IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 31, 2018, accepted March 13, 2018, date of publication March 19, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817288

Communication-Constrained Mobile Edge Computing Systems for Wireless Virtual Reality: Scheduling and Tradeoff

XIAO YANG[1], ZHIYONG CHEN[1,2], KUIKUI LI[1], YAPING SUN[1], NING LIU[1], WEILIANG XIE[3], AND YONG ZHAO[3]

[1]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China
[2]Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai 200240, China
[3]China Telecom Corporation Limited Technology Innovation Center, Beijing 100031, China

Corresponding author: Zhiyong Chen (zhiyongchen@sjtu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61671291, Grant 61671301, and Grant 61420106008.

ABSTRACT Mobile edge computing (MEC) is expected to be an effective solution to deliver virtual reality (VR) videos over wireless networks. In contrast to previous computation-constrained MEC, which reduces the computation-resource consumption at the mobile device by increasing the communication-resource consumption, we develop a communications-constrained MEC framework to reduce communication-resource consumption by fully exploiting the computation and caching resources at the mobile VR device in this paper. Specifically, according to a task modularization, the MEC server only delivers the components which have not been stored in the VR device, and then the VR device uses the received components and other cached components to construct the task, yielding low communication cost but high delay. The MEC server also computes the task by itself to reduce the delay, however, it consumes more communication-resource due to the delivery of entire task. Therefore, we propose a task scheduling strategy to decide which computation model should the MEC server operates to minimize the communication-resource consumption under the delay constraint. Finally, the tradeoffs among communications, computing, and caching are also discussed, and we analytically find that given a target communication-resource consumption, the transmission rate is inversely proportional to the computing ability of mobile VR device.

INDEX TERMS Mobile edge computing, virtual reality, communications-computing-caching tradeoffs.

I. INTRODUCTION

Virtual reality (VR) application over wireless networks is gaining an unprecedented attention due to the ability to bringing an immersive 360 viewing experience to users. A new report forecasts that the data consumption from wireless VR headsets (smartphone-based and standalone) will grow by over 650% over the next 4 years (2017-2021) [1]. The VR application is computational-intensive, capacity-intensive and delay-sensitive, bringing the fact that most of VR devices are now wired with cables. Taking 6 degree-of-freedom (DoF) VR video for an example, the required transmission rate is from 200 *Mbps* to 1 *Gps* per user of less than 20 *ms* end-to-end latency [2]. Meanwhile, different from regular 4K/8K video, 6-DoF VR video also requires heavy computation to stitch footage from multiple regular cameras [3]. Current wireless systems (e.g., LTE) cannot cope with the ultra-low latency and ultra-high throughput requirements of wireless VR application (e.g., VR video/game) [3]. Due to the popularity of the VR application, *how to* deliver the VR video in wireless networks becomes one of the main challenges for future 5G or beyond networks.

There is no effective way to address this challenge so far, but one generally accepted and promising solution is the collective usage of three primary resources (*communications, computing and caching*) in the wireless network [3]–[5], e.g., mobile edge computing (MEC) [4]. The MEC technology moves computation abilities from the cloud computing center to the edge of wireless radio access network (RAN), e.g., base stations (BS). By deploying computation resource at the network edge, MEC performs the computation tasks closer to the VR device, reducing the latency and improving the quality of service.

Most existing studies of MEC focus on **Computation-Constrained MEC**, which migrating the computation tasks from mobile VR device to the MEC server due to the limited computation capability of mobile VR device [6]–[8]. In computation-constrained MEC, mobile devices should be able to upload the task to the MEC server, and then the MEC server executes the task and delivers the computation results to mobile devices. This approach results in one drawback: *increasing the communication-resource consumption*, although in a reduced computation-resource consumption of mobile devices. Therefore, it is quite suitable for the computational-intensive and delay-sensitive application with low bandwidth consumption, e.g., some simple augmented reality (AR) games [9].

In fact, videos can be modularized today [10], e.g., MPEG Media Transport (MMT) standard, as shown in Fig. 1. The required video can be partitioned into chunks, and rearranged at the mobile device. The video modularization enables the request video to be organized and delivered in different ways based on chunk popularity and the user request. This technology results in two benefits: *i*) *storing* most popular chunks *ii*) *eliminating* the redundancy chunk in a video. In a word, the content popularity and modularization provide a higher potential to save the network bandwidth by combining with the *caching* resource at the mobile VR device [11].

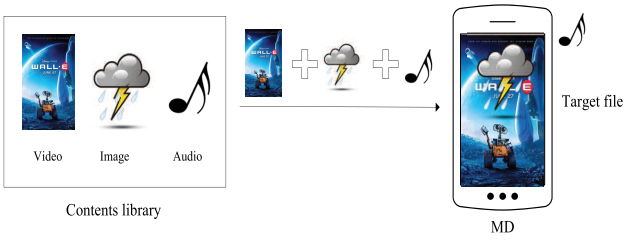

FIGURE 1. **The tasks can be modularized [10].**

Although the computation-constrained MEC solution may solve the challenge partially, the growth rate of mobile VR data has far exceeded the capacity increase of wireless network [4]. By 2021, VR will require more data demands than required for 4K, which requires fast data speeds to stream content effectively [1]. We can also use the MEC architecture to improve network responsiveness and reduce latency, however, we should not consume extra communication resource to reduce the cost of computation resource. In contrast, we should try to *save the communication resource by taking advantage of the computation and caching resources* at the mobile VR devices, where we propose this solution and name it as **Communication-Constrained MEC** in this paper.

In this paper, we present a communications-constrained MEC framework to exploiting the computing and caching resources in MEC-enabled wireless networks aiming to deliver the VR video effectively. Our goal is to minimize the average transmission data per task under the delay constraint in this system and find out the tradeoffs among the wireless transmission rate, the computing ability and the cache size at the mobile VR device. Here, we use the average transmission data per task as the communication-resource consumption. Our major contributions are summarized as follows:

- *We propose a communications-constrained MEC framework to reduce the communication-resource consumption by exploiting the caching resource and increasing the consumption of computing resource at the mobile VR device*. When the mobile VR device submits a task request, the MEC server can only deliver the corresponding components which have not been stored in the mobile VR device, and then the VR device uses the received components and the corresponding cached components to construct the task by exploiting the local computation resource.
- *We develop an optimal task scheduling policy to minimize the average transmission data per task*. Of course, the MEC server can also select MEC computation model, which the MEC server computes all corresponding chunks as the target task and then deliver the entire task to the mobile VR device. The MEC computation mode is a reliable way to reduce the latency due to the fast computation capability at the MEC server, but this model delivers more data per task to the user. Therefore, we formulate the transmission data consumption minimization problem under the delay constraint and propose a task scheduling strategy by leveraging the Lyapunov theory. In each time slot, the scheduling is determined by solving a linear and discrete programming problem.
- *We discuss the tradeoffs between communications, computing, and caching (3C)*. We derive the closed-form expression about the average transmission data per task, the CPU frequency and the caching size. Our analysis reveals that the minimum of average transmission data per task $\bar{D}^{opt}$ decreases with the computing ability or the caching size of the mobile VR device under certain condition. Besides, we also derive the upper bound of the end-to-end latency, and then present how to joint allocate communications, computing and caching resources in the proposed communications-constrained MEC system to achieve a target $\bar{D}^{opt}$.
- *We conduct extensive simulation results to verify the theoretical analysis results and evaluate the performance of the proposed framework.* Simulation results show that the proposed scheduling strategy achieve a significant saving in the average transmission data per task. The impacts of different system parameters, e.g., the caching size/the arrival rate, on the average transmission data per task and the end-to-end latency are also investigated.

II. RELATED WORKS

The investigation of computation-constrained MEC has attracted significant attention recently, in terms of task scheduling policy [6], [7], [12]–[17] and resource allocation [8], [18]–[20]. Task scheduling policy plays an important

role in MEC, where it determines a task to be executed in the mobile device or the MEC server. In [6], a dynamic computation scheduling algorithm based on Lyapunov theory was proposed to minimize the execution delay and task failure. Similarly, [12] analyzed the average delay of each task and the average power consumption at the mobile device, and then proposed a stochastic computation task policy in the computation-constrained MEC system, in order to minimize the average delay subject to the average power constraint at the mobile device. Multi-edge device scenarios were taken into consideration in [7], and a semidefinite relaxation (SDR)-based algorithms was proposed to optimize both offloading decisions and the computation allocation of mobile device, which minimizes the execution delay and the use's energy cost. Besides, the channel side information was considered in [12], the execution delay minimization problem was investigated using Markov chain modeling. The multi-user and multi-channel wireless interference environment was further investigated in [14] and [16]. Reference [15] considered the specific application scenario. Reference [15] investigated the MIMO scenario, the radio resources and the computational resources were optimized by iterative algorithm based on novel successive convex approximation technique, in order to minimize users' energy cost.

The authors of [8] and [18]–[21] focused on the resource allocation in the MEC system. In [8], users were divided into different priorities, which based on user's channel gain and local computation ability, then cloud computation resource and radio resource are allocated to each user to reduce their energy cost. Reference [18] further extended this work to both TDMA and FDMA systems. In [20], a user-centric energy-aware mobility management (EMM) scheme was developed to optimize the delay under the long-term energy consumption constraint of the user. Reference [21] jointly optimized communications and computation resources for partial computation offloading using dynamic voltage scaling.

To investigated the caching technology in mobile edge network, many issues be studied in [22] and [23]. For on-demand video streaming in MEC networks, [22] proposed a collaborative joint caching and processing scheme, to minimized the backhaul network traffic under the constraints of cache storage and processing ability. A base station with caching capability was introduced in [23] for computation-constrained MEC systems, and a joint caching and offloading scheme was proposed to minimize the average energy cost subject to the caching and deadline constraints. But we should note that the caching resource in the mobile device is also getting cheaper. Therefore, the caching resource of the mobile device is another good choice since the cached contents can be directly used for local computation. The design principles for our cache-enabled MEC systems are different from those for edge caching systems.

The rest of this paper is organized as follows. In Section III, we present the system model. The system state analysis, scheduling strategy and the transmitting data minimization problem are presented in Section IV. In Section V, we propose the offloading decision optimization algorithm based on Lyapunov theory. Then the tradeoffs of communications, computing and caching are shown in Section VI. The numerical and simulation results are presented in Section VII, and the conclusions are drawn in Section VIII.

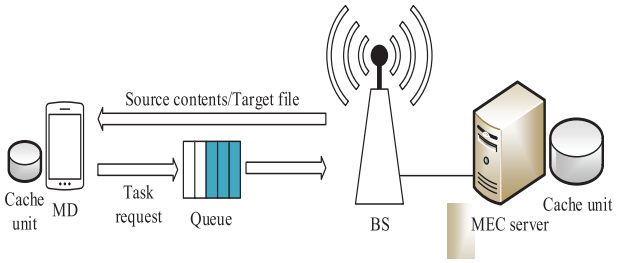

FIGURE 2. A MEC system with a caching enabled mobile device.

III. SYSTEM MODEL

As shown in Fig.2, we consider an MEC system, where a cache-enabled mobile VR device can access BS with an MEC server to obtain task. The MEC server has an abundance of computing and caching resource, while the mobile VR device has limited computing ability and cache capacity.

A. TASK MODEL

We consider each task consists of a number of chunks, e.g., MMT assets. All the chunks composing each task come from a set of N possible chunks, which is denoted by $\mathbb{F} = \{F_1, F_2 \ldots\ldots F_N\}$. Note that one chunk may be used more than once in a task. The popularity distribution of the chunks is denoted by $\mathbf{p} = [p_1, \ldots, p_N]$, where $\sum_{i=1}^{N} p_i = 1$. We assume all the chunks are of equal size τ and the MEC server has all N chunks. The cache capacity of mobile VR device is M with $M < N$, which can store at most M chunks. We adopt the most popular caching strategy in this paper, and the stored chunks set can be denoted as $\mathbb{M} = \{F_1, F_2 \ldots\ldots F_M\}$.

The system is time-slotted with the time slot length Δ. Let H_t be the task scheduled at time slot t, which consists of K_t contents. We denote $H(t) = [h_1(t), \ldots, h_{K_t}(t)]$ as the content index vector of the task H_t, where $h_{k_t}(t) \in \{1, \ldots, N\}$ indicates that the k_t-th content in H_t is $F_{h_{k_t}(t)}$. Thus the size of H_t is $D(t) = \tau K_t$. Let $G_n^t (1 \le n \le N)$ denote whether $F_n \in \mathbb{F}$ is requested in H_t and not cached in mobile VR device, which can be given by

$$G_n^t = \begin{cases} 1, & \text{for } M+1 \le n \le N, \text{ and } h_{k_t}(t) = n \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

B. COMPUTATION MODEL

When the mobile VR device requests a task, the MEC server first decides whether the desired task can be computed at the MEC server or not. If it does not execute the task, the requested task or the corresponding chunks should be delivered and executed at the mobile VR device. Thus,

we discuss two computation modes in this paper: MEC computation mode and local computation mode. Let W denote the required CPU cycles for computing one bit. The CPU frequency of the MEC server and the mobile VR device is f_c and f_l, respectively. In general, the MEC server has more powerful computation ability than the mobile VR device, i.e., $f_c > f_l$. The wireless transmission rate is R (in bits per second).

1) MEC COMPUTATION MODE

In this mode, when the mobile VR device submits a task request, the MEC server can execute a computation operation to combine the corresponding chunks as the target task, and then deliver the task to the mobile VR device. The size of the task input is $D_{cc}(t) = D(t)$ because the MEC server executes all the corresponding chunks. We assume $D_{ct}(t) = \phi D(t)(\phi \geq 1)$ is the size of task output that transmitted form MEC server to mobile VR device. Therefore, the total $N_c(t) = \lceil D_{cc}(t)W/(f_c\Delta) + D_{ct}(t)/R\Delta \rceil$ time slots are required to satisfy this task request. Similar to [12], we use $S_c(t) \in \{0, 1, \ldots, N_c(t)-1\}$ to denote the state of the MEC server. $S_c(t) = 0$ means the MEC server is idle, while $S_c(t) = n$ $(n \neq 0)$ indicates a task is processing at the MEC server and $N_c(t) - n$ time slots are required to complete the computation and delivery the task.

2) LOCAL COMPUTATION MODE

When the MEC server runs this mode, the MEC server does not execute the task but delivers the components of the task to the mobile VR device. If one component is stored at the mobile VR device, the MEC server does not deliver it to the mobile VR device. If one component is not stored on the mobile VR device, the MEC server transmits it to the mobile VR device. Besides, the MEC server also eliminates the redundancy among the chunks contained in the task. For instance, a chunk file F_{M+1} is used to one task $H(t)$ three times, but the MEC server only needs to deliver F_{M+1} once. As a result, the size of transmission data is $D_{lt}(t) = \tau \sum_{n=M+1}^{N} G_n^t$, and it is easy to see $D_{lt}(t) \leq D_{ct}(t)$, saving the network traffic and bandwidth.

The mobile device can rearrange the target task by combing the received chunks with the corresponding cached chunks. Because all the corresponding chunks are executed in the mobile VR device, the size of task input is $D_{lc}(t) = D(t)$. The total $N_l(t) = \lceil D_{lc}(t)W/(f_l\Delta) + D_{lt}(t)/R\Delta \rceil$ time slots are required to complete H_t in this mode. Similar to the definition of $S_c(t)$, let $S_l(t) \in \{0, 1, \ldots, N_l(t)-1\}$ denote the mobile VR device state. If the system allocates a task to the mobile device at time slot t, $S_l(t)$ updates to $S_l(t+1) = N_l(t) - 1$.

C. TASK QUEUEING MODEL

The task request arrival process is modeled as a Bernoulli process with probability λ. When a task request arrivals, the request first enters into a task queue with infinite capacity. Let us define the queue state $Q(t) = \{0, 1, 2, 3 \ldots\ldots\}$ as the number of the request waiting in the queue, where $Q(t)$ updates according to the following equation

$$Q(t+1) = (Q(t) - (u_l^1(t) + u_l^2(t) + u_c^1(t) + u_c^2(t)))^+ + A(t), \tag{2}$$

where $A(t)$ denotes whether a task request arriving in the time slot t or not. Thus we have $\Pr\{A(t) = 1\} = \lambda$ and $\Pr\{A(t) = 0\} = 1 - \lambda$. Here, $\{u_l^1(t), u_l^2(t), u_c^1(t), u_c^2(t)\}$ denotes the task scheduling decision at the time slot t.

Note that at most two task requests can be scheduled at a time slot. The first task request should be scheduled before the second task request. If the first task request is scheduled to do the computation on the mobile VR device (MEC server), we have $u_l^1(t) = 1$ $(u_c^1(t) = 1)$. For this scenarios, the second task request can not be scheduled to operate in the local computation mode (MEC computation mode) because the CPU has been occupied by the first task, yielding $u_l^2 = 0$ $(u_c^2(t) = 0)$. Otherwise, we have $u_l^1(t) = 0$ $(u_c^1(t) = 0)$ for the first task, and the local computation mode (MEC computation mode) could be scheduled for the second task, i.e., $u_l^2 = 1$ $(u_c^2(t) = 1)$. As a result, there are five possible states for the task scheduling decision in each time slot, i.e., $\{u_l^1(t), u_l^2(t), u_c^1(t), u_c^2(t)\} = \{(0,0,0,0), (1,0,0,0), (0,0,1,0), (1,0,0,1), (0,1,1,0)\}$.

IV. TASK SCHEDULING STRATEGY AND PROBLEM FORMULATION

The MEC server is a reliable way to reduce the computation latency due to $f_c \geq f_l$, but consumes more communication resource due to $D_{ct}(t) \geq D_{lt}(t)$. In this paper, our performance metric of interest is the average transmission data per task. Hence, the MEC server needs to make the task scheduling decision at each time slot to minimize the average transmission data per task under the average delay constraint.

A. TASK SCHEDULING STRATEGY

When the MEC server (mobile VR device) is idle, the task can be scheduled to the MEC computation mode (local computation mode). The queue state $Q(t) = 0$ denotes the task queue is empty and there is no task will be scheduled in time slot $t+1$, while $Q(t) = \infty$ indicates that there are infinite tasks in the task queue, yielding the unstable system. According to $Q(t)$, $S_l(t)$, and $S_c(t)$, we can describe the system state.

Case 1: $S_l(t) = S_c(t) = 0$. Both the mobile VR device and the MEC server are idle. The system can process at most two tasks. If there are two tasks in $Q(t)$ at least, i.e., $Q(t) \geq 2$, one task can be processed in the mobile VR device (MEC server) and the other task remains wait in the task queue or to be processed in the MEC server (mobile VR device). The task scheduling policy can be expressed as the following:

$$u^1(t) = \begin{cases} (0,0,0,0) \\ (0,0,1,0) \\ (1,0,0,0) \\ (1,0,0,1) \\ (0,1,1,0) \end{cases} \quad \text{for } Q(t) \geq 2. \tag{3}$$

If there is only one task in $Q(t)$, the task can be processed in the mobile VR device, the MEC server or remains wait in the task queue. We thus have

$$u^2(t) = \begin{cases} (0,0,0,0) \\ (0,0,1,0) \\ (1,0,0,0) \end{cases} \quad \text{for } Q(t) = 1. \tag{4}$$

Case 2: $S_l(t) \neq 0, S_c(t) = 0$. In this case, the MEC server is idle and the mobile VR device is busy so that the system can process one task at most for the MEC server. The task scheduling policy is:

$$u^3(t) = \begin{cases} (0,0,0,0) \\ (0,0,1,0) \end{cases} \quad \text{for } Q(t) \geq 1. \tag{5}$$

Case 3: $S_l(t) = 0, S_c(t) \neq 0$. In this case, the mobile VR device operates in idle mode and the MEC server is occupied. Only one task can be scheduled for the mobile VR device. The task scheduling policy can be expressed as following:

$$u^4(t) = \begin{cases} (0,0,0,0) \\ (1,0,0,0) \end{cases} \quad \text{for } Q(t) \geq 1. \tag{6}$$

Case 4: $S_c(t) \neq 0, S_l(t) \neq 0$ or $Q(t) = 0$. If both the MEC server and mobile device are busy, i.e., $(S_c(t) \neq 0, S_l(t) \neq 0)$, or there is no task in the task queue $Q(t) = 0$, no task is scheduled. We then have

$$u^5(t) = (0,0,0,0). \tag{7}$$

At the time slot t, $S_l(t)$ and $S_c(t)$ can be expressed as:

$$S_l(t+1) = \begin{cases} \max(S_l(t)-1, 0) & u_l^1(t) = 0 \text{ or } u_l^2(t)=0, \\ N_l^1(t) - 1 & u_l^1(t) = 1, \\ N_l^2(t) - 1 & u_l^2(t) = 1. \end{cases} \tag{8}$$

$$S_c(t+1) = \begin{cases} \max(S_c(t)-1, 0) & u_c^1(t)=0 \text{ or } u_c^2(t)=0, \\ N_c^1(t) - 1 & u_c^1(t) = 1, \\ N_c^2(t) - 1 & u_c^2(t) = 1. \end{cases} \tag{9}$$

where $N_l^i(t)$ and $N_c^i(t)$ denote $N_l(t)$ and $N_c(t)$ of the i-th task for $i = 1, 2$, respectively. $S_l(t+1) = N_l^i(t) - 1$ means the mobile VR device is occupied by a task in the time slot $t+1$ and will be busy in the follow $N_l^i(t) - 1$ time slots. Similarly, we have $S_c(t+1) = N_c^i(t) - 1$.

B. PROBLEM FORMULATION

When $T \to \infty$ and the length of task queue is not infinite, the total number of the task is close to λT. Therefore, the average transmission data per task can be expressed as:

$$\lim_{T\to\infty} \frac{1}{\lambda T} \Big\{ \sum_{t=0}^{T-1} \sum_{i=1}^{2} u_l^i(t) D_{lt}^i(t) + u_c^i(t) D_{ct}^i(t) \Big\}. \tag{10}$$

where $D_{lt}^i(t)$ and $D_{ct}^i(t)$ denote $D_{lt}(t)$ and $D_{ct}(t)$ of the i-th task for $i = 1, 2$, respectively.

From the system model, we know that each task requires transmission time, waiting time and processing time. The computation processing time of the MEC server or the mobile VR device is the dominant influence on the execution delay. Based on the Little Law [24], [25], the execution delay, including the waiting time and processing time, is proportional to the average queue length of the task buffer. The execution delay is written as:

$$\lim_{T\to\infty} \frac{1}{T} \mathbb{E}\Big[\sum_{t=0}^{T-1} Q(t) \Big]. \tag{11}$$

Let us denote the task scheduling policy $\pi(t) \triangleq \{u_l^1(t), u_l^2(t), u_c^1(t), u_c^2(t)\}$. Thus, the communication-resource consumption minimization problem is formulated as:

$$\text{P1}: \min_{\pi(t)} \lim_{T\to\infty} \frac{1}{\lambda T} \Big\{ \sum_{t=1}^{T} \sum_{i=1}^{2} u_l^i(t) D_{lt}^i(t) + u_c^i(t) D_{ct}^i(t) \Big\}$$

$$s.t.\ \pi(t) \in u^k(t), \quad k \in \{1,2,3,4,5\}, \tag{12}$$

$$\lim_{T\to\infty} \frac{1}{T} \mathbb{E}\Big[\sum_{t=1}^{T} Q(t) \Big] < \infty, \tag{13}$$

where (13) indicates the delay constraint to ensure the task requires can be completed with a finite delay. Unfortunately, P1 is a stochastic optimization problem. The system state changes after a offloading decision is made, and P1 is impossible to be solved by convex optimization methods.

V. OPTIMAL TASK SCHEDULING ALGORITHM BASED ON LYAPUNOV THEORY

In this section, we propose an optimal task scheduling algorithm to solve P1 based on Lyapunov theory. To simplify P1, we consider Lyapunov optimization theory. We first define the Lyapunov function:

$$L(Q(t)) = \frac{1}{2} Q^2(t). \tag{14}$$

Consider the initial state $Q(0) = 0$, and then we have $L(Q(0)) = 0$. If the queue is unstable, $L(Q(t))$ is more volatile than $Q(t)$. Thus the expectation of $L(Q(t))$ is:

$$\begin{aligned} \mathbb{E}[L(Q(t))] &= \mathbb{E}\Big\{ \sum_{i=0}^{t-1} [L(Q(i+1)) - L(Q(i))] \Big\} \\ &= \sum_{i=0}^{t-1} \mathbb{E}\{ L(Q(i+1)) - L(Q(i)) | Q(i) \}. \end{aligned} \tag{15}$$

The system is stable when $\mathbb{E}[L(Q(t))] < \infty$. Therefore the Lyapunov drift function can be given by:

$$\Delta L(Q(t)) = \mathbb{E}\Big\{ L(Q(t+1)) - L(Q(t)) | Q(t) \Big\}. \tag{16}$$

We can see from (15) and (16) that to maintain the stability of the queue, we should minimize (16) in each time slot. Therefore the expectation of the $L(Q(t))$ would not tend to infinite. As a result, we have the following Lemma 1.

Lemma 1: Let us define the scheduling rate $U(t) = u_l^1(t) + u_l^2(t) + u_c^1(t) + u_c^2(t)$. In order to ensure

$\mathbb{E}[L(Q(t))] < \infty$, we have:

$$\Delta L(Q(t)) \leq C_{max} + Q(t)\mathbb{E}[A(t) - U(t)|Q(t)], \quad (17)$$

where we use $C_{max} = (\lambda + \frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c} + \frac{2}{\bar{N}_l\bar{N}_c})/2$. And we use $\mathbb{E}[N_c(t)] = \bar{N}_c$ and $\mathbb{E}[N_l(t)] = \bar{N}_l$ to denote the exception of the time slots required to complete a task in the MEC computation model and the local computation model, respectively.

Proof: The proof is provided in Appendix A. ■

According to Lyapunov theory [25], when we make the task scheduling strategy $\pi(t)$ to minimize $\Delta L(Q(t))$, the queue state $Q(t)$ can also approach a lower length. However, the minimization of $\Delta L(Q(t))$ can not lead to the minimization of (10). Thus, we define the Lyapunov drift-plus-penalty function:

$$\begin{aligned}&\Delta L(Q(t)) + V\mathbb{E}[D(t)|Q(t)] \\ &\leq C_{max} + Q(t)\mathbb{E}[A(t) \\ &\quad - U(t)|Q(t)] + V\mathbb{E}[D(t)|Q(t)]. \end{aligned} \quad (18)$$

where V is a non-negative control parameter, which denotes that the system is sensitive to the communication cost. When $V = 0$, the system is only sensitive to the delay. With the increase of V, the Lyapunov drift-plus-penalty becomes more sensitive to the communication cost. Notice that the optimal task scheduling decision $\pi^*(t)$ for minimizing the right side of (18) also minimize $D(t)$ under the queue length stability constraint. Therefore, we can solve P2 in each time slot t:

$$\begin{aligned}\text{P2}: \min_{\pi(t)} \quad & -Q(t)U(t) + VD(t) \\ s.t \quad & (12). \end{aligned} \quad (19)$$

For each time slot t, we can obtain $D(t)$ based on $\pi(t)$ and $Q(t)$. Because there are only five possible choices for $\pi(t)$, we can solve P2 in each time slot t by an enumeration method. Thus, we propose an optimal task scheduling algorithm based on Lyapunov theory, as shown in Algorithm 1.

Algorithm 1 Optimal Task Scheduling Algorithm Based On Lyapunov Theory

1: **Obtain the queue state $Q(t)$, mobile device state $S_l(t)$, MEC server state $S_c(t)$ at the beginning of each time slot t.**
2: **Find the system case discussed in Section III.**
3: **Obtain the system case k.**
4: **Determine $\pi(t)$ by solving:**
5: $\min_{\pi(t)} \quad -Q(t)U(t) + VD(t)$
6: $s.t. \quad (12)$
7: **Set $t = t + 1$ and update $Q(t)$, $S_l(t)$, $S_c(t)$ according to (2), (8), (9) respectively.**

Meanwhile, P1 is not equivalent to P2. However, if the control parameter V is sufficiently large, the solution of $P1$ is very close to P2. In order to investigate how the performance of the proposed algorithm is, we have the following lemma.

Lemma 2: Let $\mathbb{E}[D^{Alg}(t)] = \bar{D}^{Alg}$ and $\mathbb{E}[D^{Opt}(t)] = \bar{D}^{Opt}$ be the average transmission data per task obtained by solving P2 and the optimal value of $P1$, respectively. We then have:

$$\bar{D}^{Opt} \leq \bar{D}^{Alg} \leq \frac{C_{max}}{V} + \bar{D}^{Opt}. \quad (20)$$

Proof: The proof is provided in Appendix B. ■

VI. TRADEOFFS BETWEEN COMMUNICATIONS, COMPUTING AND CACHING

In this section, we reveal the tradeoffs between the average transmission data per task $\bar{D}^{Opt}$, the computing f_l and caching M abilities of the mobile VR device. Then, for maintaining a target $\bar{D}^{Opt}$, the tradeoff between R and f_l is also discussed.

A. TRADEOFFS OF $\bar{D}^{Opt}$, f_l AND M

When $T \to \infty$, the total number of the task is close to λT. The time slots needed to process a task are at least $\bar{N}_l$ and $\bar{N}_c$ in the local computation mode and the MEC computation mode, respectively. The total time slots is T, hence the number of the task can be scheduled to the local computation mode and the MEC computation mode is at most $\frac{T}{\bar{N}_l}$ and $\frac{T}{\bar{N}_c}$, respectively. Let $\beta\frac{T}{\bar{N}_l}$ and $\beta'\frac{T}{\bar{N}_c}$ be the number of the task scheduled to the local computation mode and the MEC computation mode, respectively. Here, we have $\beta \in [0, 1]$ and $\beta' \in [0, 1]$. Therefore, we can obtain

$$\lim_{T\to\infty}\sum_{t=1}^{T} U(t) = \beta\frac{T}{\bar{N}_l} + \beta'\frac{T}{\bar{N}_c}, \quad (21)$$

$$\lim_{T\to\infty}\sum_{t=1}^{T} A(t) = \lambda T. \quad (22)$$

In order to ensure the delay constraint, we have:

$$\lim_{T\to\infty}\frac{1}{T}Q(T+1) = 0. \quad (23)$$

According to (3)-(7), we have $Q(t) \geq U(t)$. Therefore, we have the following condition based on (2)

$$\begin{aligned}\lim_{T\to\infty}\frac{1}{T}Q(T+1) &= \frac{1}{T}(Q(T) - U(T) + A(T)) \\ &= \frac{1}{T}(Q(1) - \sum_{t=1}^{T} U(t) + \sum_{t=1}^{T} A(t)). \end{aligned} \quad (24)$$

Consider the initial state $Q(1) = 0$, and substituting (21) and (22) into (24), we have

$$\lim_{T\to\infty}\{\beta\frac{1}{\bar{N}_l} + \beta'\frac{1}{\bar{N}_c}\} = \lambda. \quad (25)$$

The average transmission data per task $\bar{D}_t$ can be given by

$$\bar{D}_t = p\bar{D}_{lt} + (1-p)\bar{D}_{ct}. \quad (26)$$

where p is denoted as the proportion of the tasks processed at the mobile VR device, and can be obtained by the following condition

$$\lim_{T\to\infty} p = \beta\frac{T}{\bar{N}_l}/\lambda T = \frac{\beta}{\lambda\bar{N}_l}. \quad (27)$$

Consider $P1$ is feasible. Notice $\bar{D}_{ct} \geq \bar{D}_{lt}$, in order to get $\bar{D}^{Opt}$, the proportion p maximization problem can be formulated as:

$$\begin{aligned} \text{P3}: \max \ & \frac{\beta}{\lambda \bar{N}_l} \\ s.t. \ & \lim_{T\to\infty}\{\beta \frac{1}{\bar{N}_l} + \beta^{'} \frac{1}{\bar{N}_c}\} = \lambda, && (28) \\ & \beta \in [0,1], && (29) \\ & \beta^{'} \in [0,1]. && (30) \end{aligned}$$

We next solve P3, and with the optimal value p^*, $\bar{D}^{Opt}$ is the minimum $\bar{D}^*$, as illustrated as following proposition.

Proposition 1: Let $\mathbb{E}[D_{ct}(t)] = \bar{D}_{ct}$ and $\mathbb{E}[D_{lt}(t)] = \bar{D}_{lt}$ denote the average transmission data of MEC computation model and local computation model, respectively. And let $\mathbb{E}[D_{lc}(t)] = \bar{D}_{lc}$ denote the average computation data of local computation model. We thus have

$$\bar{D}^{Opt} = \begin{cases} \bar{D}_{ct} - \dfrac{1}{\lambda \bar{N}_l}(\bar{D}_{ct} - \bar{D}_{lt}), & \dfrac{1}{\bar{N}_l} + \dfrac{1}{\bar{N}_c} \geq \lambda > \dfrac{1}{\bar{N}_l}, \\ \bar{D}_{lt}, & \dfrac{1}{\bar{N}_l} \geq \lambda, \\ \textit{infeasible}, & \dfrac{1}{\bar{N}_l} + \dfrac{1}{\bar{N}_c} < \lambda. \end{cases} \tag{31}$$

where we use $\mathbb{E}[N_c(t)] = \bar{N}_c$ and $\mathbb{E}[N_l(t)] = \bar{N}_l$ to denote the exception of the time slots required to complete a task in MEC computation model and local computation model, respectively.

Proof: The proof is provided in Appendix C. ■

It is worth mentioning that the optimal average transmission data per task $\bar{D}^{Opt}$ is dependent on $\bar{D}_{ct}$, $\bar{D}_{lt}$ and $\bar{N}_l$. Besides, we have $\bar{N}_l = \mathbb{E}[N_l(t)] = \mathbb{E}[\lceil D_{lc}(t)W/(f_l\Delta) + D_{lt}(t)/R\Delta \rceil]$, where $D_{lc}(t)$ is independent of f_l. We thus present the following proposition.

Proposition 2: Define $K = \mathbb{E}[K_t]$, and then we have

$$\bar{D}_{ct} = \mathbb{E}[D_{ct}(t)] = \phi \tau K, \tag{32}$$

$$\bar{D}_{lt} = \mathbb{E}[D_{lt}(t)] = \tau \sum_{n=M+1}^{N} \sum_{k} \{1 - (1-p_n)^k \Pr(K_t = k)\}. \tag{33}$$

Proof: The proof is provided in Appendix D. ■

Interestingly, we can observe the following results from Proposition 1 and Proposition 2

- **The minimum of average transmission data per task $\bar{D}^{opt}$ decreases with the computing ability of the mobile VR device f_l when $\frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c} \geq \lambda > \frac{1}{\bar{N}_l}$.**
- **The minimum of average transmission data per task $\bar{D}^{opt}$ is independent of the computing ability of the mobile VR device f_l when $\lambda \leq \frac{1}{\bar{N}_l}$.**
- **The minimum of average transmission data per task $\bar{D}^{opt}$ decreases with the caching size M when $\lambda < \frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c}$.**

Proposition 3: The average queue length should satisfy the following condition

$$\bar{Q} \leq \frac{C_{max}}{\theta_{max}} + \phi V \bar{D}. \tag{34}$$

where $\theta_{max} = \frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c} - \lambda$ is the maximum gap between arrival rate and server rate, and $\bar{D} = \mathbb{E}[D(t)] = \tau K$ is the expectation of size of H_t.

Proof: The proof is provided in Appendix E. ■

From Proposition 3, we can see that the average delay is within a bounded deviation $\mathcal{O}(V)$, while the average transmission data per task by using the proposed algorithm decreases inversely proportional to V in Lemma 2. Similar to [6] and [20], there also exists a transmission data-delay tradeoff of $[\mathcal{O}(1/V), \mathcal{O}(V)]$, which means we can balance the average transmission data per task and delay consumption by adjusting V.

B. TRADEOFF OF R AND f_L

Proposition 1 and Proposition 2 show that by changing R, f_l or M, $\bar{N}_l$ or $\bar{D}_{lt}$ influence the average transmission data per task D^{Opt} when $\frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c} \geq \lambda > \frac{1}{\bar{N}_l}$. Besides, we know that when V is sufficiently large, D^{Alg} is very close to D^{Opt}. This suggest that, as long as V is sufficiently large, we can interchange the communications rate R, the computing ability f_l, and the caching ability M to maintain the same system performance D^{Opt} or D^{Alg}, and hence we can get the tradeoff between these parameters at a target D^{Opt}. Here, we show a tradeoff in the following theorem.

Theorem 1: Given a fixed values D^{Opt} and the caching size M, the tradeoff between the transmission rate R and the computing ability f_l of mobile VR device is given by

$$R \approx \frac{Z_1}{Z_2 - \frac{Z_3}{f_l}}, \ \text{when} \ f_l \in [f_l^{min}, f_l^{max}], \tag{35}$$

where

$$\begin{aligned} D(M) &= \bar{D}_{lt} = \mathbb{E}[D_{lt}(t)] \\ &= \tau \sum_{n=M+1}^{N} \sum_{k} \{1 - (1-p_n)^k \Pr(K_t = k)\}, && (36) \\ Z_1 &= \lambda(\phi K \tau - D^{Opt}) D(M), && (37) \\ Z_2 &= \phi K \tau \Delta - D(M)\Delta, && (38) \\ Z_3 &= KW\lambda\tau(\phi K \tau - D^{Opt}). && (39) \end{aligned}$$

f_l^{max} and f_l^{min} are shown at the bottom of the next page.

Proof: It is easy to obtain Theorem 1 by taking $\bar{N}_l = \mathbb{E}[N_l(t)] = \mathbb{E}[\lceil D_{lc}(t)W/(f_l\Delta) + D_{lt}(t)/R\Delta \rceil] \approx \mathbb{E}[D_{lc}(t)W/(f_l\Delta) + D_{lt}(t)/R\Delta]$ into (31). Here, we have $f_l \in [f_l^{min}, f_l^{max}]$ to satisfy $\frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c} \geq \lambda > \frac{1}{\bar{N}_l}$ given M and D^{Opt}. ■

From Theorem 1, we can see that R is inversely proportional to f_l, when $f_l \in [f_l^{min}, f_l^{max}]$.

VII. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed optimal scheduling policy by simulations. We consider a cache-enabled MEC system where the number of contents $N = 100$, the time slot $\Delta = 1s$, the length of each content $\tau = 1Mbits$, the average arrival rate $\lambda = 0.2$, the mobile device CPU frequency $f_l = 500MHz$, the MEC server CPU frequency $f_c = 1GHz$, cache capacity $M = 10$, $\phi = 2$ and K_t is distributed uniformly in [40, 60]. We assume the content popularity distribution is identical among all elements of a task, which follows the Zipf distribution. Thus, at time slot t, the probability that the k_t-th content of a task is the j-th content in $\mathbb{F}$ is given by

$$p_j = \frac{1/j^{\alpha}}{\sum_{k=1}^{N} 1/k^{\alpha}}, \quad j = 1 \cdots N \tag{42}$$

where $\alpha \geq 0$ characterizes the skewness of the popularity distribution. We set $\alpha = 0.8$ in simulations. In the following, the average communication cost is defined as the average number of transmission contents per task since all contents have the same size. We consider the **MEC computation policy** and the **local computation policy** as two baselines, which executes all the tasks in the MEC server and the mobile device, respectively.

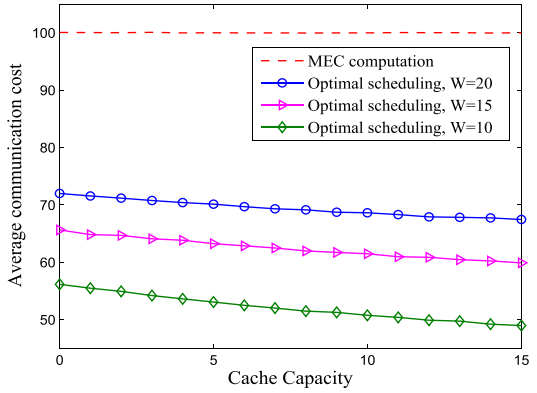

FIGURE 3. The average communication cost per task vs. cache capacity.

A. COMMUNICATION VS. CACHING

Fig.3 shows that the average communication cost achieved by the proposed optimal scheduling policy decreases with the cache capacity. That means the average communication cost can be traded off by the cache capacity to keep the queue length stable, which verifies the tradeoff presented by Proposition 2. Taking $M = 4$ and $W = 10$ for example, the computing and caching resources of mobile VR device can bring 45% gain in saving the communication cost. Moreover, the scheduling policy always outperforms MEC computation policy even when there are no contents cached in the mobile device. This is because the optimal scheduling always executes a part of tasks by local computation policy, and the redundant transmission of contents needed in those tasks can be avoided.

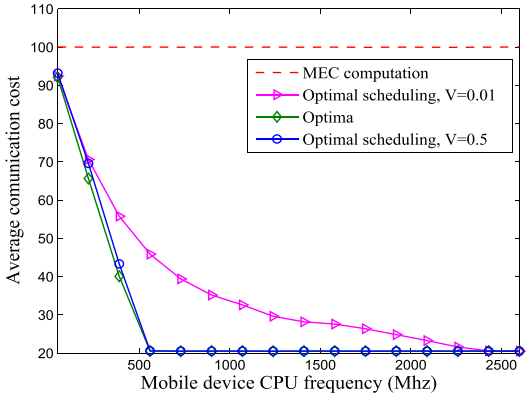

FIGURE 4. The tradeoff between average communication cost and mobile device computing ability.

B. COMMUNICATION VS. COMPUTING

Fig.4 presents the tradeoff between the average communication cost and the mobile device computing ability f_l, to keep the average queue length stable. The increase of the mobile device computing ability f_l decreases the average communication cost. The reason for it is that the increase of f_l decreases the time slots of local execution, and more tasks will be executed by local computation policy. With large V, more tasks are scheduled to mobile device and contribute to save average communication cost. And we can see from the figure that the D^{Alg} is close to D^{Opt} when V is sufficiently large, which verify the lemma 2. Further, when f_l is sufficiently large, the optimal scheme becomes **Local Computation Mode**.

C. IMPACTS OF THE AVERAGE ARRIVAL RATE

From Fig.5, it can be observed that the average communication cost increases with the average arrival rate λ, which

$$f_l^{max} = \frac{\lambda \bar{D} W - \frac{\lambda D(M) Z_3}{Z_1}}{\Delta - \frac{\lambda D(M) Z_2}{Z_1}}, \tag{40}$$

$$\begin{aligned} f_l^{min} = {} & 2(-WZ_1 f_c \lambda \bar{D}^2 Z_3 + D(M) f_c \lambda \bar{D} Z_3^2))/(Z_1(\bar{D}^4 W^4 Z_1^2 \lambda^2 + 2\bar{D}^4 W^3 Z_1 Z_2 f_c \lambda^2 + \bar{D}^4 W^2 Z_2^2 f_c^2 \lambda^2 - 2\bar{D}^3 D(M) W^3 Z_1 Z_3 \lambda^2 \\ & - 2\bar{D}^3 D(M) W^2 Z_2 Z_3 f_c \lambda^2 - 2\bar{D}^3 W^3 Z_1^2 \Delta f_c \lambda - 2\bar{D}^3 W^2 Z_1 Z_2 \Delta f c^2 \lambda - 2\bar{D}^3 W^2 Z_1 Z_3 \Delta f_c \lambda - 2\bar{D}^3 W Z_2 Z_3 \Delta f_c^2 \lambda \\ & + \bar{D}^2 D^2(M) W^2 Z_3^2 \lambda^2 + 4\bar{D}^2 D(M) W^2 Z_1 Z_3 \Delta f_c \lambda + 2\bar{D}^2 D(M) W Z_2 Z_3 \Delta f_c^2 \lambda + 2\bar{D}^2 D(M) W Z_3^2 \Delta f_c \lambda + \bar{D}^2 W^2 Z_1^2 \Delta^2 f_c^2 \\ & - 2\bar{D}^2 W Z_1 Z_3 \Delta^2 f_c^2 + \bar{D} C^2 Z_3^2 \Delta^2 f_c^2 - 2\bar{D} D^2(M) W Z_3^2 \Delta f_c \lambda - 2\bar{D} D(M) W Z_1 Z_3 \Delta^2 f_c^2 + D^2(M) Z_3^2 \Delta^2 f_c^2)^{\frac{1}{2}} - \bar{D}^2 W^2 Z_1^2 \lambda \\ & - \bar{D} Z_1 Z_3 \Delta f_c - D(M) Z_1 Z_3 \Delta f_c + \bar{D} W Z_1^2 \Delta f_c + \bar{D} D(M) W Z_1 Z_3 \lambda + 2\bar{D} D(M) Z_2 Z_3 f_c \lambda - \bar{D}^2 W Z_1 Z_2 f_c \lambda). \end{aligned} \tag{41}$$

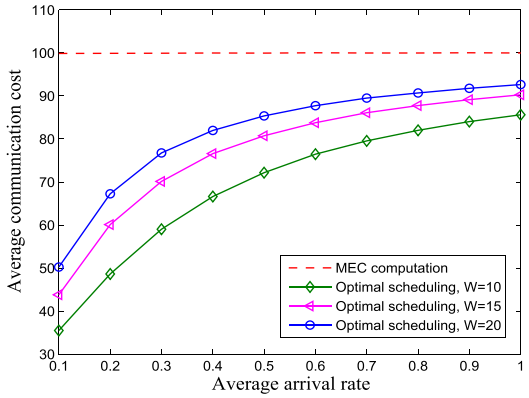

FIGURE 5. The impact of the average arrival rate on the average communication cost.

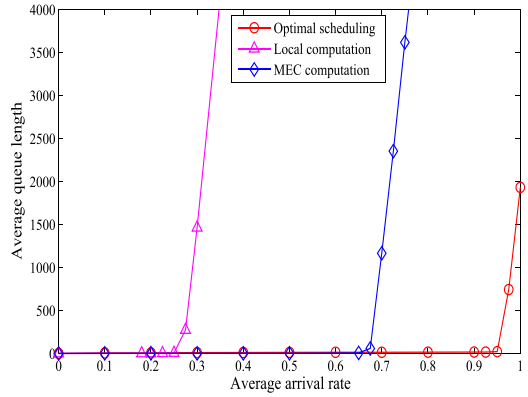

FIGURE 6. The impact of the average arrival rate on the average queue length.

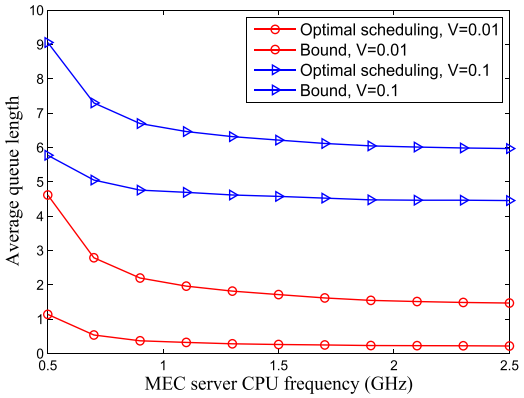

FIGURE 7. The relationship between delay and MEC computation ability.

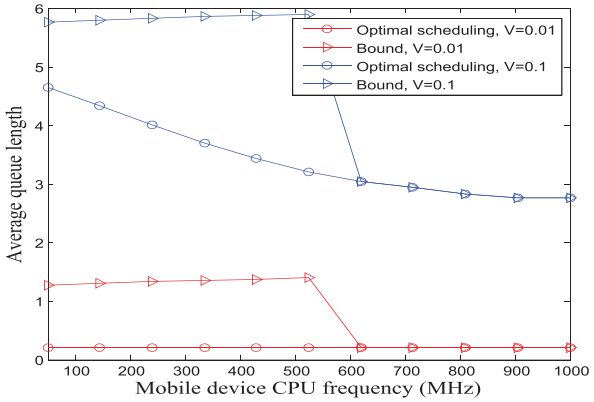

FIGURE 8. The relationship between delay and mobile device computation ability.

follows the lower bound (31) given by Proposition 1. Comparing the optimal scheduling policy with the MEC computation policy, it can be seen that when the arrival rate λ is small, more tasks are scheduled to mobile device. And when the arrival rate λ becomes large, the optimal scheduling simultaneously use both MEC and local computation policy to execute tasks, since only local computation policy can not maintain the queue length stable. And MEC computation policy dominates in the optimal scheduling when λ is very large, since MEC computation policy performs better than local computation policy at this case.

As shown in Fig. 6, the average queue length increases with the average arrival rate. Only when the arrival rate $\lambda \leq 0.25$ ($\lambda \leq 0.65$), the queue length of local (MEC) computation policy are stable. Thus, the proposed optimal scheduling policy performs better than the two baselines when the average arrival rate is very large. This is because the optimal scheduling can simultaneously exploit the computing abilities of MEC server and mobile device to handle the heavy computation load.

D. THE RELATIONSHIP BETWEEN DELAY AND COMPUTATION ABILITY

Fig. 7 shows the average queue length against MEC computation ability f_c under different control parameter V. It can be observed that the bound of the average queue length $\bar{Q}(t)$ decrease with the increase of MEC server computation ability f_c and the smaller V can obtain lower delay, which verify Proposition 3. The curve finally becomes flat because processing a task requires at least one time slot.

Fig. 8 shows the average queue length against mobile device computation ability f_l under different control parameter V. When $V = 0.01$, the queue length has almost no change. This is because when the V is sufficiently small, the system is more sensitive to delay. As such, there are only a few of the task be scheduled to mobile device, hence the increase of f_l has little impact on average delay. When $V = 0.1$, more tasks be scheduled to mobile device so that the processing delay of this part of tasks can be decreased by increasing f_l. The upper bound of the queue length has almost no change in both cases because of $f_c \gg f_l$ and $\frac{1}{N_l} \ll \frac{1}{N_c}$. Although the increase of f_l can also increase $\frac{1}{N_l}$, $\frac{1}{N_l}$ is still small and has little impact on the upper bound of the queue length according to (34).

E. THE TRADEOFF OF R, M, AND f_l

In Fig. 9, the average transmission data per task D^{opt} is 20 ms of the proposed system with different 3C resources allocation,

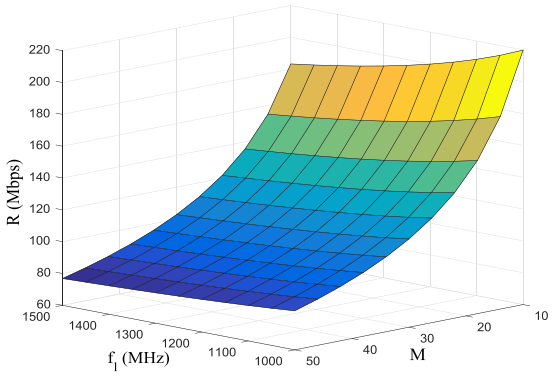

FIGURE 9. Tradeoff among R, f_l and M, where any (R, f_l, M) point in this 3D figure can achieve $D^{opt} = 200$ Mbits.

e.g., $\{R, f_l, M\} = \{220\ Mbps, 1000\ MHz, 10\}$ or $\{R, f_l, M\} = \{80\ Mbps, 1300\ MHz, 45\}$. This means when the mobile VR device has 1300 MHz computing ability and 45 caching capacity, the system takes only 80 Mbps transmission rate to serve the request with $D^{opt} = 200$ Mbit. As one can see, the communication throughput R decreases with increasing computing capability f_l and caching capacity M. As f_l increases, more task be scheduled to the mobile VR device, yielding lower the communication cost. M is similar to f_l. We also observe that the caching ability has more impact on the communication-resource consumption than that of the computing capacity to maintain the same D^{opt}. When the system has small computing ability and caching capability, the large transmission rate is required.

VIII. CONCLUSION

In this paper, we investigated the communication-constrained MEC systems for wireless virtual reality. A transmission data consumption minimization problem with the execution delay constraints was formulated, and we proposed a task scheduling strategy based on Lapunov theory. The tradeoffs between communications, computing, and caching in the proposed system was also dicussed. Simulation results shown that the proposed scheduling strategy achieve a significant reduction in the average transmission data consumption.

APPENDIX A
PROOF OF LEMMA 1

According to (2), we first have

$$Q^2(t+1) \le Q^2(t) + U^2(t) + A^2(t) + 2Q(t)(A(t) - U(t)). \tag{43}$$

Substituting (43) into (16), then (16) can be rewritten as

$$\Delta L(Q(t)) \le \frac{1}{2}\mathbb{E}[U^2(t) + A^2(t)|Q(t)] + Q(t)\mathbb{E}[(A(t) - U(t))|Q(t)]. \tag{44}$$

The task arriving rate $A(t)$ is independent of $Q(t)$. So that we have $\mathbb{E}[A(t)|Q(t)] = \mathbb{E}[A(t)] = \lambda$. The $\mathbb{E}[U^2(t)|Q(t)]$ can be rewritten as

$$\begin{aligned}\mathbb{E}[U^2(t)|Q(t)] &= \mathbb{E}[(U_l(t) + U_c(t))^2|Q(t)] \\ &= \mathbb{E}[U_l^2(t)|Q(t)] + \mathbb{E}[U_c^2(t)|Q(t)] \\ &\quad + 2\mathbb{E}[U_l(t)U_c(t)|Q(t)]\end{aligned} \tag{45}$$

where $U_l(t) = \sum_{i=1}^{2} u_l^i(t)$ and $U_c(t) = \sum_{i=1}^{2} u_c^i(t)$.

According to the definition of expectation, we have

$$\begin{aligned}\mathbb{E}[U_l^2(t)|Q(t)] &= 0^2 \Pr\{U_l(t) = 0|Q(t)\} \\ &\quad + 1^2 \Pr\{U_l(t) = 1|Q(t)\} \\ &= \Pr\{U_l(t) = 1|Q(t)\},\end{aligned} \tag{46}$$

$$\begin{aligned}\mathbb{E}[U_c^2(t)|Q(t)] &= 0^2 \Pr\{U_c(t) = 0|Q(t)\} \\ &\quad + 1^2 \Pr\{U_c(t) = 1|Q(t)\} \\ &= \Pr\{U_c(t) = 1|Q(t)\},\end{aligned} \tag{47}$$

$$\mathbb{E}[U_l(t)U_c(t)|Q(t)] = 1^2 \Pr\{U_c(t)=1,\, U_l(t)=1|Q(t)\}. \tag{48}$$

Notice that the system can not schedule task to mobile VR device or MEC server when they are idle. The server time is longer than the processing time $N_l(t)$ and $N_c(t)$. Thus, we have

$$\mathbb{E}[t_l] \ge \mathbb{E}[N_l(t)] = \bar{N}_l, \tag{49}$$

$$\mathbb{E}[t_c] \ge \mathbb{E}[N_c(t)] = \bar{N}_c, \tag{50}$$

where t_l and t_c denote the server time for the local computation mode and the MEC computation mode, respectively.

According to the definition, the reciprocal of server rate is server time, then for any possible scheduling strategy $\pi(t)$ we have

$$\mathbb{E}[U_l(t)] = \frac{1}{\mathbb{E}[t_l]} \le \frac{1}{\bar{N}_l}, \tag{51}$$

$$\mathbb{E}[U_c(t)] = \frac{1}{\mathbb{E}[t_c]} \le \frac{1}{\bar{N}_c}. \tag{52}$$

Based on the definition of exception, we have

$$\begin{aligned}\mathbb{E}[U_l(t)] &= 0 \Pr\{U_l(t) = 0\} + 1 \Pr\{U_l(t) = 1\} \\ &= \Pr\{U_l(t) = 1\} \le \frac{1}{\bar{N}_l},\end{aligned} \tag{53}$$

$$\begin{aligned}\mathbb{E}[U_c(t)] &= 0 \Pr\{U_c(t) = 0\} + 1 \Pr\{U_c(t) = 1\} \\ &= \Pr\{U_c(t) = 1\} \le \frac{1}{\bar{N}_c}.\end{aligned} \tag{54}$$

For any possible scheduling strategy $\pi(t)$, $\Pr\{U_l(t) = 1\}$ and $\Pr\{U_c(t) = 1\}$ should satisfy (53) and (54). Taking (53) and (54) into (46) and (47), we have

$$\mathbb{E}[U_l^2(t)|Q(t)] \le \frac{1}{\bar{N}_l}, \tag{55}$$

$$\mathbb{E}[U_c^2(t)|Q(t)] \le \frac{1}{\bar{N}_c}. \tag{56}$$

According to the scheduling strategy, $U_l(t)U_c(t)$ can only be non-zero in Case 2. And in this case, $U_l(t)$ and $U_c(t)$ are

independent. Then we have

$$\begin{aligned}\mathbb{E}[U_l(t)U_c(t)|Q(t)] &= \Pr\{U_c(t)=1, U_l(t)=1|Q(t)\}\\ &= \Pr\{U_c(t)=1|Q(t)\}\Pr\{U_l(t)=1|Q(t)\}\\ &\le \frac{1}{\bar{N}_l\bar{N}_c}. \end{aligned} \tag{57}$$

Based on (45), $\mathbb{E}[U^2(t)|Q(t)]$ should satisfy the following condition

$$\mathbb{E}[U^2(t)|Q(t)] \le \frac{1}{\bar{N}_l}+\frac{1}{\bar{N}_c}+\frac{2}{\bar{N}_l\bar{N}_c}. \tag{58}$$

It is easy to obtain $\mathbb{E}[A^2(t)|Q(t)] = \mathbb{E}[A^2(t)] = \lambda^2$ because the arrivals are Bernoulli. Finally, we can obtain (17).

APPENDIX B
PROOF OF LEMMA 2

We assume $P1$ is feasible, and there exists at least one $\pi^*(t)$ for satisfying the constraints of $P1$. $\bar{D}^{Alg}$ and $\bar{D}^{Opt}$ satisfies the following condition:

$$\mathbb{E}[D(t)|Q(t)] = \bar{D}^{Alg} \le \bar{D}^{Opt}+\gamma, \tag{59}$$

where γ is a positive value. According to Little Theorem [25], if the average arriving rate is larger than the average service rate, the queue length tends to infinity with the increase of time slot t. Therefore, if $P2$ can be solved by proposed algorithm, the following condition should be satisfied

$$\mathbb{E}[U(t)|Q(t)] = \lambda+\theta, \tag{60}$$

where θ is a positive value. Substituting (59) and (60) into (18), and with $\gamma \to 0$, we obtain:

$$\Delta L(Q(t)) + V\{D^{Alg}(t)|Q(t)\} \le C_{max} + V\bar{D}^{opt} - Q(t)\theta. \tag{61}$$

Then taking iterated expectation and using the telescoping sums over $t \in \{1.....T\}$, we get

$$\begin{aligned}\mathbb{E}[L(Q(T))] - \mathbb{E}[L(Q(1))] + V\sum_{t=1}^{T}\mathbb{E}[D^{Alg}(t)|Q(t)]\\ \le T(C_{max}+V\bar{D}^{opt}).\end{aligned} \tag{62}$$

We divide (62) with VT and let $T \to \infty$, then we have:

$$\bar{D}^{Alg} \le \frac{C_{max}}{V}+\bar{D}^{Opt}. \tag{63}$$

APPENDIX C
PROOF OF PROPOSITION 1

In order to solve P3, we consider three scenarios as following.

- When $\lambda \le \frac{1}{\bar{N}_l} \le \frac{1}{\bar{N}_c}$, (28) can be rewritten as

$$\beta = \bar{N}_l(\lambda - \beta^{'}\frac{1}{\bar{N}_c}). \tag{64}$$

Because β increase with the decrease of $\beta^{'}$, we set $\beta^{'}$ the minimum value $\beta^{'}=0$, which satisfy constraint (30). Then we have $\beta = \lambda\bar{N}_l$, and it is obviously that β satisfy constraint (29). Therefore, $p=1$ is the optimal solution for P3, yielding $\bar{D}^{Opt} = \bar{D}_l t$ based on (26).

- When $\frac{1}{\bar{N}_l} \le \lambda \le \frac{1}{\bar{N}_c}$, we set β the maximum value $\beta = 1$. According to (28), we have

$$\beta^{'} = (\lambda - \frac{1}{\bar{N}_l})\bar{N}_c. \tag{65}$$

Notice $\bar{N}_c \le 1/\lambda$, then we have

$$\beta^{'} \le (\lambda - \frac{1}{\bar{N}_l})\frac{1}{\lambda} = 1 - \frac{1}{\bar{N}_l\lambda}. \tag{66}$$

Due to $\bar{N}_l\lambda \ge 1$, (66) satisfies constraint (30). Therefore $p = 1/\lambda\bar{N}_l$ is the optimal solution for P3. Substituting $p = 1/\lambda\bar{N}_l$ into (26), we have

$$\bar{D}^{Opt} = \bar{D}_{ct} - \frac{1}{\lambda\bar{N}_l}(\bar{D}_{ct} - \bar{D}_{lt}). \tag{67}$$

- When $\frac{1}{\bar{N}_l} \le \frac{1}{\bar{N}_c} \le \lambda$, there are two possible conditions. If $\frac{1}{\bar{N}_l}+\frac{1}{\bar{N}_c} \ge \lambda$, similar to the analysis of $\frac{1}{\bar{N}_l} \le \lambda \le \frac{1}{\bar{N}_c}$, we set $\beta = 1$ and it is easy to obtain

$$\beta^{'} = (\lambda - \frac{1}{\bar{N}_l})\bar{N}_c \le (\frac{1}{\bar{N}_c}+\frac{1}{\bar{N}_l}-\frac{1}{\bar{N}_l})\bar{N}_c = 1. \tag{68}$$

$\beta^{'}$ satisfies the constraint (30). Therefore, $p = 1/\lambda\bar{N}_l$ is the optimal solution for P3 and $\bar{D}^{Opt}$ is (67).

If $\frac{1}{\bar{N}_l}+\frac{1}{\bar{N}_c} < \lambda$, this condition can not satisfy constraint (28), which means the queue length is unstability. Based on above analysis, the proposition is proved.

APPENDIX D
PROOF OF PROPOSITION 2

Equation (32) can be simply rewritten as

$$\mathbb{E}[D_{ct}(t)] = \mathbb{E}[\phi\tau K_t] = \phi\tau\mathbb{E}[K_t] = \phi\tau K. \tag{69}$$

For (33), the MEC server only transmits the corresponding chunks which are not stored in the mobile VR device, and eliminates the redundancy among the chunks in the task. For task H_t, the probability that the k_t-th chunk is not the n-th chunk in $\mathbb{F}$ is $1-p_n$. If the number of the chunks contained in a task is k, the probability that chunk F_n exists in a task is $1-(1-p_n)^k$. G_n^t denotes whether F_n is requested in H_t, hence we have

$$\begin{aligned}\Pr(G_n^t=1) &= \sum_k [1-(1-p_n)^k]\Pr(K_t=k)\\ &= \sum_k \Pr(K_t=k)] - \sum_k (1-p_n)^k\Pr(K_t=k)\\ &= 1-\sum_k (1-p_n)^k\Pr(K_t=k).\end{aligned} \tag{70}$$

Equation (33) thus can be rewritten as

$$\mathbb{E}[D_{lt}(t)] = \mathbb{E}[\tau \sum_{n=M+1}^{N} G_n^t] = \tau(\mathbb{E}[G_{M+1}^t] + \ldots + \mathbb{E}[G_N^t])$$
$$= \tau \sum_{n=M+1}^{N} \Pr(G_n^t = 1). \quad (71)$$

Taking (70) into (71) we can obtain:

$$\mathbb{E}[D_{lt}(t)] = \tau \sum_{n=M+1}^{N} \sum_{k} 1 - (1-p_n)^k \Pr(K_t = k). \quad (72)$$

APPENDIX E
PROOF OF PROPOSITION 3

According to Lemma 2, by solving $P2$ in each time slot t, we can obtain D^{Alg}, and the scheduling strategy $\pi^*(t)$ which minimize the right-hand-side of the drift-plus-penalty inequality (18) on every time slot t. We use the $U(\pi^*(t))$ denote the server rate achieved by decisions $\pi(t)$. The exception of the communication cost achieved by solving $P2$ in each time slot is $\mathbb{E}[D(\pi^*(t))]$. For a giving arrival rate λ, we thus have

$$\mathbb{E}[U(\pi^*(t))] \geq \lambda,$$
$$\mathbb{E}[D(\pi^*(t))] = D^{Alg}(\lambda), \quad (73)$$

where $D^{Alg}(\lambda)$ is the expected communication cost by solving $P2$ in each time slot when arrival rate is λ.

Based on (51) and (52), the exception of the server rate that can be achieved by any possible $\pi(t)$ should satisfy

$$\mathbb{E}[U(\pi(t))] \leq \mathbb{E}[U_l(t)] + \mathbb{E}[U_c(t)] \leq \frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c}. \quad (74)$$

To satisfy the delay constraint (13), the exception of server rate should greater than arrival rate. When the arrival rate is λ, the gap between arrival rate and server rate θ should satisfy

$$0 \leq \theta \leq \frac{1}{\bar{N}_l} + \frac{1}{\bar{N}_c} - \lambda. \quad (75)$$

For the arrival rate $\lambda + \theta$ with $0 \leq \theta \leq \theta_{max}$, the exception of the communication cost achieved by solving $P2$ in each time slot is $\mathbb{E}[D(\pi'(t))]$. We thus have

$$\mathbb{E}[U(\pi'(t))] \geq \lambda + \theta \geq \lambda, \quad (76)$$
$$\mathbb{E}[D(\pi'(t))] = D^{Alg}(\lambda + \theta). \quad (77)$$

Notice that π' is also the feasible solution for $P2$ when arrival rate is λ. And π^* is the optimal solution for $P2$ when arrival rate is λ. According to (18), we have

$$\Delta L(Q(t)) + V\mathbb{E}[D(\pi^*(t))|Q(t)] \leq C_{max} + Q(t)\mathbb{E}[A(t)$$
$$- U(\pi'(t))|Q(t)] + V\mathbb{E}[D(\pi'(t)))|Q(t)]. \quad (78)$$

Plugging (76) and (77) into the right side of the inequality and we thus have

$$\Delta L(Q(t)) + V\mathbb{E}[D(t)|Q(t)]$$
$$\leq C_{max} + Q(t)\lambda - Q(t)(\lambda + \theta) + VD^{Alg}(\lambda + \theta)$$
$$= C_{max} + VD^{Alg}(\lambda + \theta) - \theta Q(t) \quad (79)$$

Then taking iterated expectation and using the telescoping sums over $t \in \{1 \ldots\ldots T\}$, we get

$$\mathbb{E}[L(Q(T))] - \mathbb{E}[L(Q(1))] + V \sum_{t=1}^{T} \mathbb{E}[D(\pi(t))]$$
$$\leq C_{max}T + VTD^{Alg}(\lambda + \theta) - \theta \sum_{t=1}^{T} Q(t). \quad (80)$$

Dividing (80) by θT and taking limits as $T \to \infty$, we obtain

$$\bar{Q} \leq \frac{C_{max} + V[D^{Alg}(\lambda + \theta) - D^{Alg}(\lambda)]}{\theta} \quad (81)$$

For the increasing task arrival rate from λ to $\lambda + \theta$, the communication cost is at most $\bar{D}_{ct}\theta = \phi\bar{D}\theta$. We thus have

$$D^{Alg}(\lambda + \theta) - D^{Alg}(\lambda) \leq \phi\bar{D}\theta. \quad (82)$$

Taking (82) into (81), we can obtain

$$\bar{Q} \leq \frac{C_{max}}{\theta} + \phi V\bar{D} \quad (83)$$

Notice the inequality holds for all θ which satisfy $0 \leq \theta \leq \theta_{max}$, proposition 3 is thus proved.

REFERENCES

[1] Juniper. (Aug. 2017). *Virtual Reality Markets: Hardware, Content & Accessories 2017–2022*. [Online]. Available: https://www.juniperresearch.com/researchstore/innovation-disruption/virtual-reality/hardware-content-accessories
[2] Qualcomm. (Feb. 2017). *Augmented and Virtual Reality: The First Wave of 5G Killer Apps*. [Online]. Available: https://www.qualcomm.com/documents/augmented-and-virtual-reality-first-wave-5g-killer-apps
[3] 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.
[4] H. Liu, Z. Chen, and L. Qian, "The three primary colors of mobile systems," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 15–21, Sep. 2016.
[5] K. Wang, Z. Chen, and H. Liu, "Push-based wireless converged networks for massive multimedia content delivery," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2894–2905, May 2014.
[6] Y. Mao, J. Zhang, Z. Chen, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[7] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.
[8] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
[9] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.
[10] L. Zhang, Y. Xu, W. Huang, L. Yang, J. Sun, and W. Zhang, "A MMT-based content classification scheme for VoD service," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2015, pp. 1–5.
[11] K. Li, C. Yang, Z. Chen, and M. Tao, "Optimization and analysis of probabilistic caching in N -tier heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1283–1297, Feb. 2018.

[12] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.
[13] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[14] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[15] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.
[16] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *Proc. Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.
[17] L. Tianze, W. Muqing, and Z. Min, "Consumption considered optimal scheme for task offloading in mobile edge computing," in *Proc. Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–6.
[18] C. You and K. Huang, "Multiuser resource allocation for mobile-edge computation offloading," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–3.
[19] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4581–4596, Aug. 2013.
[20] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.
[21] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.
[22] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. Wireless On-Demand Netw. Syst. Services (WONS)*, Feb. 2017, pp. 1–2.
[23] Y. Cui, W. He, C. Ni, C. Guo, and Z. Liu. (2017). "Energy-efficient resource allocation for cache-assisted mobile edge computing." [Online]. Available: https://arxiv.org/abs/1708.04813
[24] Z. Wang, Z. Chen, B. Xia, L. Luo, and J. Zhou, "Cognitive relay networks with energy harvesting and information transfer: Design, analysis, and optimization," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2562–2576, Apr. 2016.
[25] S. Ross, *Introduce to Probability Models*. San Francisco, CA, USA: Academic, 2014.

XIAO YANG received the B.E. degree in electronic and information engineering from the School of Electrical and Electronic Information Engineering, North China Electrical Power University, Beijing, China, in 2016. He is currently pursuing the master's degree with the Department of Electronic Engineering, Institute of Wireless Communications Technology, Shanghai Jiao Tong University, Shanghai, China. His research expertise and interests include mobile edge computing and mobile VR/AR delivery system.

ZHIYONG CHEN received the B.S. degree in electrical engineering from Fuzhou University, Fuzhou, China, and the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2011. From 2009 to 2011, he was a visiting Ph.D. student with the Department of Electronic Engineering, University of Washington, Seattle, WA, USA. He is currently an Associate Professor with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China. His research interests include mobile communications-computing caching networks, mobile VR/AR delivery, and mobile AI systems. He served as the Local Arrangement Chair of the IEEE ICC 2019, the Publicity Chair of the IEEE/CIC ICCC 2014, and a TPC member for major international conferences. He currently serves an Associate Editor of the IEEE Access.

KUIKUI LI received the B.E. degree in communications engineering from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electric Engineering, Shanghai Jiao Tong University, China. His research interests include cache-enabled heterogeneous networks, cooperative communications, and mobile edge computing.

YAPING SUN received the B.Eng. degree in communication engineering from Xidian University. She is currently pursuing the Ph.D. degree in electronic engineering with Shanghai Jiao Tong University. Her research interests include applications of stochastic optimization and future wireless communication and computing networks.

NING LIU received the Ph.D. degree from Shanghai Jiao Tong University in 2010. He is currently an Associate Professor with Shanghai Jiao Tong University. His research interests include wireless and mobile communication systems, network media information security, intelligent hardware, and mobile Internet.

WEILIANG XIE received the B.E. and M.E. degrees in information science and technology from Nankai University, Tianjin, China, in 1997 and 2000, respectively, and the Ph.D. degree in information science and technology from Peking University, Beijing, China, in 2003. He is currently a Professorate Senior Engineer with China Telecom Corporation Limited Technology Innovation Center, Beijing. His research interests include mobile networks and wireless communication systems.

YONG ZHAO received the B.E. and M.E. degrees in electromagnetic field and microwave technology from the Beijing Institute of Technology, Beijing, China, in 2000 and 2004, respectively. He is currently a Senior Engineer with China Telecom Corporation Limited Technology Innovation Center, Beijing. His research interests include mobile networks and wireless communication systems.