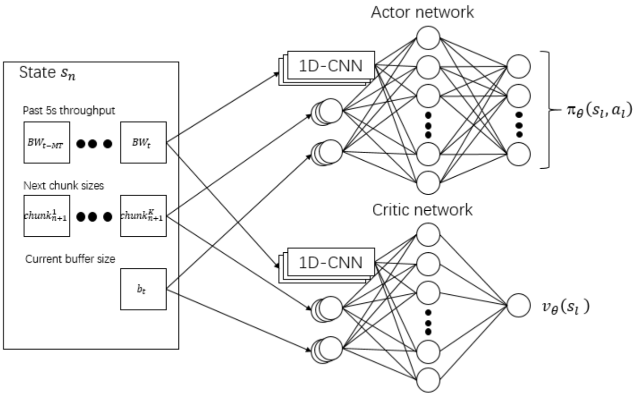


FIGURE 2. The actor network and critic network used in this paper to generate NAVSA.

- **Action Space**: We use the total number of viewpoints and resolution combinations K as the size of the action space A, and $a_n \in \{1, 2, \cdots, K\}$ denotes the combination number of the viewpoints and resolution selected by the client player when downloading the n-th video chunk.

- **Policy**: The policy refers to the mapping rule from state s_n to action a_n. This mapping can be denoted by

$$\pi : \pi(s_n, a_n) \rightarrow [0, 1] \tag{9}$$

which is a probability distribution function, indicating the probability of selecting a_n under state s_n. In the A3C framework, the probability distribution function $\pi(s_n, a_n)$ is specifically implemented by the actor network, whilst the critic network is used to implement the value estimation function $\pi(s_n, a_n)$, which is used to represent the value of the state s_n. The specific structure of these two neural networks can be found in [39].

- **Reward Function**: We use $r(s_n, a_n)$ to indicate the reward returned to the policy after the client downloads the n-th video chunk. $r(s_n, a_n)$ is defined as

$$\begin{aligned} r(s_n, a_n) &= \alpha q(a_n) - \beta B_n - \gamma|rep(a_n) - r(a_{n-1})| \\ r(s_1, a_1) &= \alpha q(a_1) - \beta B_1 \end{aligned} \tag{10}$$

where $q(a_n), B_n, r(a_n), \alpha, \beta, \gamma$ are defined in previous subsections.

With the reward function defined in (10), the QoE of the user watching naked-eye 3D video can also be expressed by:

$$QoE = \sum_{n=1}^{N} r(s_n, a_n). \tag{11}$$

- **Environment**: Because the conditions and devices required to transmit naked-eye 3D video are not available yet, and the actual video downloading and playback is too time-consuming. So we simulate the process of slicing, downloading and playing back the naked-eye 3D video. We re-coded a naked-eye 3D video to generate 10 video sources, each of which has 48 video chuncks and corresponds to a combination of viewpoints and resolution. The duration of each video chunk is set to 1 second. We used the method in [39] to generate a synthetic dataset containing 400 randomly network traces as training set and 120 randomly network traces as testing set. Specifically, these traces are generated using a Markovian model in which each state represented an average throughput in the aforementioned range. The time interval between two adjacent bandwidth values in each trace is 0.5 seconds and the duration of each trace is 200 seconds. Each throughput value is between 5Mbps and 150Mbps and is drawn from a Gaussian distribution centered around the average throughput for the current state, with variance uniformly distributed between 0.05 and 0.5, to simulate channel fluctuations of 5G wireless networks.

2) TRAINING METHODOLOGY

We use θ to represent the parameters of the two neural networks in the policy, then the actor network is represented by $\pi_\theta(s_n, a_n)$, and the critic network is represented by $v_\theta(s_n)$. The purpose of reinforcement learning is to maximize the reward expectation $\mathcal{J}(\theta)$ of the policy model, i.e.

$$\theta^* = \underset{\theta}{argmax}\{\mathcal{J}(\theta)\} \tag{12}$$

where θ^* represents the neural network parameters that we finally hope to learn, $\mathcal{J}(\theta)$ is the objective function of the maximization which is defined as

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[r(\tau)] \tag{13}$$

where

$$\tau \in \{(a_1, a_2, \cdots, a_n)|a_n = 1, 2, \cdots, K\}$$

represents an action selection trajectory. $p_\theta(\tau)$ is a probability distribution function, indicating the probability of the action selection trajectory τ appearing under the policy model with parameter θ. $r(\tau)$ represents the total reward value under this action selection trajectory.

The gradient of the objective function $\mathcal{J}(\theta)$ is given by [42]

$$\begin{aligned} \nabla_\theta &= \int \nabla_\theta p_\theta(\tau) r(\tau) d\tau \\ &= \int p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) r(\tau) d\tau. \end{aligned} \tag{14}$$

Since $p_\theta(\tau)$ can be expressed as

$$p_\theta(\tau) = p(s_1) \prod_{n=1}^{N} \pi_\theta(s_n, a_n) p(s_{n+1}|s_n, a_n), \tag{15}$$

we have

$$\nabla_\theta \log p_\theta(\tau) = \sum_{n=1}^{N} \nabla \log \pi_\theta(s_n, a_n) \tag{16}$$

From this, the gradient $\nabla\mathcal{J}(\theta)$ of the objective function $\mathcal{J}(\theta)$ can be calculated as

$$\nabla\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}\left[\sum_{n=1}^{N} \nabla_\theta \log \pi_\theta(s_n, a_n) r(\tau)\right]. \tag{17}$$

Theoretically, $\pi_\theta(s_{l_2}, a_{l_2})$ does not affect $\pi_\theta(s_{l_1}, a_{l_1})$ if $l_2 > l_1$, so it is concluded that:

$$\nabla\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}\left[\sum_{n=1}^{N} \nabla_\theta \log \pi_\theta(s_n, a_n) \sum_{n'=n}^{N} r(s_{n'}, a_{n'})\right]. \tag{18}$$

The variance of the gradient calculated by above formula can be very large. To address this problem, [43] proposed the advantage function $A(s_n, a_n)$ defined as

$$A(s_n, a_n) = r(s_n, a_n) + \gamma\, v(s_{n+1}) - v(s_n) \tag{19}$$

where $v(s_n)$ is the value estimate of state s_n, which represents the expected value of the discount reward of the current state s_n and all subsequent states during the state transition, implemented by the critic network. γ represents the discount weight.

Substituting the advantage function $A(s_n, a_n)$ for $\sum_{n'=n}^{N} r(s_{n'}, a_{n'})$ in (18), the calculation formula of the objective function gradient becomes:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\sum_{n=1}^{N} \nabla_\theta \log \pi_\theta(s_n, a_n) A(s_n, a_n)\right] \tag{20}$$

The actual training process is mainly divided into two steps, which are performed alternately until the training is completed:

- **Pull**: Synchronize the parameters θ, θ_v of the two global networks to the actor network and the critic network of each thread:

$$\begin{aligned} \theta' &\leftarrow \theta \\ \theta'_v &\leftarrow \theta_v \end{aligned} \tag{21}$$

  After the parameter synchronization completed, perform the step *Push*.
- **Push**: Each thread runs an agent on the copy of the environment, simulating the download process and playback process of the naked-eye 3D video in parallel, and sampling multiple state transitions (s_i, a_i, r_i, s_{i+1}) as a batch under the policy $\pi_{\theta'}$ with the parameter θ'. Then update the global critic network using the timing difference algorithm [42]:

$$\theta_v \leftarrow \theta_v - \alpha_{critic} \sum_i \nabla_{\theta_v} A(s_i, a_i) \tag{22}$$

  where α_{critic} represent the learning rate of global critic network.

  Calculate the advantage function $A(s_n, a_n)$ and update the global actor network:

$$A(s_i, a_i) = r_i + \gamma\, v_{\theta_v}(s_{i+1}) - v_{\theta_v}(s_i) \tag{23}$$

$$\theta \leftarrow \theta + \alpha_{actor} \sum_i \nabla_\theta \log \pi_\theta(s_i, a_i) A(s_i, a_i) \tag{24}$$

  where α_{actor} represents the learning rate of the global actor network.

  Each agent passes the data of the respective batch to the central agent to complete the parameter update and then executes the step *Pull*.

Fig. 3 shows the change of the average QoE of 400 network traces with training episodes in the training process of NAVSA model. The average QoE was sampled every 10 episodes.After training, we tested the NAVSA model with 120 network traces in the testing set, and compared it with a Pensieve model under the same training set.

Fig. 4 shows the CDF curves of the total QoE values of the two algorithms under each network trace.

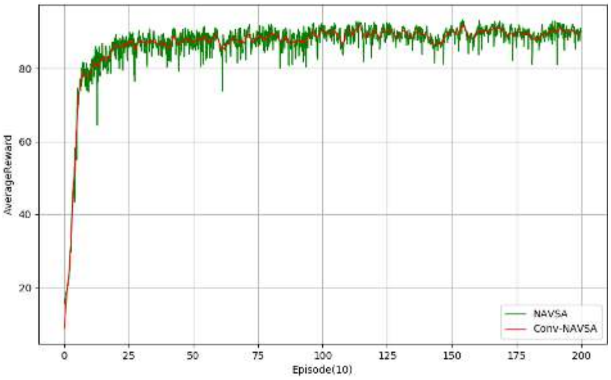


FIGURE 3. The average reward in NAVSA training process varies with the episodes over 60 random network traces. The red curve is the moving average value of the green curve.

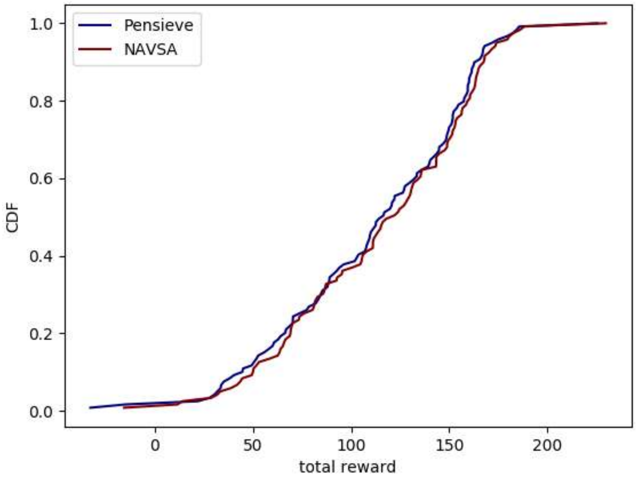


FIGURE 4. Comparing the average QoE values of Pensieve and NAVSA under testing set network traces.

Fig. 5 shows the bitrate selection, buffer occupancy and bandwidth estimation of the two algorithms under a certain network trace. The total QoE of Pensieve under this network trace is 67.1086, while the total QoE of NAVSA is 75.8431. As can be seen, NAVSA's bandwidth calculation is more precise, and brings about the improvement of QoE.

III. SIMULATION

In this section, we used the model trained in Section II to simulate naked-eye 3D video application scenarios on the NS3 platform based on [44], as shown in Fig. 6.

Each client is considered as a user equipment (UE) in the LTE network. The LTE-EPC network simulator in NS3 [45] is used to simulate the fading characteristics of the mobile channel during DASH client's motion.

The remote host (eNb) is set as the DASH server, and the naked-eye 3D video stream of 10 code rate levels between 10-100 Mbps is stored, and each bitrate level is mapped to the highest viewpoints and resolution combination of QoE. In the simulation, the DASH client is set to send a request to the DASH server. After downloading a naked-eye 3D video

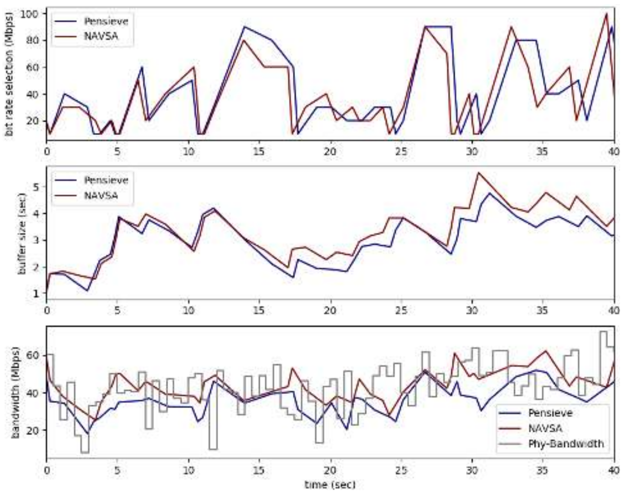


FIGURE 5. Comparing bitrate selection, buffer occupancy and bandwidth estimation of NAVSA and Pensieve on one trace.The gray line Phy-Bandwidth represents the network trace.

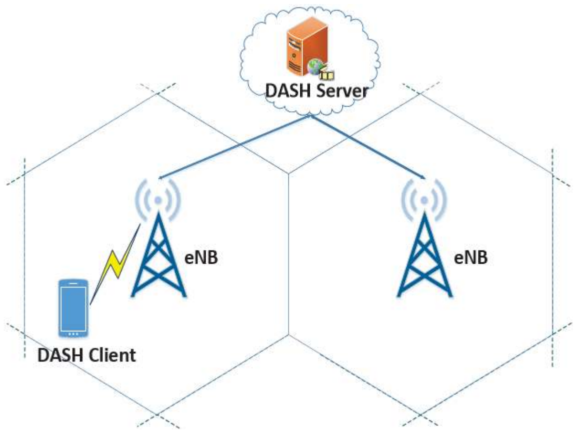


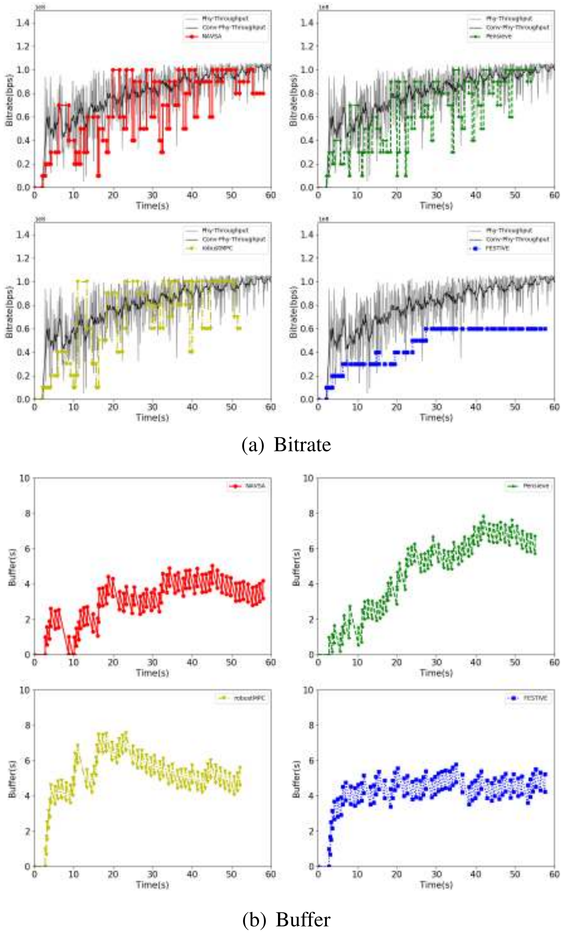
FIGURE 6. DASH application framework.

chunk, the client loads the video streaming into buffer, and playbacks it. Of course, the NS3 platform does not have a real video player, nor does it have a real server to store naked-eye 3D video chunks at various bitrates. We store the size of the naked-eye 3D video chunks of each bitrate level into a text file. When the DASH client makes a request to the server, the server reads the size of the video chunk of a specific bitrate requested by the client, and then send the same size packets to the client. We also set a buffer on the client. When the server finish sending certain size packets, the length of the buffer will increase by a video chunk length. Then when the server is sending the next "video clip data", the cache length will be reduced evenly, to simulate the entire naked-eye 3D video streaming and playback process.

The process of the above server sending packets to the client does exist and has a record at the physical layer, so the client can obtain the TB size at a certain moment by querying the physical layer log and use this to accurately estimate the wireless channel throughput.

TABLE 2. Simulation parameters.

Parameter	Value
System Bandwidth	20MHz
Number of Resource Block	100
Transmission Time Interval	1ms
Number of UE	1
Number of eNB	9
Carrier frequency	2GHz
Cell radius	150m
eNB Tx Power	49dBm
UE Tx Power	23dBm
Transmission Mode	MIMO
α	1
β	4.3
γ	0.02



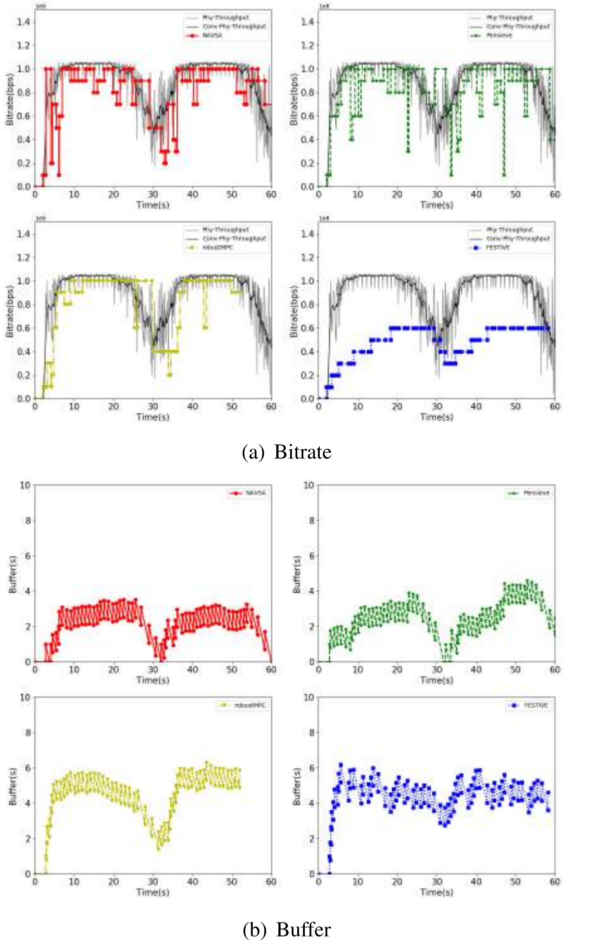
(a) Bitrate

(b) Buffer

FIGURE 7. Profiling bitrate selections, buffer occupancy with four algorithms under Scenario 1. (a) Bitrate. (b) Buffer.

A. SIMULATION PARAMETERS

The NS3 simulation parameters are shown in Table. 2. The naked-eye 3D video chunks stored by the DASH server is 1s in length. The following 3 scenarios have been considered in the simulation:



(a) Bitrate

(b) Buffer

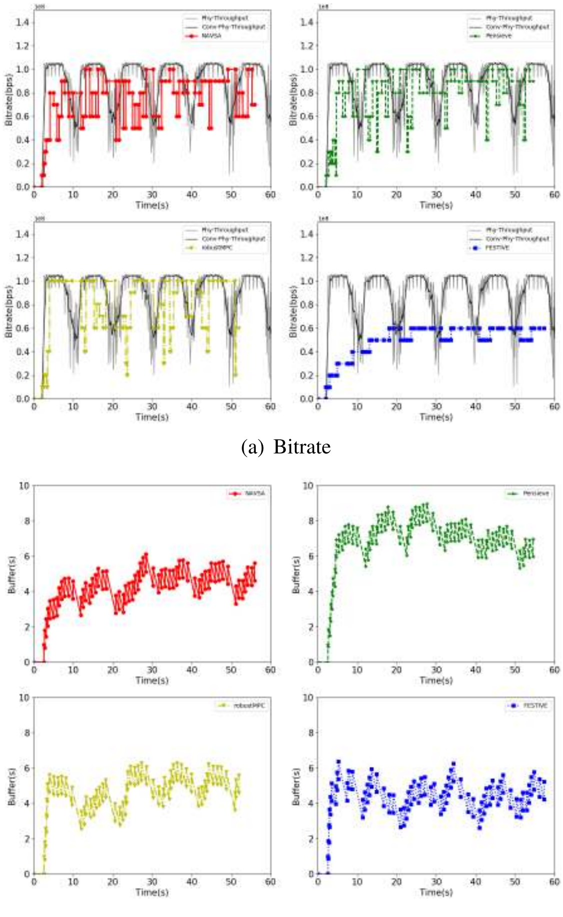
FIGURE 8. Profiling bitrate selections, buffer occupancy with four algorithms under Scenario 2. (a) Bitrate. (b) Buffer.

- **Scenario 1:** The DASH client is a user on the road, and moves from the cell edge to the cell center at a speed of 3 km/h.
- **Scenario 2:** The DASH client is a user of the electric vehicle such as sightseeing bus, and traverses multiple adjacent cells at a speed of 20 km/h.
- **Scenario 3:** The DASH client is a user on the car, and traverses multiple adjacent cells at a speed of 60 km/h.

B. SIMULATION RESULTS

We compare the performance of our proposed algorithm NAVSA with three existing algorithms, including Pensieve, robustMPC (this is a version of MPC that normalizes the calculated throughput by using the maximum prediction error of the past 5 video blocks to conservatively estimate the bandwidth values of the next 5 video blocks), and FESTIVE [33]. The performance comparison is based on the QoE_{hd} metric, and then we analyze the bitrate selection of each algorithm in the three scenarios. The parameters α, β and γ of QoE_{hd} metric are shown in Table. 2, and the average bitrate in each scenario is calculated as follows:

$$R_{ave} = \frac{1}{N}\sum_{n=1}^{N} rep(k_n) \tag{25}$$



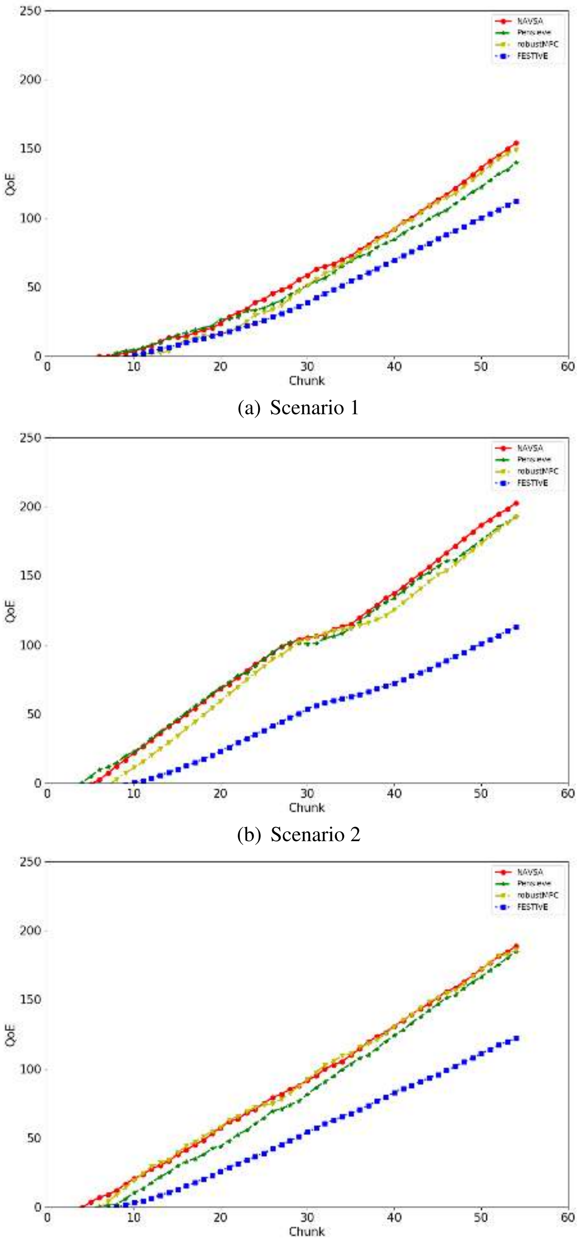
(a) Bitrate

(b) Buffer

FIGURE 9. Profiling bitrate selections, buffer occupancy with four algorithms under Scenario 3. (a) Bitrate. (b) Buffer.

Fig. 7 (a), Fig. 8 (a) and Fig. 9 (a) show the curve of the bitrate selection of the four algorithms in 3 scenarios with the simulation time. The bitrate of each level in the figure is mapped to the highest viewpoint and resolution combination of QoE. The gray line ''Phy-Throuput'' represents the throughput fluctuation of scenarios and the black line is the sliding average of the gray line. In addition to the FESTIVE algorithm, the NAVSA, Pensieve and robustMPC are all QoE-driven. Because the chosen QoE_{hd} metric prefers to select high-quality video chunks, it relatively weakens the penalty of rebuffering and bitrate fluctuations, which result in the three algorithms are very aggressive. They frequently perform bitrate switching, filling low quality video chunks into buffer in order to select a high quality video chunk at the next moment.

The buffer length changes shown in Fig. 8 (b) also illustrate the aggressiveness of the three algorithms, in which the client buffer is always at a lower level and fiercely fluctuates. Because the bandwidth of scenario 3 is the highest for most of the time, the buffer occupancy of the three algorithms can be maintained at a stable value as shown in Fig. 9 (b) when the high bit rate is chosen as far as possible.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

FIGURE 10. QoE varies with the number of video blocks viewed under scenario 1, scenario 2 and scenario 3. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

FESTIVE is a relatively conservative algorithm which tries to keep the buffer length at a stable level. When changing the policy selection, FESTIVE will try to avoid large bitrate switching, and the quality of the selected video is not high.

Fig. 10 shows the variation of QoE of each algorithm with the number of video chunks viewed under 3 scenarios. It can be seen that the final QoE of NAVSA is the highest, and Pensieve and robustMPC are slightly worse due to the QoE_{hd} metric. In order to select high quality video chunks, the average bitrate histogram shown in Fig. 11 also reflects the performance of each algorithm to some extent. In addition,

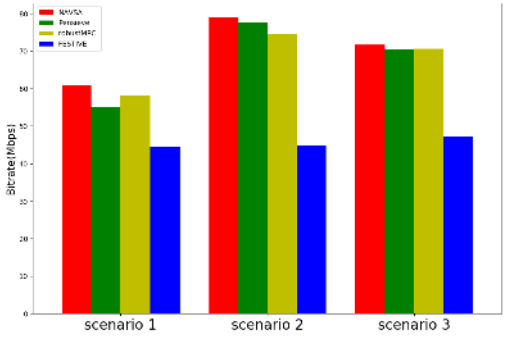


FIGURE 11. Average bitrate of four algorithms under scenario 1, scenario 2 and scenario 3.

we find that the performance of Pensieve is more unstable than NAVSA under the same training set, as shown in Fig. 11. Because the network bandwidth estimate in Pensieve and robustMPC is at the application layer. The same state may have different channel conditions, which leads to errors in the value estimation of the current state. NAVSA directly samples the change of the mobile channel from the physical layer, so that the estimation of the network bandwidth is more accurate, which result in the value error of the state smaller, and the action with higher reward is more reliable, making the QoE of NAVSA the highest in these state-of-the-art algorithms.

IV. CONCLUSION

Taking the naked-eye 3D video transmission as an example, this paper proposes an adaptive video stream algorithm NAVSA combining network-assisted and reinforcement learning. The algorithm can effectively cope with the fluctuation characteristics of 5G mobile channels and improve the QoE of users watching VR video. Additionally, NAVSA requires less information inputs at the same cost of model training time of Pensieve. In the simulation, under QoE_{hd} metric, the QoE value and average bitrate of NAVSA is about 5∼8% better than state-of-the-art Pensieve and robustMPC algorithms, and 30∼40% better than the traditional DASH algorithm FESTIVE.

REFERENCES

[1] Sandvine, Fremont, CA, USA. (2016). *Global Internet Phenomena Report*. [Online]. Available: https://www.sandvine.com/downloads/general/global-internet-phenomena/2012/1h-2012-global-internet-phenomena-report.pdf

[2] Cisco Syst., "Cisco visual networking: Forecast and methodology," San Jose, CA, USA, White Paper, Jun. 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[3] D. Wu, Z. Zhang, S. Wu, J. Yang, and R. Wang, "Biologically inspired resource allocation for network slices in 5G-enabled Internet of Things," *IEEE Internet Things J.*, to be published. doi: 10.1109/JIOT.2018.2888543.

[4] Z. Zhang and L. Wang, "Social tie-driven content priority scheme for D2D communications," *Inf. Sci.*, vol. 480, pp. 160–173, Apr. 2019.

[5] P. Zhang, X. Kang, X. Li, Y. Liu, D. Wu, and R. Wang, "Overlapping community deep exploring based relay selection method towards multi-hop D2D communication," *IEEE Wireless Commun. Lett.*, to be published. doi: 10.1109/LWC.2019.2917907.

[6] J. Dai, K. Niu, and J. Lin, ''Iterative Gaussian-approximated message passing receiver for MIMO-SCMA system,'' in *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 753–765, Jun. 2019.

[7] J. Dai, J. Gao, and K. Niu, ''Learning to mitigate the FAR in polar code blind detection,'' *IEEE Wireless Commun. Lett.*, to be published. doi: 10.1109/LWC.2019.2937299.

[8] P. Zhang, X. Kang, D. Wu, and R. Wang, ''High-accuracy entity state prediction method based on deep belief network toward IoT search,'' *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 492–495, Apr. 2019.

[9] Z. Li, Y. Jiang, Y. Gao, L. Sang, and D. Yang, ''On buffer-constrained throughput of a wireless-powered communication system,'' *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 283–297, Feb. 2019.

[10] T. Taleb, A. Ksentini, M. Chen, and R. Jantti, ''Coping with emerging mobile social media applications through dynamic service function chaining,'' *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2859–2871, Apr. 2016.

[11] D. Wu, H. Shi, H. Wang, R. Wang, and H. Fang, ''A feature-based learning system for Internet of Things applications,'' *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1928–1937, Apr. 2019.

[12] D. Wu, L. Deng, H. Wang, K. Liu, and R. Wang, ''Similarity aware safety multimedia data transmission mechanism for Internet of vehicles,'' *Future Gener. Comput. Syst.*, Vol. 99, pp. 609–623, 2019.

[13] Z. Li, J. Chen, and Z. Zhang, ''Socially aware caching in D2D enabled fog radio access networks,'' *IEEE Access*, vol. 7, pp. 84293–84303, 2019.

[14] J. Steuer, ''Defining virtual reality: Dimensions determining telepresence,'' *J. Commun.*, vol. 42, no. 4, pp. 73–93, Dec. 2010.

[15] R. Anderson, D. Gallup, J. T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S. M. Seitz, ''Jump: Virtual reality video,'' *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–13, Nov. 2016.

[16] E. Bastug, M. Bennis, M. Médard, and M. Debbah, ''Toward interconnected virtual reality: Opportunities, challenges, and enablers,'' *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[17] K. Chaiwong, K. Tamee, S. Punthawanunt, F. H. Suhailin, M. S. Aziz, J. Ali, G. Singh, and P. Yupapin, ''Naked-eye 3D imaging model using the embedded micro-conjugate mirrors within the medical micro-needle device,'' *Microsyst. Technol.*, vol. 24, no. 6, pp. 2695–2699, 2018.

[18] P. Youplao, N. Pornsuwancharoen, I. S. Amiri, V. N. Thieu, and P. Yupapin, ''Naked-eye 3D imaging employing a modified MIMO micro-ring conjugate mirrors,'' *Proc. SPIE*, vol. 10714, Mar. 2018, Art. no. 107140K.

[19] G. Yang, S. Li, and H. Xie, ''Naked-eye 3D multi-viewpoint video display,'' in *Digit. Holography 3-D Imag. Meeting, OSA Tech. Dig.*, May 2015, pp. 1–3.

[20] X. Nan, X. Guo, Y. Lu, Y. He, L. Guan, S. Li, and B. Guo, ''Delay–rate–distortion optimization for cloud gaming with hybrid streaming,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2687–2701, Dec. 2017.

[21] T. Fujihashi, Y. Hirota, and T. Watanabe, ''Bandwidth-based adaptive coding control method for real-time multi-view video streaming,'' in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–7.

[22] C. Liu, L. Tian, Y. Zhou, J. Shi, J. Liu, S. He, Y. Pu, and X. Wang, ''Video content redundancy elimination based on the convergence of computing, communication and cache,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[23] Z. Zhang, T. Zeng, X. Yu, and S. Sun, ''Social-aware D2D pairing for cooperative video transmission using matching theory,'' *Mobile Netw. Appl.*, vol. 23, no. 3, pp. 639–649, Feb. 2018.

[24] A. Wieckowski, J. Ma, H. Schwarz, D. Marpe, and T. Wiegand, ''Fast partitioning decision strategies for the upcoming versatile video coding (VVC) standard,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 4130–4134.

[25] H. Chen, X. Zhang, Y. Xu, Z. Ma, and W. Zhang, ''Efficient mobile video streaming via context-aware RaptorQ-based unequal error protection,'' *IEEE Trans. Multimedia*, to be published. doi: 10.1109/TMM.2019.2928497.

[26] D. Wu, Q. Liu, H. Wang, Q. Yang, and R. Wang, ''Cache less for more: Exploiting cooperative video caching and delivery in D2D communications,'' *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1788–1798, Jul. 2019.

[27] T. Tian and H. Wang, ''Large-scale video compression: Recent advances and challenges,'' *Frontiers Comput. Sci.*, vol. 12, no. 5, pp. 825–839, 2018.

[28] *Information Technology—Dynamic Adaptive Streaming Over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats*, Standard ISO/IEC 23009-1:2014, Motion Picture Experts Group (MPEG), 2012.

[29] M. Hosseini, ''View-aware tile-based adaptations in 360 virtual reality video streaming,'' in *Proc. IEEE Virtual Reality (VR)*, Los Angeles, CA, USA, Mar. 2017, pp. 423–424.

[30] S. Zhao and D. Medhi, ''SDN-assisted adaptive streaming framework for tile-based immersive content using MPEG-DASH,'' in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Berlin, Germany, Nov. 2017, pp. 1–6.

[31] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, ''MPEG DASH SRD: Spatial relationship description,'' in *Proc. 7th Int. Conf. Multimedia Syst.*, May 2016, Art. no. 5.

[32] Z. Li, Q. Wang, and H. Zou, ''QoE-aware video multicast mechanism in fiber-wireless access networks,'' *IEEE Access*, vol. 7, pp. 123098–123106, 2019. doi: 10.1109/ACCESS.2019.2938422.

[33] J. Jiang, V. Sekar, and H. Zhang, ''Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive,'' *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 326–340, Feb. 2014.

[34] 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 *Proc. ACM SIGCOMM Conf.*, Aug. 2016, pp. 272–285.

[35] 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,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, 2015.

[36] 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 J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 719–733, Apr. 2014.

[37] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, ''A control-theoretic approach for dynamic adaptive video streaming over HTTP,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 325–338, 2015.

[38] Y. Wang, Y. Liu, H. Yang, G. Li, Y. Chen, and Z. Liu, ''QoE-driven and network-assisted naked-eye 3D adaptive video streaming over wireless network,'' in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Bologna, Italy, Sep. 2018, pp. 11–16.

[39] H. Mao, R. Netravali, and M. Alizadeh, ''Neural adaptive video streaming with pensieve,'' in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2017, pp. 197–210.

[40] Y. Guo, F. R. Yu, J. An, K. Yang, Y. He, and V. C. M. Leung, ''Buffer-aware streaming in small-scale wireless networks: A deep reinforcement learning approach,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6891–6902, Jul. 2019.

[41] Y. Zhang, P. An, Y. Wu, and Z. Zhang, ''A multiview video quality assessment method based on disparity and SSIM,'' in *Proc. IEEE 10th Int. Conf. Signal Process.*, Beijing, China, Oct. 2010, pp. 1044–1047.

[42] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[43] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, ''Asynchronous methods for deep reinforcement learning,'' in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[44] H. Ott, K. K. Miller, and A. Wolisz, ''Simulation framework for HTTP-based adaptive streaming applications,'' in *Proc. ACM Workshop NS*, 2017, pp. 95–102.

[45] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, ''An open source product-oriented LTE network simulator based on NS-3,'' in *Proc. 14th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, 2011, pp. 293–298.

YITONG LIU received the Ph.D. degree in communication engineering from the Beijing University of Posts and Telecommunications, in 2015, where she is currently a Lecturer. Her research interests include adaptive streaming, video coding, and video quality.

WEI HE is currently pursuing the bachelor's degree with the Beijing University of Posts and Telecommunications. His main research interests include reinforcement learning, data analysis, and algorithm optimization.

YIDI WANG received the B.S. and M.S. degrees from the Beijing University of Posts and Telecommunications, in 2016 and 2019, respectively. Her research interest includes naked-eye 3D adaptive video streaming optimization over wireless networks.

HONGWEN YANG received the B.S. and M.S. degrees from the Beijing University of Posts and Telecommunications (BUPT), in 1984 and 1987, respectively. After graduating, he joined the Faculty of BUPT, where he is currently a Professor. His research mainly focus on wireless physical layer, including modulation and coding, MIMO, and OFDM.