MEC-Assisted Immersive VR Video Streaming Over Terahertz Wireless Networks: A Deep Reinforcement Learning Approach

Jianbo Du, *Member, IEEE*, F. Richard Yu, *Fellow, IEEE*, Guangyue Lu, Junxuan Wang, Jing Jiang, *Member, IEEE*, and Xiaoli Chu, *Senior Member, IEEE*

***Abstract*—Immersive virtual reality (VR) video is becoming increasingly popular owing to its enhanced immersive experience. To enjoy ultrahigh resolution immersive VR video with wireless user equipments, such as head-mounted displays (HMDs), ultralow-latency viewport rendering, and data transmission are the core prerequisites, which could not be achieved without a huge bandwidth and superior processing capabilities. Besides, potentially very high energy consumption at the HMD may impede the rapid development of wireless panoramic VR video. Multiaccess edge computing (MEC) has emerged as a promising technology to reduce both the task processing latency and the energy consumption for HMD, while bandwidth-rich terahertz (THz) communication is expected to enable ultrahigh-speed wireless data transmission. In this article, we propose to minimize the long-term energy consumption of a THz wireless access-based MEC system for high quality immersive VR video services support by jointly optimizing the viewport rendering offloading and downlink transmit power control. Considering the time-varying nature of wireless channel conditions, we propose a deep reinforcement learning-based approach to learn the optimal viewport rendering offloading and transmit power control policies and an asynchronous advantage actor–critic (A3C)-based joint optimization algorithm is proposed. The simulation results demonstrate that the proposed algorithm converges fast under different learning rates, and outperforms existing algorithms in terms of minimized energy consumption and maximized reward.**

***Index Terms*—Asynchronous advantage actor–critic (A3C), computation offloading, deep reinforcement learning (DRL), terahertz (THz) communication, virtual reality (VR).**

I. Introduction

IN THE last few years, the rapid development of fast video processing and omnidirectional cameras has bred a new media form, known as the immersive (360°, or panoramic) virtual reality (VR) video [1]. Using user equipment (UE), such as head-mounted displays (HMDs), smartphones, and personal computers, immersive VR video can provide a 360° omnidirectional immersive experience of, e.g., concerts, exhibitions, sports, etc. [2]. The realization of immersive VR video relies on extremely large amount of data processing and transferring. Current VR systems depend largely on wired transmission, which restricts its applications, while wireless VR can potentially unleash its potential to the maximum [3], [4]. Moreover, processing computationally intensive tasks for immersive VR video on HMDs will result in excessive heat, short battery life, and high unit prices.

In order to deliver immersive VR video over wireless networks, three fundamental challenges need to be urgently addressed. The first challenge lies that it is hard for the current cellular networks to provide sufficient high wireless transmission rate and thus to support the extremely high data rate requirement of immersive VR video transmission [3], e.g., 350 Mb/s [5]. The second major challenge lies in the heavy energy consumption in HMDs. The portion of an immersive VR video that a user is watching needs to be projected to a 2-D plane referred to as the viewport. This portion mapping is called viewport rendering [6], which requires mapping the spherical VR video signal to the viewport pixel by pixel on an HMD, where complex matrix computation is needed and a large amount of energy will be consumed from the HMD's battery. The third challenge lies in the strict latency requirements (e.g., no more than 20 ms) imposed on the total delay of immersive VR video decoding, wireless transmission [7], and viewport rendering. The video decoding and viewport rendering operations typically take about

Manuscript received December 16, 2019; revised March 27, 2020 and May 21, 2020; accepted June 8, 2020. Date of publication June 18, 2020; date of current version October 9, 2020. This work was supported in part by the Natural Science Foundation of Shaanxi Province under Grant 2020JQ-844; in part by the Natural Science Foundation of China under Grant 61901367, Grant 61871321 Grant 61901381, and Grant 61901370; in part by the Science and Technology Innovation Team of Shaanxi Province for Broadband Wireless and Application under Grant 2017KCT-30-02; in part by the National Science Foundation of Shaanxi Province under Grant 2019JM-442; in part by the National Key Research and Development Program of China under Grant 2018YFE0126000; in part by the Key Research and Development Plan of Shaanxi Province under Grant 2017ZDCXL-GY-05-01; and in part by the Xi'an Key Laboratory of Mobile Edge Computing and Security under Grant 201805052-ZD3CG36. *(Corresponding authors: F. Richard Yu; Guangyue Lu; Junxuan Wang.)*

Jianbo Du, Guangyue Lu, Junxuan Wang, and Jing Jiang are with the Shaanxi Key Laboratory of Information Communication Network and Security, School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (e-mail: dujianboo@163.com; tonylugy@163.com; wangjx@xupt.edu.cn; jiangjing@xupt.edu.cn).

F. Richard Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: richard.yu@carleton.ca).

Xiaoli Chu is with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: x.chu@sheffield.ac.uk).

Digital Object Identifier 10.1109/JIOT.2020.3003449

6–100 ms, while wireless transmission will take 100–200 ms. The large end-to-end latency will degrade the Quality of Experience (QoE) of interactive immersive VR video playback significantly [5], [8].

To overcome the challenge in supporting very high data rates, terahertz (THz) communication (0.1–10 THz) [9], [10] has been proposed as a promising enabler of the super-high data rate, ultrareliable, and low delay applications [3], such as immersive VR video. Meanwhile, as a powerful supplement and enhancement of cloud computing [11], multi-access edge computing (MEC) enables HMDs to offload their energy-demanding viewport rendering tasks to MEC servers (MECSs) [12], [13], and consequently offers an opportunity to tackle the last two challenges [14]. However, since the problem of task offloading decision optimization is usually coupled with resource management, making the problem usually nonconvex and NP-hard [15], [16]. Moreover, in fast time-varying and highly dynamic mobile wireless networks, it is challenging to make optimized decisions for binary [17] task offloading and resource allocation at all times. Recently, deep reinforcement learning (DRL) [18], [19] has been employed as an effective tool to obtain optimized solutions to nonconvex and sophisticated optimization problems in highly dynamic wireless environments, especially the problems with continuous state and action spaces. Motivated by their respective benefits, it is quite interesting and inspiring to apply THz communication with MEC to support immersive VR video for QoE promotion, and employ DRL for further performance improvement in problem solving. However, this is a brand new area, and is full of opportunities and challenges, which promote the study of this article. The main contributions are summarized as follows.

1) We propose a THz wireless access-based MEC system to support high quality immersive VR video services, where THz communication is employed to achieve low latency for the elephant immersive VR video data downlink transmission. The MECS will perform proactive immersive VR video content caching, real-time transcoding, and pixel-by-pixel viewport rendering on behalf of HMD, depending on the viewport offloading decision.
2) We formulate a novel optimization problem to minimize the long-term averaged energy consumption of an HMD by jointly optimizing the binary viewport rendering offloading decision for each immersive VR video chunk and the downlink transmit power of the MECS, with data queue stability guaranteed.
3) Considering the complexity, the joint optimization problem is solved by employing asynchronous advantage actor–critic (A3C) algorithm, where multiple deep neural networks (DNNs) are trained asynchronously using gradient descent method, and the optimal viewport rendering offloading decision and transmit power control policy can be obtained with a fast convergence speed and good performance compared with other existing algorithms.

The remainder of this article is organized as follows. Related works are presented in Section II. Section III introduces the system model and Section IV presents the problem formulation. In Section V, the problem is solved efficiently employing the A3C-based algorithm. The simulation results are provided in Section VI. Finally, the article is concluded in Section VII.

II. Related Works

With the rapid development of VR, MEC, THz communication, and DRL, the attempt of using DRL to imporve the performance of MEC, and the idea of using MEC and THz communication to support wireless VR, have attracted increasing attention.

The combination of MEC and DRL has become a hot topic in recent years. Sun *et al.* [20] considered task offloading among neighboring vehicles in vehicular-edge computing systems. Based on the multiarmed bandit theory, they proposed a learning-based task offloading algorithm where vehicles could learn the offloading delay information from their adjacent vehicles in the process of task offloading, and thereby to minimize the average offloading latency. Min *et al.* [21] studied a scenario where multiple edge devices acted as the MECS, and one energy harvesting enabled IoT device could offload its task to one of the edge devices. According to the battery level, previous radio transmit rate, and the predicted amount of harvested energy, they presented a reinforcement learning (RL)-based offloading scheme to obtain the optimal offloading decision for the IoT device. In order to minimize the service delay, Zhao *et al.* [22] investigated the joint optimization of computation resource allocation and network resource assignment in an integrated software-defined MEC system, and proposed a deep *Q* network (DQN)-based algorithm for adaptive resource allocation optimization. Qiu *et al.* [23] considered the task offloading problems in a blockchain-empowered MEC system where the offloading decisions of both mining tasks and data processing tasks were jointly optimized to minimize the long-term cost in task offloading. Leveraging DRL, the optimal offloading decision was obtained based on past experience, and the convergence was sped up by integrating the genetic algorithm in the exploration process of DRL where useless exploration was discarded. Chen *et al.* [24] investigated the joint computation resource allocation and task offloading in a space-air-ground integrated network, where unmanned aerial vehicles (UAVs) served as the MECS and satellites acted as the remote cloud center. To address the system dynamics and the complex control process, they leveraged DRL to learn the optimal offloading decision and actor–critic algorithm to accelerate the learning process.

With the increasing popularity of VR/AR, combining VR-related services with other technologies for performance improvement is a fairly new and valuable area. To support immersive VR video stream processing and transmission, Liu *et al.* [5] proposed a MEC platform operating in both mmWave and sub-6 GHz bands that could maximize the wireless bandwidth utilization and the mobile device's energy efficiency by jointly optimizing the link adaptation, video chunk quality adaptation, and viewport rendering optimization.

Chaccour *et al.* [3] studied the elephant data transmission of VR services in a THz cellular network, derived the PDF of the transmission delay, and showed that THz band wireless communication can support the elephant VR data flow with high reliability and high data rates. Chen *et al.* [25] considered a data correlation-aware resource allocation problem in a VR system in order to maximize the VR users' successful data transmission probability, and developed a Q-learning-based algorithm to find the optimal resource allocation scheme, which could adapt to different users' VR content requests and data correlation. Chen *et al.* [26] studied a VR system where VR users sent their requests to the BS for downlink 360° image transmission, and formulated a problem that jointly optimizes the image transmission and image rotation to maximize the users' successful transmission probability. To solve this optimization problem, they proposed a transfer learning algorithm based on liquid state machine, which could transfer the learned successful transmission into a new one in order to increase the convergence speed.

The above studies have provided some insightful ideas about using DRL to obtain the optimal offloading decision and/or resource allocation in MEC systems [20]–[24], or support VR service from some certain aspects, e.g., using MEC to support VR services for more powerful processing [5], using THz/mmWave to support the elephant VR data flow [3], and using RL to make full use of resources in VR supporting [25], [26]. However, as an increasingly important and highly resource-demanding application nowadays and in the near future, VR still needs to be supported from multiple aspects in order for better QoE. To achieve this goal, some important technologies in 5G/6G can play their own roles from different aspects, where THz can provide extremely high date rate for VR data transmission, MEC can provide VR with stronger processing capability, and DRL algorithms can obtain the optimal solution and thus to further improve the performance of using THz and MEC for VR, where A3C is an effective and outstanding algorithm among the DRL family [27], [28]. Motivated by the above considerations, in this article, we use MEC and THz to improve the performance of VR, and develop an effective algorithm based on A3C for optimal viewport rendering offloading decision making and downlink transmit power control for THz wireless link.

III. System Model

The proposed system is composed of a MECS and an HMD user, with a THz cellular network connecting the two entities. The MECS is located at the THz base station and is connected to the content provider where the compressed original immersive VR video resources are stored through wired fiber links. In our system, all immersive VR videos that the user requests are cached at the MECS [12] and the case when the requested content is not cached and should be retrieved from the content provider is out of the scope of this article. Next, we will introduce the MECS and the THz downlink communication model in detail.

A. MEC Server

The MECS contains a transcoding module, a decision maker, a information acquisition module, and a computing module [5], [29].

Originally, immersive VR video is encoded into space-partitioned tiles, and then each tile is further partitioned into chunks temporally in order to facilitate viewport rendering operation [5], [6]. The small data chunk can be used to avoid large motion-to-photon latency in HMD when HMD's viewport changes rapidly during the interactive of immersive VR video. The decoding module in MECS is used for decompressing each chunk from the original compressed immersive VR video stream, including viewport tiles and nonviewport tiles, and provides the uncompressed full-resolution immersive VR video to HMD [30], as shown in Fig. 1. The output chunk data stream from the transcoding module is first assembled into a series of chunks where each chunk is with a duration of one Group of Pictures (GOPs) and consisting of four frames, and then transmitted over high-speed THz link to the HMD [5].

For full-resolution videos, the tiles within the viewport region are called viewport tiles, which are with high resolution and need the aforementioned viewport rendering operation, while the tiles outside the viewport are called standby tiles, which is with low quality and are not necessary to be rendered. When the viewport video data could not be able to keep pace with the rapid variation of viewport in HMD, the low-quality standby tiles will be mapped to the viewport at HMD for smoothing viewport viewing experience. Viewport rendering can be performed at different places. Fig. 1(a) illustrates local rendering where the uncompressed full-resolution immersive VR video data are first transmitted to HMD and viewport rendering will be performed locally at the HMD. Fig. 1(b) shows the situation where viewport is rendered on the MECS, where the uncompressed viewports tiles is first rendered at the "Viewport Rendering" module, and then the rendered viewport as well as the uncompressed standby tiles are then transmitted to the HMD. If viewport rendering is conducted by MECS, the computing module in the MECS will work and perform viewport rendering operation for the HMD.

The information acquisition module is in charge of collecting the power and the viewport information of the HMD, and estimates the THz downlink quality information from the uplink reference signals broadcasted by the HMD.

According to the collected link quality information, and HMD's viewport information, the decision maker performs optimization under given latency constraints. The optimization terms include downlink transmit power control, and viewport rendering offloading optimization, i.e., should the viewport be rendered on the MECS and then be transmitted to the HMD, or first be transmitted to HMD and then rendered there.

B. THz Band Channel Model

In this section, we present some basic knowledge and the characteristics of the THz band channel.

The THz wireless propagation model is a multipath model, including LOS and the reflected path rays. Since the scattered and diffracted rays play insignificant roles on the received signal, similar to [9], they are not taken into consideration in our system model. In our system, time is slotted where the length and the index of a time slot are denoted as Δt and t, and the set and number of time indices are denoted as $\mathcal{T}$ and T, respectively. We consider a quasistatic scenario, where the environment keeps static at each time slot but varies between different time slots.

For a distance d at time slot t, supposing there are totally $U^d(t)$ THz subwindows[1] [9] and the uth subwindow is composed by $N_{\text{ref}}(t)$ reflected rays, the multipath channel response model can be given by [9]

$$\begin{aligned} h_u^d(t) &= \alpha_{u,\text{LOS}}^d(t)\delta\Big(t-\tau_{u,\text{LOS}}^d(t)\Big)\mathbb{1}_{u,\text{LOS}}^d(t) \\ &\quad + \sum_{q=1}^{N_{\text{ref}}(t)} \alpha_{u,q}^d(t)\delta\Big(t-\tau_{u,q}^d(t)\Big) \end{aligned} \tag{1}$$

where $\mathbb{1}(\cdot)$ is the indicator function and $\mathbb{1}_{u,\text{LOS}}^d(t)$ equals 1 or 0 denotes the presence of LOS path or not. The terms $\alpha_{u,\text{LOS}}^d(t)$ and $\alpha_{u,q}^d(t)$ are attenuation factors indicating the attenuation of the LOS path and the qth reflected ray of the uth frequency subwindow, and $\tau_{u,\text{LOS}}^d(t)$ and $\tau_{u,q}^d(t)$ are the propagation delay of the LOS path and the qth reflected ray, respectively. The set and number of all multipath components for the uth subwindow is denoted by $\mathcal{N}_u^d(t)$ and $N_u^d(t)$, and we have $N_u^d(t) = \mathbb{1}_{u,\text{LOS}}^d(t) + N_{\text{ref}}(t)$. In this article we suppose $N_{\text{ref}}(t) = 2$. In the HMD side, the received signal is constructed by a superposition of the LOS and the reflected rays, and the material parameters are available and could refer to [10].

Invoking the Wiener–Khinchin theorem [10], the attenuations of the LOS and the qth reflected rays of the uth subwindow can be given by [10]

$$\begin{aligned} \alpha_{u,\text{LOS}}^d(t) &= |H_{\text{LOS}}(f_u, t)| \\ \alpha_{u,q}^d(t) &= \left|H_{\text{ref},q}(f_u, t)\right| \end{aligned} \tag{2}$$

where f_u is the center frequency of the uth subwindow, and $H_{\text{LOS}}(f_u, t)$ and $H_{\text{ref},q}(f_u, t)$ are the corresponding transfer functions, which are functions of f_u.

The transfer function of LOS channel $H_{\text{LOS}}(f_u, t)$ composes the spreading loss function $H_{\text{spr}}(f_u)$ and the molecular absorbtion loss function $H_{\text{abs}}(f_u)$, which is given by [10]

$$\begin{aligned} H_{\text{LOS}}(f_u, t) &= H_{\text{spr}}(f_u) \cdot H_{\text{abs}}(f_u) \cdot e^{-j2\pi f_u \tau_{u,\text{LOS}}^d(t)} \\ &= \frac{c}{4\pi f d} \cdot e^{-\frac{1}{2}k(f_u)d} \cdot e^{-j2\pi f_u \tau_{u,\text{LOS}}^d(t)} \end{aligned} \tag{3}$$

where c is the speed of light, d is the distance between the THz base station (the transmitter) and HMD (the receiver), $\tau_{u,\text{LOS}}^d(t) = (d/c)$ is the time of arrival of the LOS ray, and $k(f_u)$ is the frequency-dependent medium absorption coefficient which depends on the material of the transmission medium at molecular levels.

The transfer function of the reflected path can be obtained as follows. Denote $d_{q,1}$ as the distance between the THz base station and the reflector, and $d_{q,2}$ as the distance between the reflector and HMD, and d_q as the distance between the THz base station and HMD, then the transfer function of the qth reflected ray, $H_{\text{ref},q}(f_u, t)$, is given by

$$\begin{aligned} H_{\text{ref},q}(f_u, t) = &\frac{c}{4\pi f_u\big(d_{q,1}+d_{q,2}\big)} \cdot e^{-j2\pi f_u \tau_{u,q}^d(t) - \frac{1}{2}k(f_u)\left(d_{q,1}+d_{q,2}\right)} \\ &\times R_{u,q}(f_u) \end{aligned} \tag{4}$$

where $\tau_{u,q}^d(t) = \tau_{u,\text{LOS}}^d(t) + (d_{q,1} + d_{q,2} - d_q)/c$ is the time of arrival of the reflected ray, and $R_{u,q}(f_u)$ is the rough surface reflection coefficient. According to Kirchhoff scattering theory [10], $R_{u,q}(f_u)$ can be obtained by multiplying the smooth surface reflection coefficient $\eta_{u,q}(f_u)$ with the Rayleigh roughness factor $\rho_{u,q}(f_u)$ as follows:

$$\begin{aligned} R_{u,q}(f_u) &= \eta_{u,q}(f_u) \cdot \rho_{u,q}(f_u) \\ &= -\exp\left(\frac{-2\cos\big(\theta_q\big)}{\sqrt{n_t^2-1}}\right) \cdot \exp\left(-\frac{8\pi^2 f_u^2 \sigma^2 \cos^2\big(\theta_q\big)}{c^2}\right) \end{aligned} \tag{5}$$

where θ_q is the angle of the qth reflected ray and can be obtained by $\theta_q = (1/2)\cos^{-1}([d_{q,1}^2 + d_{q,2}^2 - d_q^2]/[2d_{q,1}d_{q,2}])$, and n_t is referred to as the refractive index which depends on the frequencies and the transmit medium [10], and σ is a parameter called the rough surface height standard deviation coefficient [10].

C. Transmit Rate of THz Channel

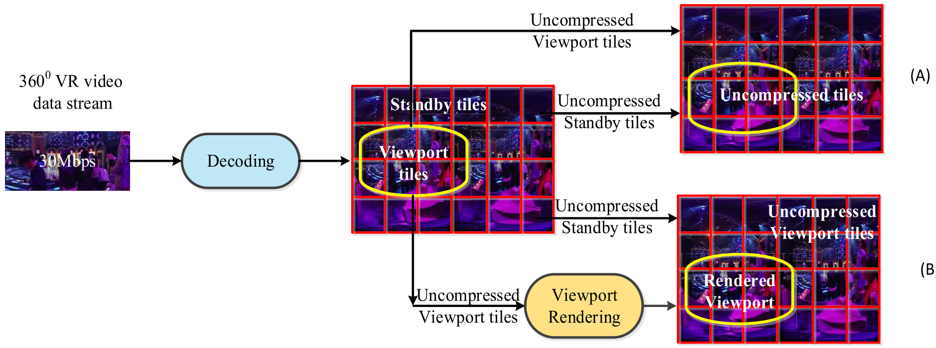
To obtain the THz transmit rate, we first derive the expression of SINR in downlink THz wireless transmission. For a distance d in slot t, there are $U^d(t)$ subwindows can be used for data transmission. Here, the number $U^d(t)$ is the ratio between the total available bandwidth and the bandwidth of a general subwindow [10]. In the THz band, the number of subwindows for a general distance d is at the order of multiple tens. As in [10], the bandwidth of each subwindow can be set as $B_g = 10$ GHz, which is less than the coherence bandwidth and therefore, the intersymbol interference (ISI) [31] can be eliminated and narrowband communication on each subwindow can be enabled. However, severe interband interference (IBI) caused by the power leakage from the adjacent subwindows occurs and could not be ignored. Han *et al.* [10] have shown that the IBI from adjacent subwindows can be approximated as a Gaussian distributed random variable, and the distribution of the IBI on the uth subwindow follows:

$$I_u^d(t) \sim \mathcal{N}\left(0, \int_{f_u} \sum_{v,v\neq u}^{U^d(t)} P_v(t) \left| G_v(f_u) \sum_{m\in\mathcal{N}_u} \alpha_{v,m}^d(t) \right|^2 df_u \right) \tag{6}$$

where G_v is the waveform, $P_v(t)$ denote the transmit power allocated on the vth subwindow, and the path attenuation vector is given by

$$\begin{aligned} \boldsymbol{\alpha}_v^d(t) &= \left\{\alpha_{v,m}^d(t),\ m \in \mathcal{N}_u\right\} \\ &= \left[\alpha_{v,\text{LOS}}^d(t), \alpha_{v,1}^d(t), \ldots, \alpha_{v,N_{\text{ref}}^{(v)}}^d(t)\right]. \end{aligned} \tag{7}$$

[1] In the THz band, the basic unit of wireless resource allocation is called a subwindow, and the available bandwidth and the number of subwindows changes with the variation of distance d.



360^0 VR video data stream
30Mbps
Decoding
Standby tiles
Viewport tiles
Uncompressed Viewport tiles
Uncompressed Standby tiles
Uncompressed tiles
(A)
Uncompressed Standby tiles
Uncompressed Viewport tiles
Rendered Viewport
(B)
Uncompressed Viewport tiles
Viewport Rendering

Fig. 1. MEC-based immersive VR video transcoding framework.

Based on the above defined channel response $h_u^d(t)$ and interference $I_u^d(t)$, the instantaneous SINR $\gamma_u^d(t)$ can be given by

$$\gamma_u^d(t) = \frac{G_t(t)G_r(t)\left|h_u^d(t)\right|^2 P_u(t)}{G_t(t)G_r(t)I_u^d(t) + B_g n_0} \tag{8}$$

where n_0 is the power spectral density of Gaussian white noise. In THz band, the noise mainly comes forme the molecular absorption, which is frequency dependent [32]. Another major noise source results from the receiver and depends on the production technology. As in [10], we suppose the noise can be approximated as Gaussian white noise. Moreover, let ϵ_u denote the bit error rate (BER) on the uth subwindow, and for a given BER, the transmit rate of each Hz on the subwindow u in slot t (in bit/Hz) is given by [33]

$$k_u^d(t) = \log_2\left(1 - \frac{1.5\gamma_u^d(t)}{\ln(5\epsilon_u)}\right) \tag{9}$$

where the expression in $\log(\cdot)$ stands for the maximum supported constellation of MQAM. Consequently, the data rate of each subwindow at distance d in slot t can be given by

$$R^d(t) = B_g \sum_{u=1}^{U^d(t)} k_u^d(t). \tag{10}$$

Remark 1: The potential of using THz frequencies to support super high data rate applications, including immersive VR video, while ensuring ultrareliable, low-latency communications has been demonstrated in [2], [3]. By analyzing the delay and reliability, the authors demonstrated that it is feasible to provide satisfactory immersive VR services by operating on THz cellular networks. Standing on the shoulders of giants, we build our MEC system on THz frequency, and shows our system performs well in energy minimizing in the simulations.

IV. Problem Formulation

In this section, we first analyze the latency and energy consumption in different scenarios, based on which we give our problem formulation. Finally, we transform our problem into a new form that is easy to solve.

A. Latency and Energy Consumption Models

As was mentioned, we suppose all the requested immersive VR video tiles have been cached at the MECS, so the delay and energy consumption during data delivery from the content provider to the MECS does not need to be considered. Recall that the computational-intensive immersive viewport rendering process can be performed on the HMD as in Fig. 1(a) or on the MECS as in Fig. 1(b). We use a binary variable $\eta(t)$ to indicate where viewport rendering is performed, i.e., $\eta(t) = 1$ indicates viewport rendering is offloaded to the MECS, and $\eta(t) = 0$ means the viewport is rendered locally at the HMD. Next, we will analyze the energy consumption under different scenarios.

1) Viewport Rendering at HMD Locally: In this case, the task requesting session includes the following steps: decoding the original data representation by the MECS, transmitting the uncompressed immersive VR video chunks, including viewport tiles and standby tiles to HMD over the THz link, and rendering the viewport on the HMD.

In order to prepare the requested data of a chunk, MECS first needs to decode the original cached immersive VR video data. Assume the playback duration of a chunk equals to the length of a time slot Δt, and denote the bitrate of one original full-resolution video chunk as b (in bps), then we need to decode $b \cdot \Delta t$ bits in order to obtain the uncompressed original data. Denote the size of the uncompressed original data as $x_v(t) + x_s(t)$ bits, where $x_v(t)$ (in bit) and $x_s(t)$ (in bit) are the data volume of viewport tiles and the standby tiles, respectively. Denote the MECS's decoding speed as v^{de} (in bps), and then it needs $[(b \cdot \Delta t)/v^{de}]$ seconds for the MECS to perform data decoding. To transmit the decoded original uncompressed immersive VR video chunk over the THz band from MECS to HMD requires $([x_v(t)+x_s(t)]/[R_d(t)])$ seconds. Then HMD needs to perform viewport rendering. Denote the processing capability of HMD as z_l (in CPU cycles/s), and let the data volume that one CPU cycle can process as b_{zl} (in bits/cycle), then local viewport rendering will consume $([x_v(t)]/[z_l \cdot b_{zl}])$ seconds. Consequently, the overall energy consumption of HMD in local viewport rendering is given by

$$E_l(t) = \frac{b \cdot \Delta t}{v^{de}} \cdot P_{id} + \frac{[x_v(t) + x_s(t)]}{R^d(t)} \cdot P_b + \frac{x_v(t)}{z_l \cdot b_{zl}} \cdot \xi \tag{11}$$

where P_{id} (in Watt) is the idle power of HMD, and $[(b \cdot \Delta t)/v^{de}] \cdot P_{id}$ is the energy consumed by HMD in waiting for the data when MECS performs viewport rendering. Denote the data receiving power of HMD as P_b (in Watt), then the energy for receiving the original uncompressed immersive VR video is $([x_v(t) + x_s(t)]/[R^d(t)]) \cdot P_b$. Moreover, denote ξ (in W) as the power of HMD in task processing, then the required energy for local viewport rendering can be calculated as $([x_v(t)]/[z_l \cdot b_{zl}]) \cdot \xi$.

In addition, HMD maintains a first in first out (FIFO) data buffer for caching the not yet rendered tasks. At the beginning of time slot t, the queue length of HMD's buffer is $Q_l(t)$ (bits), i.e., the amount of tasks that have not yet been executed till the beginning of solt t, then $Q_l(t+1)$ varies dynamically according to

$$Q_l(t+1) = \left[Q_l(t) + (1-\eta(t)) \cdot (x_v(t) - z_l \cdot b_{zl} \cdot \Delta t)\right]^+ \quad (12)$$

where $[x]^+ = \max(x, 0)$.

2) Viewport Rendering on MEC Server: When viewport rendering is performed on MECS, MECS should first decode the original data with the viewport data volume as $x_v(t)$ and the standby data volume as $x_s(t)$, which is the same as in local rendering, and then render the viewport data, and the volume of the rendered viewport data is denoted as $x_r(t)$. Then, the standby data together with the rendered viewport data will be transferred to HMD through the THz link. The only task that HMD should implement is to receive those data. However, extra energy will be consumed when HMD waiting for the MECS to decode the original data and perform viewport rendering. Denote the computation capability of MECS as z_f (in CPU cycles/s), and the data volume one CPU cycle can process as b_{zf} (in bits/cycle), the overall energy consumption of HMD in MECS viewport rendering is given by

$$E_f(t) = \left[\frac{b \cdot \Delta t}{v^{de}} + \frac{x_r(t)}{z_f \cdot b_{zf}}\right] \cdot P_{id} + \frac{x_r(t) + x_s(t)}{R^d(t)} \cdot P_b. \quad (13)$$

Similarly, MECS maintains a data queue $Q_f(t)$, which evolves according to

$$Q_f(t+1) = \left[Q_f(t) + \eta(t) \cdot \left(x_r(t) - z_f \cdot b_{zf} \cdot \Delta t\right)\right]^+. \quad (14)$$

B. Problem Formulation and Transformation

Our objective is to minimize the long-term averaged energy consumption while ensuring the QoE of the HMD, by a joint optimization on the viewport rendering offloading decision $\boldsymbol{\eta} = \{\eta(t),\ t \in \mathcal{T}\}$ and the downlink transmit power control $\mathbf{P}_{tx} = \{P_{tx},\ t \in \mathcal{T}\}$. Our problem is formulated as

$$\begin{aligned}(\mathcal{P}_1):\ \min_{\boldsymbol{\eta}, \mathbf{P}_{tx}}\ & \frac{1}{T}\sum_{t\in\mathcal{T}}(1-\eta(t)) \\ & \times\ [E_l(t) + \omega_l H_l(t)] + \eta(t)\left[E_f(t) + \omega_f H_f(t)\right] \\ \text{s.t.}\ & (\text{C1}): \eta(t) \in \{0,1\}, t \in \mathcal{T} \\ & (\text{C2}): P_{tx} \in (0, P_{\max}],\quad t \in \mathcal{T}.\end{aligned} \quad (15)$$

In problem $(\mathcal{P}_1)$, $H_l(t)$ and $H_f(t)$ are the punishment terms on HMD's energy consumption in local rendering and MECS rendering, respectively, which are used to avoid the long latency caused by HMD/MECS accepting tasks when their queues are too long to process. The two coefficients ω_l and ω_f (in J/bit) are the corresponding punishment factors. At the end of each time slot t, when the queue is nonempty, the energy consumption of HMD will gain a punishment $\omega_l H_l(t)$ or $\omega_f H_f(t)$. Thus, $H_l(t)$ and $H_f(t)$ represent the amount of the backlogged data, i.e., the data has not been executed by HMD and MECS at the end of time slot t, respectively. According to their meanings, we have $H_l(t) = Q_l(t+1)$ and $H_f(t) = Q_f(t+1)$. Consequently, at the end of each time slot t, once the queues are not empty, the HMD's energy consumption will gain a punishment $\omega_l H_l(t)$ in local rendering, or a punishment $\omega_f H_f(t)$ in MECS rendering, respectively. The two punishment terms can avoid the HMD or the MECS to accept excessive viewport rendering tasks, and can avoid queue overflow effectively, and thus to guarantee fluent playback of immersive VR videos on HMD [34].

V. DRL-Based Joint Optimization

In this section, we first introduce some basics about DRL and the newly emerging DRL algorithm A3C, then we propose an A3C-based joint viewport rendering offloading decision and transmit power control algorithm to solve it.

A. Deep Reinforcement Learning

RL [35] is a learning framework where an agent interacts periodically with the environment, by continuously making decisions, observing the rewards, and then automatically adjusting its parameters, finally to obtain the optimal policy that can maximize the long-term expected cumulative reward that the agent could get. However, the learning process of RL converges too slow since it has to explore and obtain knowledge of the entire system. In recent years, deep learning [36] is introduced and deemed as a promising technique to break the curse of high dimensionality when used in RL, which is known as DRL. DRL employs DNNs as the function approximator [37] to train the learning process and updating parameters, so it could not only improve the poor performance of traditional RL methods in dealing with high dimension state space or large action spaces, and especially, DRL could also manage continuous state and action spaces effectively. As a consequence, DRL has been adopted in broad areas, such as robotics, VR, computer vision, etc.

In the area of wireless communications and networking, DRL has also been employed as an effective technique to handle various issues and challenges. Modern wireless communication networks become more large scale, heterogeneous, high dynamic, and complicated, and need to provide various services and make decisions for large quality of UEs, to achieve different goals of different networks. However, the heterogeneity, high dynamic, and uncertainty of wireless networks make conventional approaches, such as dynamic programming, value iteration, etc., for decision making and resource management inefficient or even inapplicable, since complete and perfect system knowledge are required by these algorithms; on the other hand, since the decision-making problems are usually with both integer and continuous variables, along with the large-scale and high complexity, making traditional RL impotent and powerless. As a result, DRL has been developed as an alternative solution to overcome the challenges and has been widely used in various communication systems, where superior performance could be obtained [18], [38].

Remark 2: In the following elaboration of A3C, the time indices are represented using subscript instead of being put in parentheses for notational simplicity. For the notations that have appeared above, we still use the same notations to keep consistent.

B. A3C Algorithm

DRL algorithms are realized by a combination of RL algorithms with DNNs and are unstable since online RL updates are strongly correlated, which can be solved by experience replay as in DQN [37]. However, experience replay consumes more memory and requires off-policy learning policies, and update is performed based on the data generated by an older policy. Asynchronous execution is a promising way to replace experience replay, where multiple agents work in parallel employing different exploration policies to learn from the environment, so asynchronous execution can decorrelate data since parallel agents will experience different states and thus can stabilize the training process of DRL.

actor–critic (AC) [27], [39] algorithm is proposed based on policy-based model-free algorithms, and meanwhile also combines the advantage of the value-based algorithm. In the AC algorithm, policies are directly parameterized as $\pi(a_t|s_t;\theta)$ and the parameter θ is updated by gradient ascent on the difference between the expected accumulated return R_t and the learned value function $V(s_t,\theta_v)$, i.e., $R_t - V(s_t,\theta_v)$, under the policy $\pi(a_t|s_t;\theta)$. The actor is the learned policy function $\pi(a_t|s_t;\theta)$, under which the action that can obtain the maximum reward will be picked out and performed. The action will trigger changes in the environment, and meanwhile, the agent will receive the corresponding reward. Based on the difference between the reward and the learned value function $V(s_t,\theta_v)$, i.e., the TD-error, the critic will evaluate the policy and update the parameter θ of the actor network in order to improve the probability of choosing actions that generate higher reward and meanwhile, update the parameter θ_v of the critic network so as to receive more accurate estimation value. As thus, the AC algorithm learns the policy and the value function, in the process of iteration, the critic could obtain more accurate estimation and the actor could make more judicious decision untill the system converges.

A3C [27] was proposed based on the AC algorithm. Different from the AC method with only one agent, A3C employs multiple agents with different policies concurrently to train the DNNs asynchronously, thus can explore different parts of the environment, so that the updates are less correlated than using a single agent as AC does, and the training time are significantly reduced. Similar to other asynchronous strategies, there is a global network that stores the network parameters. Each time once the agent updates its parameters of the actor and the critic networks, it submits the parameters to the global network, based on which the global network updates the global parameters and then sends them to the agents in order to make sure that all the agents can share same policy. This process repeats until a terminal state or the maximum action index $t_{\max}$ is reached.

A3C has many advantages over other existing DRL algorithms. Compared with value-based algorithms, such as Q-learning, DQN [37], SARAS [38], etc., where optimization relies on value functions and the optimal policies is obtained only when all the states are traversed, leading to high complexity when the state and action space is large. Meanwhile, when the state space and/or action space is continuous, value-based algorithms could not play their effects [40]. A3C based on the policy-based method where policies are directly parameterized, so it can deal with continuous state and/or action spaces, and can learn policies directly and effectively in discrete systems with large numbers of states or actions. Compared with policy-based algorithms, such as REINFORCE [38], where updating are performed based on episode, A3C employs step-based updating, so the efficiency is improved significantly. Compared with AC, the multiple agents parallel training brings less training delay and more effective exploration.

Next, we will explain how the A3C algorithm works. At each time slot t, the environment is in state s_t, which has an estimated state value $V(s_t;\theta_v)$ with parameter θ_v. Under s_t, the agent performs a feasible action a_t according to policy $\pi(a_t|s_t;\theta)$ with parameter θ, and then the environment may transfer to an attainable following state s_{t+1} by certain probabilities, and receives a feedback in the form of an immediate reward r_t. The state value function of A3C is defined as

$$\begin{aligned} V(s_t;\theta_v) &= E[G_t|s=s_t,\pi] \\ &= E\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k}\middle|s=s_t,\pi\right] \end{aligned} \tag{16}$$

where G_t is the discounted accumulated return of step t, and $\gamma\in[0,1]$ is called the discount factor, reflecting the importance of immediate reward and future rewards. When $\gamma=0$, only the next following reward is considered, and when $\gamma=1$, all the future rewards are equally important no matter how soon they occur.

A3C employs k-step reward for parameter updating, where both the policy and the value function will be updated after every $t_{\max}$ actions or when a terminal state is reached. The k-step reward is defined as

$$R_t = \sum_{i=0}^{k-1}\gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta_v) \tag{17}$$

where k is upper bounded by $t_{\max}$. The A3C algorithm is based on AC architecture and also defines the advantage function A_t as the difference between the real reward R_t and the estimated value function $V(s_t;\theta_v)$. The advantage A_t can be given by $A(s_t,a_t;\theta,\theta_v) = R_t - V(s_t;\theta_v)$, which is used to measure how favorable a function a_t is compared with the value of the current state, from the standpoint of long-term expected reward. Using advantage A_t could improve the agent's learning capability so as not to overestimate or underestimate the quality of the action, and thus to enhance the decision-making capabilities.

Based on the advantage function, the loss function [28] for policy (or for actor) can be given by

$$f_\pi(\theta) = \log\pi(a_t|s_t;\theta)(R_t - V(s_t;\theta_v)) + \beta H(\pi(s_t;\theta)) \tag{18}$$

where $H(\pi(s_t;\theta))$ is an entropy term which is used for exploration during the training process and thus to avoid possible premature convergence to suboptimal policies, and β is used to control the strength of the entropy regularization term, which could help to manage exploration and exploitation in training, and higher β tends to exploration. Based on $f_\pi(\theta)$, the accumulated gradient of policy loss functions is given by

$$\begin{aligned} d\theta \leftarrow\ & d\theta + \nabla_{\theta'} \log \pi(a_t|s_t;\theta')(R_t - V(s_t;\theta_v)) \\ & + \delta\nabla_{\theta'} H\big(\pi\big(s_t;\theta'\big)\big) \end{aligned} \tag{19}$$

according to which the actor network can be updated.

The loss function for the estimated value function (i.e., for critic) is defined as

$$f_v(\theta) = (R_t - V(s_t;\theta_v))^2 \tag{20}$$

based on which the accumulated gradient of actor's loss functions is given by

$$d\theta_v \leftarrow d\theta_v + \frac{\partial(R_t - V(s_t;\theta_v))^2}{\partial\theta_v'} \tag{21}$$

and according to which the critic network can be updated. In the above, updating (19) and (21), θ' and θ_v' are the thread-specific actor and critic network parameters of each agent, and θ and θ_v are the parameters of the global actor and critic network, respectively.

Next, we use the standard noncentered RMSProp algorithm to perform training for both actor and the critic. By minimizing the two loss functions, parameters are updated based on the above-accumulated gradients. The estimated gradient under RMSProp can be given by [27], [41]

$$g = \alpha g + (1-\alpha)\Delta\theta^2 \tag{22}$$

where α is the momentum, and $\Delta\theta$ is the accumulated gradients of the policy or value loss function.

Based on the obtained g, update is performed according to

$$\theta \leftarrow \theta - \eta\frac{\Delta\theta}{\sqrt{g+\epsilon}} \tag{23}$$

where η is the learning rate, and ϵ is a tiny positive number used to avoid errors when denominator equals to 0 [27], [41].

The algorithm structure of the A3C-based optimization in this article is illustrated in Fig. 2.

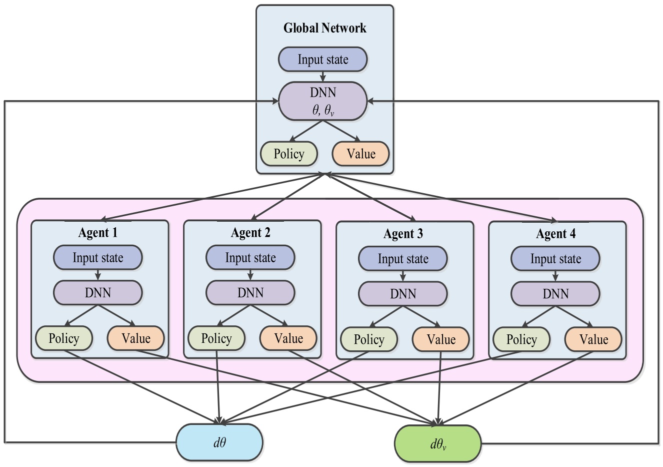


Fig. 2. Structure of the A3C-based optimization algorithm.

Algorithm 1 A3C-Based Viewport Rendering Offloading and Transmit Power Control Algorithm

```
Initialization:
Initialize the global actor network and global critic network with parameters $\theta$ and $\theta_v$.
Initialize global shared counter as $T = 0$ and thread-specific counter as $t = 1$.
Initialize the thread-specific actor and thread-specific critic network parameters $\theta'$ and $\theta_v'$.
Initialize $T_{max}$, $\eta$, $\alpha$, $\epsilon$, $\gamma$, and $t_{max}$, respectively.
Iteration:
while $T < T_{max}$ do
    for each agent do
        Set gradients of two global networks: $d\theta = 0$, $d\theta_v = 0$.
        Synchronous thread parameters by global parameters $\theta' = \theta$ and $\theta_v' = \theta_v$.
        obtain the system state $s_t$.
        for $t \le t_{max}$ do
            Perform $a_t$ according to policy $\pi(a_t|s_t;\theta')$ in thread actor network.
            Obtain reward $r_t$ and new state $s_{t+1}$.
            $t = t + 1$.
        end for
        $R = \begin{cases} 0, & \text{for terminal state } s_t, \\ V(s_t,\theta_v'), & \text{for non} - \text{terminal state } s_M. \end{cases}$
        for $t = t_{max}, t \ge 1$ do
            $R = r_t + \gamma R$.
            Obtain accumulate gradient wrt $\theta'$ based on (19);
            Obtain accumulate gradient wrt $\theta_v'$ based on (21);
        end for
        Asynchronous update $\theta$ and $\theta_v$ according to (23), respectively.
        $T = T + 1$.
    end for
end while
```

C. A3C-Based Viewport Rendering Offloading and Transmit Power Control

We consider the MECS as the decision-making agent, which interacts with the immersive VR video environment. The goal is to select actions in a fashion that maximize the cumulative future reward. The detailed information is given as follows.

1) System State: The system state at the tth time slot, denoted $s_t \in \mathcal{S}$ where $\mathcal{S}$ is the state space, represents a set of states, including the distance $\mathcal{D}(t) = \{d_t, d_{q,1}(t), d_{q,2}(t)\}$, $q \in \mathcal{N}_{\text{ref}}$, the attenuation $\boldsymbol{\alpha}(t)$, the number of subwindows $U^d(t)$, the lengths of the task queues $\mathcal{Q}(t) = \{Q_l(t), Q_f(t)\}$, and is described by s tuple $s_t \triangleq \{\mathcal{D}(t), \boldsymbol{\alpha}(t), U^d(t), \mathcal{Q}(t)\}$. The system state s_t ban be observed at the beginning of the tth time slot.

2) Actions: At each time slot t, the action $a_t \in \mathcal{A}$ includes the viewport rendering offloading and the downlink power allocation and can be given by $a_t \triangleq \{\eta(t), p_{tx}(t)\}$. Accordingly, the available actions for the tth time slot are given as $\{\boldsymbol{\eta}, \mathbf{P}_{tx}\}$.

3) Actor–Critic Network: We use two DNNs with weights θ and θ_v to approximate the stochastic policy function (actor) and the value function (critic). The output layer that estimates

TABLE I
SIMULATION PARAMETER SETTINGS

Parameter	**Value**
Length of a time slot, $\triangle t$	0.133 s [5]
HMD's data receiving power, P_b	0.01 W
HMD's power of local viewport rendering, ξ	0.8 W [5]
HMD's computation capability, z_l	0.5 G cycles/s [5]
Data volume HMD's 1 CPU cycle process, b_{zl}	0.05 Kbit/cycle
MECS's computation capability, z_f	1000 G cycles/s [5]
Data volume MECS's 1 CPU cycle process, b_{zf}	10Kbit/cycle
BER on the uth subwindow, ϵ_u	10^{-4}
Transmit/receive antenna gain, G_t, G_r	0-20 dBi [10]
Gaussian noise power spectral density, n_0	-174 dBm/Hz
The refractive index, n_t	1.2-2.8 [10]
Rough surface height standard deviation parameter, σ	0.05-0.15 [10]
Medium absorption coefficient, $k(f_u)$	0.0016/m [3]
Distance, d	1-20 m [9]
Local punishment factors, ω_l	$5 * 10^{-5}$ J/bit
MECS punishment factors, ω_f	$1 * 10^{-5}$ J/bit
Learning rate of actor, l_a	0.01
Learning rate of critic, l_c	0.01
Discount factor, γ	0.9
Bitrate each original full-resolution video chunk, b	3 Gbps
MECS's decoding speed, v^{de}	300 Gbps
The idle power of HMD, P_{id}	0.0001 W

the stochastic policy using a Softmax function. A total of six workers are trained concurrently and optimize their individual weights using gradient descent. Each worker calculates its own successive gradients during each episode. At the end of each episode, each worker updates the global network and then collects the new state of the global weights. The loss functions for the actor and the critic employ that defined in (18) and (20), respectively. The parameters θ and θ_v are optimized using gradients defined in (19) and (21), respectively.

4) Reward Function: r_t is the immediate reward, which is defined as

$$r_t = \frac{1}{(1-\eta(t))[E_l(t) + \omega_l H_l(t)] + \eta(t)\big[E_f(t) + \omega_f H_f(t)\big]}. \tag{24}$$

5) Policy: The policy of the formulated MDP is a mapping $\pi(a_t|s_t;\theta) : \mathcal{S} \rightarrow \mathcal{A}$.

Based on the defined system states, the actions, the reward function, the policy, and the update equations in (19) and (21), the proposed A3C-based viewport rendering offloading decision optimization and downlink transmit power control algorithm is detailed in Algorithm 1 [27], [28], [35], [42].

D. Implementation of Algorithm 1

As was mentioned in Section III-A, there is a decision maker in MECS to make optimization decisions, and the core of the decision maker is actually our A3C-based optimization algorithm. In Algorithm 1, there is a central brain and some agents. Both the central brain and each agent are composed by an actor and a critic. In each time slot t, each agent first synchronous its parameters by global parameters of central brain as in line 8, and then interacts with the environment simultaneously and independently, by choosing actions, i.e., the viewport rendering offloading decision and transmit power control, under its current policy. When actions are taken, each agent obtains a reward and the system transforms to the next state. The process repeats until a final state is reached, as in lines 10–14. Then each agent updates the parameters of its actor and critic networks as in lines 16–20, and sends its corresponding updated parameters to the actor and the critic of central brain asynchronously as in line 21. With time elapses, the above process repeats until the algorithm converges and the final time slot reaches, and then the optimal policy [25], [27] that can maximize the long-term expected cumulative reward can be obtained.

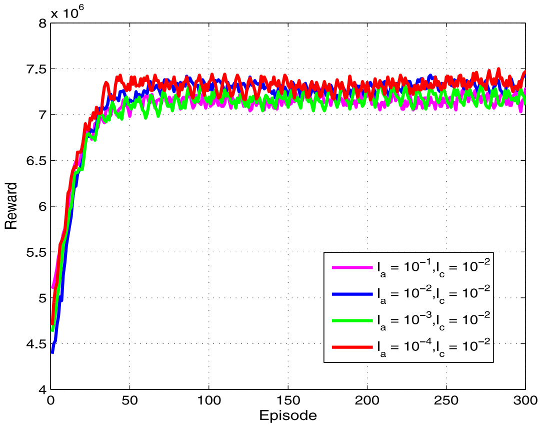


Fig. 3. System reward under different learning rate of the actor network.

VI. SIMULATION RESULTS AND DISCUSSIONS

In this section, we provide simulations to verify and discuss the performance of our proposed joint optimization algorithm. We consider an indoor system where MECS locates in the center, serving an area with the radius being 20 m. The usable bandwidth and number of subwindows refers to the Fig. 1(b) in [10], and it can be know that when the distance rises from 1 m to 20 m, the total bandwidth shrinks from 0.94 THz to 0.78 THz nearly linearly [10]. Based on this observation, we can obtain the decreasing rate of the total usable bandwidth is approximately 8.42 GHz/m. The relationship between the two can be approximately as $B \approx 984.42 - 8.42d$ GHz, so $U^d(t) = \lfloor (984.42 - 8.42d)/B_g \rfloor, t \in \mathcal{T}$. We consider the IBI leakage of 17.47% to the neighboring subwindows in (6), for the rectangular waveform [9]. Regarding the content, we use 4K immersive VR video clips from MPEG [43]. All the videos are in 3840 × 1920 resolution at 30 frame-per-second (fps), with a bit depth 8 b. Each chunk contains 4 frames. The size of viewport is 856×856. Detailed default parameters are summarized in Table I, they will keep unchanged unless otherwise specified.

A. Convergence of Algorithm 1

We first illustrate the convergence of our proposed algorithm under different learning rates. Fig. 3 shows the convergence under different actor's learning rate, with the critic's learning rate set as the default value $l_c = 10^{-2}$, and Fig. 4 shows the convergence under different critic's learning rate, while the actor's learning rate takes the default value $l_a = 10^{-2}$. As can

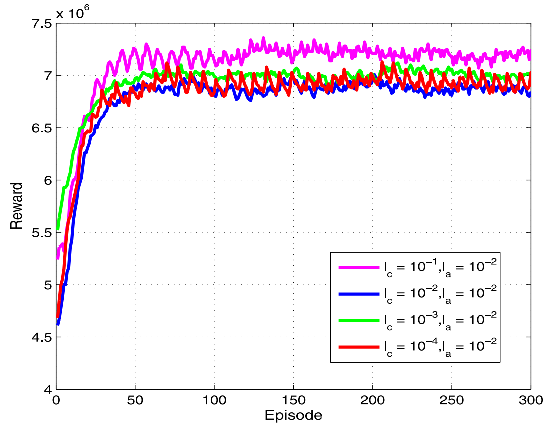


Fig. 4. System reward under different learning rate of the critic network.

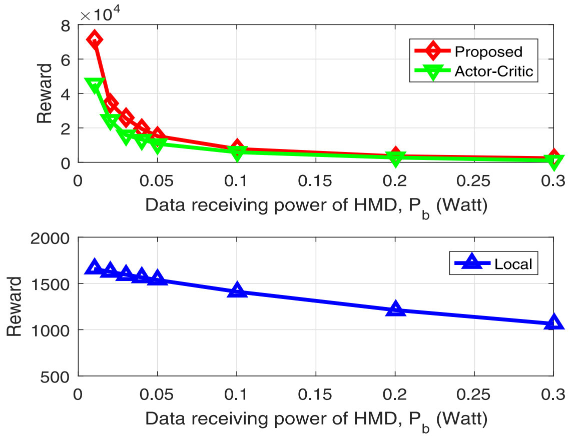


Fig. 5. System reward versus HMD's data receiving power consumption.

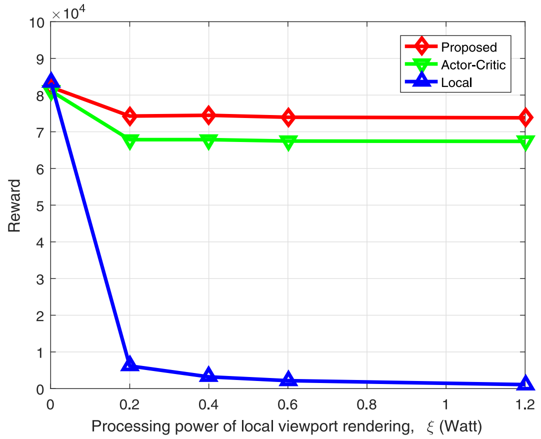


Fig. 6. System reward versus the local viewport rendering power of HMD.

be seen from the two figures, the system reward first increase sharply, and converges at nearly the 50th episode under different learning rate combinations, demonstrating our proposed algorithm converges considerably fast.

B. Performance Evaluation of Algorithm 1

Next, we evaluate the performance of our proposed algorithm by comparing it with the following two algorithms.

1) Local rendering, which is shorted as "Local" in the following context. In Local, there is no rendering offloading decision optimization, so the MECS will first decode all the original data into uncompressed immersive VR video chunks, including viewport tiles and standby tiles, and then deliver them over the THz link to HMD, and then the HMD will render the viewport by itself.
2) Actor–Critic-based algorithm, which is denoted as "Actor–Critic." The only difference between this method and our proposed algorithm is based on A3C, and "Actor–Critic" is based on AC.

Remark 3: As was mentioned, the reward is defined as the reciprocal of our objective function, i.e., the energy consumption of the HMD. So, in the following, we can consider either the system reward or the HMD's energy consumption as our performance metrics when we evaluate the performance of the algorithms. For a certain algorithm, the larger system reward, or the smaller the energy consumption of the HMD the algorithm could achieve, the better performance the algorithm could obtain.

In Fig. 5, we show how the system reward changes under HMD's different basic circuit power consumption P_b. As can be seen, the system reward decrease with P_b increase, which is the same for the four algorithms. This is easy to be understood, when P_b increase, the data receiving energy consumption will increase, so the reward will decrease. Moreover, it can be observed that our proposed A3C-based algorithm performs the optimum, followed by the AC method. Since AC also employ THz as the wireless channel, the only difference between it from our proposed algorithm lies in the DRL methods they adopt. Moreover, as a result of multiagent concurrent training, A3C performs better than AC which is adopted by AC, so AC performs worse than our proposed A3C-based algorithm. For Local method, since energy-demanding viewport rendering is always performed locally, and large quality of energy is consumed, it performs the worst among all the algorithms.

Fig. 6 plots the system reward versus HMD's local task processing power ξ.

1) When $\xi = 0.001$, i.e., the local processing power is very small, all the methods will choose local viewport rendering, i.e., the uncompressed original data will be transmitted to HMD, and the viewport data will be rendered by HMD in all the four methods. Thus, the energy consumed in viewport rendering is the same for all the four algorithms. Since the three algorithms all adopt the larger capacity THz link in wireless data transmission, so the energy consumed in data receiving is all the same for the three algorithms. Therefore, when $\xi = 0.001$, the reward of our proposed A3C-based algorithm, AC and local method nearly perform all the same.
2) When local task processing power ξ increases from 0.001 to 0.2, the energy consumed in local viewport rendering increases quickly, making the energy consumption of all the algorithms increase significantly. Therefore, the system reward decrease quickly, wherein Local decrease the sharpest, this is because all viewport data will be rendered by HMD in Local method, while in other algorithms, the viewport is not always rendered

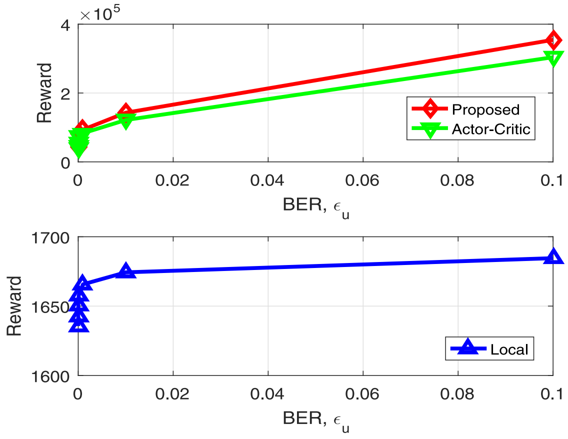


Fig. 7. System reward versus different BER requirement.

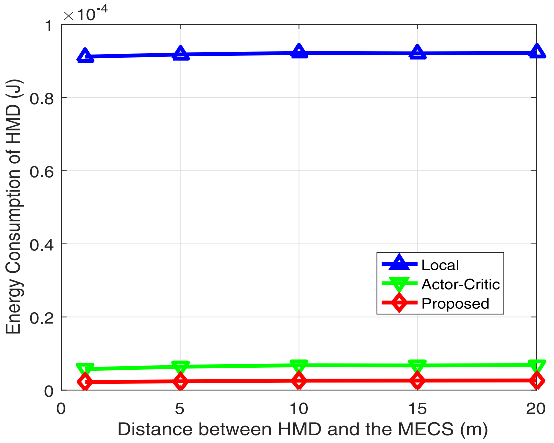


Fig. 8. HMD's energy consumption versus distance between HMD and MECS.

in HMD, and therefore, the decrease in reward is not so sharp.

3) When ξ continues to increase from 0.2 to 1.2, MECS rendering becomes more suitable. Since the Local method always chooses local viewport rendering even if this is not so appropriate, the reward of the Local method keeps droping gradually.

The other three algorithms will choose MECS rendering by offloading decision optimization, so their reward nearly keeps unchanged. From Fig. 6, we also can find that our proposed A3C-based algorithm always performs the optimum.

In Fig. 7, we plot the effect of BER ϵ_u on the system reward, where BER takes its values from 10^{-7}, $10^{-6}, \ldots, 10^{-1}$, respectively. As the required BER increases, the wireless rate increase, leading to a decrease in energy consumption, and consequently the reward will increase. It can be also known that our proposed algorithm performs the best, followed by AC and local, respectively.

Fig. 8 plots how the distance between HMD and MECS affects the energy consumption of the four algorithms. First, we can find that our A3C-based algorithm consumes the least energy, followed by AC and local method. Moreover, it can also find that the distance nearly has no effect on the energy consumption for each method, since their energy consumption nearly keeps unchanged with distance grows. This is because, in our system model, we consider the simple one user scenario, so the wireless resource will be exclusively used by this HMD user, and the wireless resource is sufficiently abundant in all the methods, no matter how far the user is. By the way, the general case with multiple users will be one of our future work, and then we will also show the effect of distance on the energy consumption of the multiple HMD users.

VII. Conclusion

In this article, we have proposed a THz access-based MEC system to support wireless immersive VR video services. We have formulated a joint viewport rendering and THz downlink transmit power control problem to investigate the HMD's long-term energy consumption minimization. Based on the A3C DRL algorithm, we developed a low-complexity algorithm to obtain the optimal solution to viewport rendering offloading decision making and transmit power control. The simulation results have verified the convergence of our algorithm, and have demonstrated that our algorithm could perform much better than other algorithms in energy consumption minimization or system reward maximization.

References

[1] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360° video viewing dataset in head-mounted virtual reality," in *Proc. 8th ACM Multimedia Syst. Conf.*, Taipei, China, Jun. 2017, pp. 211–216.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[3] C. Chaccour, R. Amer, B. Zhou, and W. Saad, "On the reliability of wireless virtual reality at terahertz (THz) frequencies," in *Proc. IEEE 10th IFIP Int. Conf. New Technol. Mobility Security (NTMS)*, Canary Islands, Spain, Jun. 2019, pp. 1–5.

[4] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 177–191, Jan. 2020.

[5] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "MEC-assisted panoramic VR video streaming over millimeter wave mobile networks," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1302–1316, May 2019.

[6] F. Rahim, M. P. Queluz, and J. Ascenso, "Objective assessment of line distortions in viewport rendering of 360° images," in *Proc. IEEE Int. Conf. Artif. Intell. Virtual Reality (AIVR)*, Taichung, Taiwan, Dec. 2018, pp. 68–75.

[7] R. Wang, H. Yang, H. Wang, and D. Wu, "Social overlapping community-aware neighbor discovery for D2D communications," *IEEE Wireless Comm.*, vol. 23, no. 4, pp. 28–34, Aug. 2016.

[8] M. S. Mahmud, H. Wang, A. M. Esfar-E-Alam, and H. Fang, "A wireless health monitoring system using mobile phone accessories," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2009–2018, Dec. 2017.

[9] C. Han and I. F. Akyildiz, "Distance-aware bandwidth-adaptive resource allocation for wireless systems in the Terahertz band," *IEEE Trans. THz Sci. Technol.*, vol. 6, no. 4, pp. 541–553, Jul. 2016.

[10] C. Han, A. O. Bicen, and I. F. Akyildiz, "Multi-ray channel modeling and wideband characterization for wireless communications in the Terahertz band," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2402–2412, May 2015.

[11] H. Cao, S. Wu, G. S. Aujla, Q. Wang, L. Yang, and H. Zhu, "Dynamic embedding and quality of service-driven adjustment for cloud networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1406–1416, Feb. 2020.

[12] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw. Mag.*, vol. 33, no. 5, pp. 156–165, Sep./Oct. 2019.

[13] J. Feng, F. R. Yu, and E. A. Q. Pei, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4321–4334, Jun. 2020.

[14] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, to be published.

[15] J. Du, X. Chu, F. R. Yu, J. Feng, and C.-L. I, "Enabling low-latency applications in LTE-A based mixed fog/cloud computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1757–1771, Feb. 2019.

[16] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1079–1092, Feb. 2019.

[17] H. Cao, Y. Zhu, G. Zheng, and L. Yang, "A novel optimal mapping algorithm with less computational complexity for virtual network embedding," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 356–371, Mar. 2018.

[18] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[19] M. Chen, U. Challita, W. Saad, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[20] Y. Sun *et al.*, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019.

[21] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[22] J. W. L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, early access, Mar. 4, 2019, doi: 10.1109/TETC.2019.2902661.

[23] X. Qiu, L. Liu, W. C. Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8050–8062, Aug. 2019.

[24] N. Chen *et al.*, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[25] M. Chen, W. Saad, C. Yin, and M. Debbah, "Data correlation-aware resource management in wireless virtual reality (VR): An echo state transfer learning approach," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4267–4280, Jun. 2019.

[26] M. Chen, W. Saad, and C. Yin, "Liquid state based transfer learning for 360° image transmission in wireless VR networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[27] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 1928–1937.

[28] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a GPU," 2019. [Online]. Available: arXiv:1611.06256.

[29] H. Wang, D. Peng, W. Wang, H. Sharif, H. Chen, and A. Khoynezhad, "Resource-aware secure ECG healthcare monitoring through body sensor networks," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 12–19, Feb. 2010.

[30] H. Wang, D. Peng, W. Wang, H. Sharif, and H. Chen, "Cross-layer routing optimization in multirate wireless sensor networks for distributed source coding based applications," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3999–4009, Oct. 2008.

[31] W. Wang, D. Peng, H. Wang, H. Sharif, and H. Chen, "Cross-layer multirate interaction with distributed source coding in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 2, pp. 787–795, Feb. 2009.

[32] J. M. Jornet and I. F. Akyildiz, "Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the Terahertz band," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3211–3221, Oct. 2011.

[33] S. T. Chung and A. J. Goldsmith, "Degrees of freedom in adaptive modulation: A unified view," *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1561–1571, Sep. 2001.

[34] Z. Zhang, F. R. Yu, F. Fu, Q. Yan, and Z. Wang, "Joint offloading and resource allocation in mobile edge computing systems: An actor-critic approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.

[35] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[36] Y. Bengio, Y. LeCun, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[37] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[38] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L. C. Wang, "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 44–52, Jun. 2019.

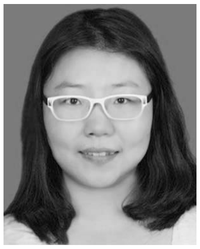
[39] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[40] H. Wang, M. Hempel, D. Peng, W. Wang, H. Sharif, and H. Chen, "Index-based selective audio encryption for wireless multimedia sensor networks," *IEEE Trans. Multimedia*, vol. 12, no. 3, pp. 215–223, Apr. 2010.

[41] T. Tieleman and G. Hinton, "Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude," *COURSERA Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[42] J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile edge computing: A deep reinforcement learning approach," *IEEE Internet Things J.*, early access, Dec. 24, 2019, doi: 10.1109/JIOT.2019.2961707.

[43] E. Alshina, J. Boyce, A. Abbas, and Y. Ye, "JVET common test conditions and evaluation procedures for 360° video," document JVET–D1030, Joint Video Exploration Team of ITU-T SG, Macao, China, Oct. 2017,



Jianbo Du (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2007 and 2013, respectively, and the Ph.D. degree in communication and information systems from Xidian University, Xi'an, in 2018.

She is currently a Lecturer with the School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications. Her research interests include mobile-edge computing, resource management, NOMA, artificial intelligence, and their applications in wireless communications.

F. Richard Yu (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2003.

From 2002 to 2006, he was with Ericsson, Lund, Sweden, and a start-up in California, USA. He joined Carleton University, Ottawa, ON, Canada, in 2007, where he is currently a Professor. His research interests include wireless cyber–physical systems, connected/autonomous vehicles, security, distributed ledger technology, and deep learning.

Prof. Yu received the IEEE Outstanding Service Award in 2016, the IEEE Outstanding Leadership Award in 2013, the Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly, Premiers Research Excellence Award) in 2011, the Excellent Contribution Award at IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from Canada Foundation of Innovation in 2009, and the Best Paper Awards at IEEE ICNC 2018, VTC 2017 Spring, ICC 2014, Globecom 2012, IEEE/IFIP TrustCom 2009, and Int'l Conference on Networking 2005. He serves on the editorial boards of several journals, including the Co-Editor-in-Chief for *Ad Hoc & Sensor Wireless Networks* and a Lead Series Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has served as the Technical Program Committee Co-Chair of numerous conferences. He is a registered Professional Engineer in the province of Ontario, Canada. He is a Distinguished Lecturer, the Vice President (Membership), and an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society. He is a Fellow of the Institution of Engineering and Technology.

Guangyue Lu received the Ph.D. degree from Xidian University, Xi'an, China, in 1999.

From September 2004 to August 2006, he was a Guest Researcher with the Signal and Systems Group, Uppsala University, Uppsala, Sweden. Since 2005, he has been a Professor with the Department of Telecommunications Engineering, Xi'an Institute of Posts and Telecommunications, Xi'an. His current research area is in signal processing in communication systems, cognitive radio, and spectrum sensing.

Prof. Lu received the Award from the Program for New Century Excellent Talents in University, Ministry of Education, China, in 2009.

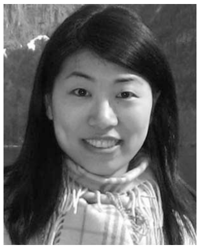
Junxuan Wang received the B.S. degree from Northwestern Polytechnical University, Xi'an, China, in 1994, the M.E. degree from the Xi'an University of Science and Technology, Xi'an, in 2002, and the Ph.D. degree from the Beijing University of Post and Telecommunications, Beijing, China, in 2005.

He is currently a Full Professor with the School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an. His research interests include the areas of 5G networks and wireless communications.

Jing Jiang (Member, IEEE) received the M.Sc. degree from Xidian University, Xi'an, China, in 2005, and the Ph.D. degree in information and communication engineering from North Western Polytechnic University, Xi'an, in 2009.

She was a Researcher and a Project Manager with ZTE Corporation, Shenzhen, China, from 2006 to 2012. She is currently a Professor with the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an. Her research interests include massive multiple-input–multiple-output systems and millimeter-wave communications.

Prof. Jiang has been a Member of 3GPP.



Xiaoli Chu (Senior Member, IEEE) received the B.Eng. degree in electronic and information engineering from Xi'an Jiaotong University, Xi'an, China, in 2001, and the Ph.D. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2005.

She is a Senior Lecturer with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K. From September 2005 to April 2012, she was with the Centre for Telecommunications Research, King's College London, London, U.K. She has published more than 100 peer-reviewed journal and conference papers. She has Lead Editored/authored the book *Heterogeneous Cellular Networks: Theory, Simulation and Deployment* (Cambridge University Press, 2013) and *4G Femtocells: Resource Allocation and Interference Management* (Springer, 2013).