Dynamic Task Offloading and Resource Allocation for Mobile Edge Computing in Dense Cloud RAN

Qi Zhang, Lin Gui, *Member, IEEE,* Fen Hou, *Member, IEEE,*
Jiacheng Chen, Shichao Zhu, Feng Tian

***Abstract*—With the unprecedented development of smart mobile devices, e.g., Internet of Things devices and smart phones, various computation-intensive applications are explosively increasing in ultra-dense networks (UDNs). Mobile edge computing (MEC) has emerged as a key technology to alleviate the computation workloads of smart mobile devices and decrease service latency for computation-intensive applications. With the benefits of network function virtualization, MEC can be integrated with cloud radio access network (C-RAN) in UDNs for computation and communication cooperation. However, with stochastic computation task arrivals and time-varying channel states, it is challenging to offload computation tasks online with energy-efficient computation and radio resource management. In this paper, we investigate the task offloading and resource allocation problem in MEC-enable dense C-RAN, aiming at optimizing network energy efficiency. A stochastic mixed-integer nonlinear programming problem is formulated to jointly optimize the task offloading decision, elastic computation resource scheduling, and radio resource allocation. To tackle the problem, Lyapunov optimization theory is introduced to decompose the original problem into four individual subproblems which are solved by convex decomposition methods and matching game. We theoretically analyze the tradeoff between energy efficiency and service delay. Extensive simulations evaluate the impacts of system parameters on both energy efficiency and service delay. Simulation results also validate the superiority of the proposed task offloading and resource allocation scheme in dense C-RAN.**

***Index Terms*—Mobile edge computing, cloud radio access network, ultra-dense network, task offloading, resource allocation, Lyapunov optimization.**

I. Introduction

A. Motivation and Goal

ULTRA-dense network (UDN) is envisioned as a promising paradigm for future mobile networks, which provides proximal transmissions and huge access capacities by the dense deployment of small cell base stations [1], [2]. With the unprecedented development of smart mobile devices (SMDs), e.g., Internet of Things devices and smart phones, various computation-intensive applications are explosively increasing in UDNs such as video stream analysis, interactive gaming and wearable virtual reality [3]. Most of these applications are quite demanding in terms of real-time processing and energy consumption. However, SMDs have limited computation capabilities and battery capacity and cannot effectively execute computation-intensive applications locally [4], [5]. Mobile edge computing (MEC) [6], [7] is presented as a prominent technology to overcome the challenge by providing cloud computing capability in close proximity to SMDs. Specifically, MEC can reduce the computation load of SMDs by offloading part of computation tasks to MEC servers. MEC also offers customized services and lower service latency, which is superior to conventional mobile cloud computing [8]. However, due to network densification, task offloading in UDNs will incur drastic inter-cell interference and heavy signaling overhead. Cloud radio access network (C-RAN) is a cost-effective realization of centralized densification [9], which can coordinate inter-cell interference by central baseband unit (BBU) pool. With the benefit of network function virtualization (NFV) [10], MEC server and BBU pool are able to share the same NFV infrastructure. Hence MEC and C-RAN can be merged based on general purpose processors [11]. In this paper, the integrated MEC and C-RAN in UDN is referred to as MEC-enabled Dense C-RAN (MDC-RAN).

Integrating MEC with C-RAN in UDN can bring enormous potential benefits. Since the functionalities of MEC server and BBU pool are virtualized based on standard general purpose processors, MDC-RAN facilitates dynamic and elastic computation resource allocation for task execution and communication processing, which significantly improves the utilization efficiency of computation resource. Besides, in MDC-RAN, MEC and C-RAN are able to interact and exploit computation and communication information to improve the service quality for computation-intensive applications.

However, there are still some challenges to address. Firstly, computation tasks can be executed at both MEC servers and SMDs. Offloading computation tasks to MEC servers reduces the task execution delay but incurs corresponding energy consumption and latency in uplink transmission. In order to balance the energy efficiency and service delay with stochastic

This work is supported in part by the National Natural Science Foundation of China under Grant 61671295, in part by the Shanghai Key Laboratory of Digital Media Processing under Grant 6141B060303, and in part by the Shanghai Sailing Program under Grant 18YF1422100; it is also partly supported by the joint fund from the Ministry of Science and Technology of China and Macau Science and Technology Development Fund under Grant 037/2017/AMJ. *(Corresponding author: Lin Gui.)*

Q. Zhang, L. Gui, and S. Zhu are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: qizhang_sjtu@sjtu.edu.cn; guilin@sjtu.edu.cn; zhushichao@sjtu.edu.cn).

F. Hou is with the State Key Laboratory of IoT for Smart City and the Department of Electrical and Computer Engineering, University of Macau, Macau, China (e-mail:fenhou@um.edu.mo).

J. Chen is with Peng Cheng Laboratory, Shenzhen 518000, China (e-mail: chenjch02@pcl.ac.cn).

F. Tian is with the Shanghai Engineering Center for Microsatellites, Chinese Academy of Sciences, Shanghai 201203, China (e-mail: tianf@microsate.com).

task arrivals, how to offload computation tasks online should be rationally determined. Secondly, most resource management methods for MEC consider the computation resource at MEC servers and BBU pools separately [12], [13], thus cannot be applied to MDC-RAN directly. Based on NFV, the dynamic resource management scheme needs to be redesigned so as to elastically schedule virtual computation resources under different network sizes and task arrival rates. Furthermore, due to the dense feature of MDC-RAN, simultaneous task offloading from SMDs to MEC servers will cause severe inter-cell interference. To guarantee the quality of service for computation-intensive applications, the radio resource assignment should be jointly optimized with task offloading and computation resource allocation.

In this paper, we consider a task offloading scenario in MDC-RAN where computation tasks are executed at MEC servers and SMDs in parallel. With the aid of NFV, the computation resource of general purpose processors can be assigned to virtual MEC servers and virtual BBU pools on demand. To guarantee the efficient utilization of computation and radio resources under stochastic computation task arrivals and time-varying channel states, we propose a dynamic task offloading and resource allocation scheme, aiming at minimizing the long-term average energy consumption per completed task. Specifically, we model the MDC-RAN as a time-slotted multi-queue system. Then, a stochastic mixed integer nonlinear programming problem is formulated to jointly optimize task offloading decision, computation and radio resource allocations. Due to NP-hardness of the joint optimization problem, we decompose it into a series of subproblems based on Lyapunov optimization technique. An online semi-distributed algorithm is developed to solve these subproblems sequentially. We analyze the energy efficiency and service delay tradeoff under the proposed scheme, which provides useful guidelines for balancing the energy and delay performances in MDC-RAN.

B. Main Contributions

In summary, the main contributions of this paper are listed as follows:

- Novel network structure: we integrate MEC with C-RAN in UDN scenario, where the virtual MEC server and virtual BBU pool are deployed and scheduled elastically based on NFV. A weighted energy efficiency optimization problem is formulated with the constraint of queue stability.
- Efficient algorithm design: an online task offloading and resource allocation scheme is developed based on Lyapunov optimization. The task offloading decision and computation resource scheduling are obtained with decomposition methods. The radio resource allocation is addressed by matching game and geometric programming. This scheme is implemented in a semi-distributed way with low-complexity.
- Extensive performance evaluation: we demonstrate the tradeoff between energy efficiency and service delay as $[O(1/V), O(V)]$, which is validated by simulation results. The impacts of system parameters on energy efficiency and service delay are also evaluated. Performance comparisons validate the superiority of proposed task offloading and resource management scheme.

The remainder of this paper is organized as follows. We review the related works in Section II. The system model and problem formulation are presented in Section III and Section IV. The dynamic task offloading and resource allocation scheme is proposed in Section V. We analyze the performance of the proposed scheme in Section VI. Simulation results are given in Section VII. Finally, we conclude this paper in Section VIII.

II. Related Works

In recent years, the integration of MEC, C-RAN and UDN has attracted significant attentions from both academia and industry. European Telecommunications Standard Institute (ETSI) suggested collocating C-RAN and MEC for 5G networks in a white paper [11] and presented a co-located framework base on NFV. This white paper listed the benefits from the perspective of mobile network operators but didn't present the task offloading and resource management method for co-located MEC and C-RAN. An hybrid edge computing framework, *Chimera*, was proposed for vehicular crowdsensing applications in [14]. The authors designed an online task scheduling algorithm to minimize the energy consumption of recruited vehicles with the constrains of application deadline and vehicle incentive. To efficiently transmit, process and cache big IoT data, [15] proposed a UDN-based hierarchical multiple access and computation offloading scheme, which reduced the end-to-end delay and computation cost for massive IoT devices. In [16], the authors investigated the task offloading policy in MEC-enabled UDN and introduced the software defined networking technology to manage the computation resource in edge cloud with a centralized controller. Many studies have considered the backhaul/fronthaul constraints in MEC and C-RAN networks and investigated backhaul/fronthaul scheduling schemes. Wang *et al.* in [13] considered the limited fronthaul capacity in an integrated C-RAN and MEC system. To maximize the time average profit of mobile server provider, an online fronthaul scheduling policy was presented to control fronthaul link state and request dispatching. [17] considered a two-tiered edge cloud enhanced C-RAN. A multi-dimension multi-choice 0-1 knapsack problem was formulated to maximize the successful rate of computation tasks. [18] optimized the offloading strategy, bandwidth and computational resource allocation to maximize the economic profit of network operator in C-RAN with MEC. A Lagrangian dual method was proposed to solve the problem with the constraint of fronthaul capacity.

Several recent studies have focused on the tradeoff between energy efficiency and service delay in MEC. Zhang *et al.* in [19] defined two objective functions for single and multi-cell MEC networks by introducing weight factors of energy and delay. Then the authors proposed a task offloading and radio resource allocation scheme to balance the energy consumption and execution latency. Lyapunov optimization theory is a powerful technique to analyze energy and delay

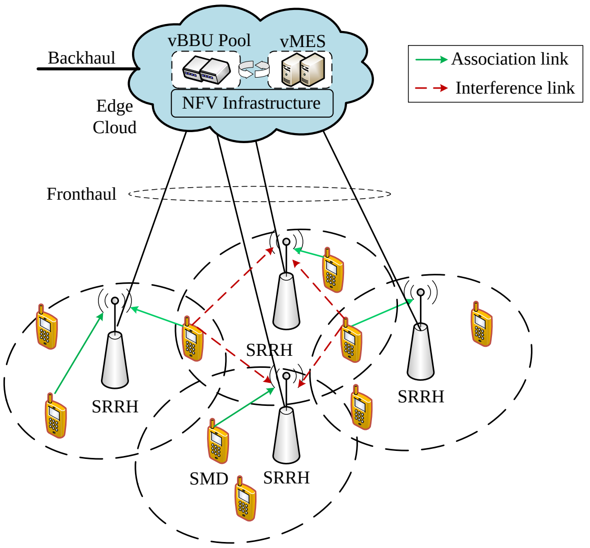


Fig. 1: The network architecture of MDC-RAN.

tradeoff in stochastic networks [20], [21]. It does not require the prior knowledge of task arrival distributions and channel statistics and has been widely used in mobile networks and MEC systems. The authors in [22] formulated a stochastic optimization problem to minimize the average weighted sum power consumption of MEC systems while guaranteeing the stability of task buffers. Different from [22], we design the novel network architecture, optimization objective and semi-distributed algorithm to enhance the resource utilization of MEC-based dense C-RAN. In [23], Mao *et al.* investigated the tradeoff between energy efficiency and delay in multi-user MEC systems with the wireless energy transfer technology. The authors in [24] proposed a gateway selection scheme for heterogeneous cloud-aided multi-UAV systems. The power-delay tradeoff was struck by jointly optimizing task scheduling and computation resource allocation. However, most of existing studies do not consider the energy consumption per completed computation task and ignore the computation resource of BBU pool in C-RAN.

Different from previous studies, this paper presents a novel network architecture based on NFV and investigates an energy efficiency optimization problem by jointly considering the stochastic computation task arrivals and time-varying channel conditions. An online semi-distributed scheme is proposed to efficiently utilize the computation and radio resources and balance the energy efficiency and service delay.

III. System Model

A. System Overview

We consider a MEC-enabled C-RAN scenario in UDN with an edge cloud, N small remote radio heads (SRRHs) and M SMDs as illustrated in Fig.1. The edge cloud consists of several standard general purpose processors. The virtual BBU (vBBU) pool and virtual MEC servers (vMES) are hosted by containers or virtual machines based on the NFV infrastructure at edge cloud. Because of the virtualization of

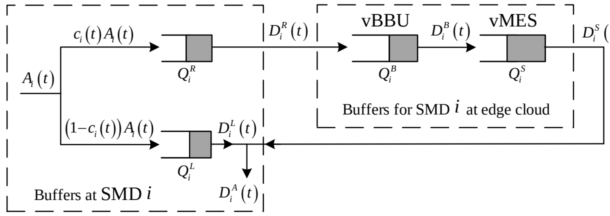


Fig. 2: The queueing model of task arrival and process in MDC-RAN.

BBU pools and MEC servers, it's flexible and scalable to assign the computation resource of general purpose processors to the vBBU pool and vMES. SRRHs are densely deployed and connected to edge cloud via high-speed fiber fronthaul links [14–16]. We denote the sets of SRRHs and SMDs as $\mathcal{N} = \{1, 2, \ldots, N\}$ and $\mathcal{M} = \{1, 2, \ldots, M\}$. The following parts will describe the model of MDC-RAN in detail.

B. Computation Task and Queueing Models

The MDC-RAN is modeled as a discrete time-slotted system. The time slot is index by $t \in \mathcal{T} = \{0, 1, 2, \ldots\}$ and the length of each time slot is T. In each time slot, there are $A_i(t)$ (bits) computation tasks generated by SMD i. We assume $A_i(t)$, bounded by $A_i^{\max}$, is independent and identically distributed (i.i.d.) in different time slots with $\mathbb{E}[A_i(t)] = \lambda_i T$, where λ_i (in bits/s) is the average task arrival rate. Typically, we consider the data-partitioned-oriented computation task model [6] in MDC-RAN. That is, the input tasks can be divided into two parts and processed at SMDs and edge cloud in parallel. Denote the proportion of tasks processed at edge cloud as $c_i(t)$. The other $(1 - c_i(t))A_i(t)$ tasks are executed locally, for all $i \in \mathcal{M}, t \in \mathcal{T}$.

The queueing model of task arrival and process is illustrated in Fig.2. There are two task buffers with sufficiently large sizes in each SMD, where Q_i^L stores local executed tasks and Q_i^R stores the tasks to be offloaded to edge cloud. At edge cloud, there also exist two task buffers for each SMD. One is at vBBU pool for baseband signal processing and the other is at vMES for task execution, which are denoted as Q_i^B and Q_i^S. Meanwhile, $Q_i^L(t)$, $Q_i^R(t)$, $Q_i^B(t)$, and $Q_i^S(t)$ denote the corresponding queue lengths of Q_i^L, Q_i^R, Q_i^B, and Q_i^S, respectively.

We take the video stream analysis as an example to depict the task queuing model in MDC-RAN. SMD i generates $A_i(t)$ video data in time slot t. $(1 - c_i(t))A_i(t)$ video data are queued in Q_i^L to be analyzed locally, and the remaining $c_i(t)A_i(t)$ video data enters into queue Q_i^R for offloading to edge cloud. In uplink transmission, video data in Q_i^R is first transmitted to Q_i^B in vBBU pool for baseband signal processing, e.g., demodulation and channel decoding, etc. After obtaining original video data from vBBU pool, video data are delivered into Q_i^S in vMES for video stream analysis. Then, the analysis results are returned to SMD i.

The number of tasks departing from Q_i^L and Q_i^R in time slot t are $D_i^L(t)$ and $D_i^R(t)$, respectively. It's assumed that the tasks arrived in time slot t are started to be executed from the next time slot. Thus, the queueing dynamics of Q_i^L and Q_i^R

are modeled as following:

$$Q_i^L(t+1) = [Q_i^L(t) - D_i^L(t)]^+ + (1-c_i(t))A_i(t), \quad (1)$$
$$Q_i^R(t+1) = [Q_i^R(t) - D_i^R(t)]^+ + c_i(t)A_i(t), \quad (2)$$

where $[x]^+ = \max(x, 0)$. Denote the number of tasks processed by vBBU pool and vMES at each time slot as $D_i^B(t), D_i^S(t)$. Hence $Q_i^B(t)$ and $Q_i^S(t)$ evolve as:

$$Q_i^B(t+1) = [Q_i^B(t) - D_i^B(t)]^+ + D_i^R(t), \quad (3)$$
$$Q_i^S(t+1) = [Q_i^S(t) - D_i^S(t)]^+ + D_i^B(t). \quad (4)$$

The transmission queues from edge cloud back to SMDs are negligible in this paper, because the sizes of computation results are generally small, and the downlink transmission rate from edge cloud to SMDs is relative high [16], [25].

C. Local Execution Model

Based on dynamic voltage and frequency scaling technique [22], SMDs and edge cloud are able to adjust CPU-cycle frequencies to decrease power consumption. The computation intensity of arrival tasks is denoted as $\gamma_i(t)$ (in cycles/bit), $\forall i \in \mathcal{M}, t \in \mathcal{T}$. Denote the CPU-cycle frequency of SMD i in time slot t as $f_i^L(t)$ (in cycles/s) bounded by the maximum computation capacity $f_i^{\max}$. Therefore, the number of tasks departing from local task queue Q_i^L is given by

$$D_i^L(t) \quad = \quad \frac{Tf_i^L(t)}{\gamma_i(t)}. \quad (5)$$

The power consumption model [6] of SMD i for local execution is $p_i^L(t) = \kappa_i^L[f_i^L(t)]^3$, where κ_i^L is a constant power coefficient. Because we consider a time-slotted system, the computation and radio resources assigned to SMDs remain unchangeable until next time slot. The energy consumption model is given by:

$$E_i^L(t) \quad = \quad \kappa_i^L T f_i^L(t)^3. \quad (6)$$

D. Communication Model

With the dense deployment of SRRHs, it is highly possible that an SMD is in the coverage of several SRRHs. The association rule between SMDs and SRRHs has been widely studied in [26], [27]. In this paper, we assume that SMDs have associated with SRRHs and orthogonal frequency division multiple access (OFDMA) is used in the uplink transmission from SMDs to SRRHs. Denoted the set of subchannels as $\mathcal{K} = \{1, 2, \ldots, K\}$ and the bandwidth of each subchannel is W_s Hz. Assume that each SMD is only permitted to occupy one subchannel. $x_{ij}^k(t) = 1$ indicates that SMD i is associated to SRRH j on subchannel k, and $x_{ij}^k(t) = 0$ otherwise. We assume SRRHs are able to reuse all the subchannels $\mathcal{K}$. Given bandwidth allocation profiles $\mathbf{X}(t) = \{x_{ij}^k(t) | i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T}\}$, the signal-to-interference-plus-noise ratio (SINR) in uplink transmissions can be written as

$$\Upsilon_{ij}^k(\mathbf{X}(t), \mathbf{P}(t)) = \frac{x_{ij}^k(t) p_i^T(t) h_{ij}^k(t)}{n_0 W_s + \sum\limits_{m \neq i} \sum\limits_{n \neq j} x_{mn}^k(t) p_m^T(t) h_{mj}^k(t)},$$

where $\mathbf{P}(t)$ is the set of SMD's transmit power $p_i^T(t)$, $h_{ij}^k(t)$ is the channel power gain from SMD i to SRRH j. Denote the power spectral density of additive Gaussian noise as n_0. The number of tasks offloaded from SMD i to edge cloud in time slot t is given by

$$D_i^R(t) = \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N}} W_s T \log_2 \left[1 + \Upsilon_{ij}^k(\mathbf{X}(t), \mathbf{P}(t))\right]. \quad (7)$$

E. Computation Model of Edge Cloud

We assume that the edge cloud is hosted on standard general purpose processors whose maximum computation capacity is $F_G^{\max}$ (in cycles/s). The computation resource of edge cloud can be assigned for communication processing at vBBU pool and task executing at vMES on demand. Let $f_i^B(t)$ and $f_i^S(t)$ (in cycles/s) denote the computation resource assigned to vMES and vBBU pool in time slot t, which satisfy

$$\sum_{i \in \mathcal{M}} [f_i^B(t) + f_i^S(t)] \quad \leq \quad F_G^{\max}, \quad \forall t \in \mathcal{T}. \quad (8)$$

The task departure rate from vBBU pool is related to the amount of assigned computation resource [12], [28]. Without loss of generality, we assume that one computation operation is equal to one CPU cycle. In OFDMA systems, given antenna number and assigned bandwidth of SMD i, the task departure rate $R_i^B(t)$ (bits/s) from vBBU pool is linearly increased with the allocated computation resource $F_i^B(t)$ (cycles/s) [12], which can be written as:

$$R_i^B(t) \quad = \quad \frac{1}{\beta}[F_i^B(t) - F_0], \quad (9)$$

where β (in cycles/bit) is the computation intensity of communication processing on general purpose processors. F_0 is a constant that denotes the minimal computation resource required by the vBBU pool when processing uplink data. For simplicity, we don't schedule the fixed computation resource F_0 and only consider the remaining computation resource $F_i^B(t) - F_0$, i.e., $f_i^B(t)$. The number of tasks processed by vBBU pool in time slot t is

$$D_i^B(t) \quad = \quad TR_i^B(t) = \frac{Tf_i^B(t)}{\beta}. \quad (10)$$

Similar to the local execution model, the number of tasks departing from Q_i^S is given by

$$D_i^S(t) \quad = \quad \frac{Tf_i^S(t)}{\gamma_i(t)}. \quad (11)$$

The energy consumption for communication processing and task execution are given by:

$$E_i^B(t) \quad = \quad \kappa^G T f_i^B(t)^3, \quad (12)$$
$$E_i^S(t) \quad = \quad \kappa^G T f_i^S(t)^3, \quad (13)$$

where κ^G is a power coefficient related to the hardware architecture of general purpose processors.

IV. Problem Formulation

In this section, we first introduce the definitions of energy efficiency of MDC-RAN, average sum queue length and queue stability of task buffers. Then, an energy efficiency optimization problem is formulated.

Definition 1. *The energy efficiency of MDC-RAN, η_{EE}, is defined as the ratio of long-term weighted sum energy consumption to long-term total completed computation tasks.*

From Definition 1, η_{EE} is formulated as:

$$\eta_{EE} \triangleq \frac{\lim\limits_{\tau\to+\infty}\frac{1}{\tau}\sum\limits_{t=0}^{\tau-1}\mathbb{E}\left\{E_A(t)\right\}}{\lim\limits_{\tau\to+\infty}\frac{1}{\tau}\sum\limits_{t=0}^{\tau-1}\mathbb{E}\left\{D_A(t)\right\}} = \frac{\bar{E}_A}{\bar{D}_A},$$

where $E_A(t) = \sum_{i\in\mathcal{M}}\{\omega_i[E_i^L(t) + p_i^T(t)T] + \omega_G[E_i^B(t) + E_i^S(t)]\}$, is the weighted sum energy consumption in time slot t, which includes the energy consumption in local execution, uplink transmission, data processing at vBBU pool and task execution at vMES. Since the energy supply of edge cloud is more sufficient than the finite battery capacity of SMDs, we set different weight factors ω_i and ω_G to balance the energy consumption at SMDs and edge cloud, where $\omega_i, \omega_G \in (0,1)$. $D_A(t) = \sum_{i\in\mathcal{M}}[D_i^L(t) + D_i^S(t)]$ denotes aggregate accomplished computation tasks at SMDs and edge cloud in each time slot. Note that η_{EE} reflects the average energy consumption for processing one-bit task. Hence smaller η_{EE} represents better energy efficiency performance.

According to Little's Law [29], given a task arrival rate, average service delay is proportional to the average sum queue length in MDC-RAN. Denote by $Q_i^A(t)$ the sum queue length of SMD i, i.e., $Q_i^A(t) \triangleq Q_i^R(t) + Q_i^B(t) + Q_i^S(t) + Q_i^L(t)$. Hence the average sum queue length in MDC-RAN can be written as

$$\bar{Q}^A = \lim_{\tau\to+\infty}\frac{1}{\tau}\sum_{t=0}^{\tau-1}\sum_{i=1}^{M}\mathbb{E}\left\{Q_i^A(t)\right\}. \quad (14)$$

The average service delay can be calculated by $\frac{\bar{Q}^A}{\sum_{i=1}^M \lambda_i}$.

Due to the random characteristic of computation task arrivals, queue stability is a crucial constraint to guarantee stringent service delay, which is defined as follows.

Definition 2. *A discrete time queuing process $Q(t)$ is mean rate stable [20], if*

$$\lim_{t\to+\infty}\frac{\mathbb{E}\left\{Q(t)\right\}}{t} = 0. \quad (15)$$

Then, we consider the whole task execution process from task generation to task completion and formulate an energy efficiency optimization problem for MDC-RAN. The system energy efficiency is jointly optimized from the aspects of task offloading decision, computation and radio resource allocation. Specifically, when computation tasks $A_i(t)$ are generated at SMD i, the task offloading decision is first made to determine the task offloading proportion $c_i(t)$. Then, for the computation tasks to be executed at SMDs, the CPU-cycle frequencies $f_i^L(t)$ of SMDs are scheduled. For the computation tasks to be executed at edge cloud, the subchannel $x_{ij}^k(t)$ and transmit power $p_i^T(t)$ in uplink transmission are allocated. Next, edge cloud determines the computation resources $f_i^B(t)$ and $f_i^S(t)$ at vBBU pool and vMES for SMD i. The energy efficiency optimization problem for MDC-RAN is illustrated as following:

$$\begin{aligned}
\min\quad & \eta_{EE} \\
\text{s.t.}\quad & \text{C1}: 0 \le c_i(t) \le 1, \quad \forall i\in\mathcal{M}, t\in\mathcal{T} \\
& \text{C2}: 0 \le f_i^L(t) \le f_i^{\max}, \quad \forall i\in\mathcal{M}, t\in\mathcal{T} \\
& \text{C3}: \sum_{i\in\mathcal{M}}[f_i^B(t) + f_i^S(t)] \le F_G^{\max}, \quad \forall t\in\mathcal{T} \\
& \text{C4}: f_i^B(t) \ge 0, f_i^S(t) \ge 0, \quad \forall i\in\mathcal{M}, t\in\mathcal{T} \\
& \text{C5}: 0 \le p_i^T(t) \le P_i^{\max}, \quad \forall i\in\mathcal{M}, t\in\mathcal{T} \\
& \text{C6}: \sum_{i\in\mathcal{N}} x_{ij}^k(t) \le 1, \quad \forall j\in\mathcal{N}, k\in\mathcal{K}, t\in\mathcal{T} \\
& \text{C7}: \sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{N}} x_{ij}^k(t) \le 1, \quad \forall i\in\mathcal{M}, t\in\mathcal{T} \\
& \text{C8}: x_{ij}^k(t) \in \{0,1\}, \quad \forall i\in\mathcal{M}, k\in\mathcal{K}, t\in\mathcal{T} \\
& \text{C9}: Q_i^L(t), Q_i^R(t), Q_i^B(t), \text{and } Q_i^S(t) \text{ are} \\
& \qquad \text{mean rate stable}, \quad \forall i\in\mathcal{M}, t\in\mathcal{T}, \quad (16)
\end{aligned}$$

where C1 is the constraint of task offloading decision. Constraint C2 implies that the local execution speed is bounded by $f_i^{\max}$. Constraint C3 ensures that the available computation resource at edge cloud does not exceed the maximum CPU-cycle frequency $F_G^{\max}$. Constraint C4 implies that the computation resources assigned to SMDs at vBBU pool and vMES are positive. Constraint C5 guarantees that the transmit power of SMD i does not exceed the maximum value $P_i^{\max}$. C6−C7 imply that each subcannel is allocated to only one SMD in the same small cell and an SMD is assigned one subchannel. Constraint C8 indicates $x_{ij}^k(t)$ are binary variables. The task queues are guaranteed stability by constraint C9.

V. Dynamic Task Offloading and Resource Allocation Algorithm

The energy efficiency optimization problem (16) is a stochastic mixed-integer nonlinear programming problem, which is NP-hard. In this section, we first introduce an equivalent transformation of (16). Next, the initial problem is decomposed into four subproblems based on Lyapunov optimization technique. Finally, a dynamic task offloading and resource allocation scheme is proposed to solve these subproblems.

To tackle the nonlinear fractional property of (16), we introduce a variable $\eta_{EE}(t)$ which is defined by

$$\eta_{EE}(t) = \sum_{\tau=0}^{t-1} E_A(\tau) \Big/ \sum_{\tau=0}^{t-1} D_A(\tau). \quad (17)$$

Then, the original problem (16) can be reformulated as

$$\begin{aligned}
\min\quad & \bar{E}_A - \eta_{EE}(t)\bar{D}_A \\
\text{s.t.}\quad & \text{C1} - \text{C9}. \quad (18)
\end{aligned}$$

Inspired by the Dinkelbachs algorithm [30], the equivalent transformation has been proven effective in [31] and similar methods are widely adopted in [23], [32]. To solve the stochastic optimization problem and investigate the energy efficiency and service delay tradeoff, we adopt Lyapunov optimization technique in this section. Let $\mathbf{\Theta}(t) = (\boldsymbol{Q}^R(t), \boldsymbol{Q}^B(t), \boldsymbol{Q}^S(t), \boldsymbol{Q}^L(t))$ denote the vector of current queue length. Then we define the Lyapunov function and Lyapunov drift function as follows.

$$L(\mathbf{\Theta}(t)) \triangleq \frac{1}{2} \sum_{i\in\mathcal{M}} [Q_i^R(t)^2 + Q_i^B(t)^2 + Q_i^S(t)^2 + Q_i^L(t)^2], \tag{19}$$

$$\Delta\mathbf{\Theta}(t) \triangleq \mathbb{E}\{L(\mathbf{\Theta}(t+1) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\}. \tag{20}$$

Next, the Lyapunov drift-plus-penalty function can be written as

$$\Delta\mathbf{\Theta}(t) + V\mathbb{E}\left\{E_A(t) - \eta_{EE}(t)D_A(t)|\mathbf{\Theta}(t)\right\}, \tag{21}$$

where V (in bit^2/Joule) is a critical control parameter to tune energy efficiency and service delay in MDC-RAN. By minimizing (21), network stability and optimal energy efficiency can be obtained. However, the stochastic and nonlinear Lyapunov drift-plus-penalty function (21) is intractable. Instead of minimizing (21) directly, we first derive it's upper bound in Lemma 1 and then aim to minimize the upper bound of (21).

Lemma 1. *In each time slot, for all $V > 0$ and any queue state $\mathbf{\Theta}(t)$, the upper bound of Lyapunov drift-plus-penalty under any task offloading and resource allocation algorithm is given by*

$$\begin{aligned}
&\Delta\mathbf{\Theta}(t) + V\mathbb{E}\left\{E_A(t) - \eta_{EE}(t)D_A(t)|\mathbf{\Theta}(t)\right\} \\
&\le B + V\mathbb{E}\left\{E_A(t)|\mathbf{\Theta}(t)\right\} \\
&\quad + \mathbb{E}\Bigg\{\sum_{i\in\mathcal{M}} [c_i(t)^2 A_i(t) + c_i(t)(Q_i^R(t) - Q_i^L(t) - A_i(t)) \\
&\qquad\qquad + (Q_i^L(t) + \frac{1}{2}A_i(t))]A_i(t)|\mathbf{\Theta}(t)\Bigg\} \\
&\quad - \mathbb{E}\left\{\sum_{i\in\mathcal{M}} D_i^R(t)[Q_i^R(t) - Q_i^B(t)]|\mathbf{\Theta}(t)\right\} \\
&\quad - \mathbb{E}\left\{\sum_{i\in\mathcal{M}} D_i^B(t)[Q_i^B(t) - Q_i^S(t)]|\mathbf{\Theta}(t)\right\} \\
&\quad - \mathbb{E}\left\{\sum_{i\in\mathcal{M}} D_i^S(t)[Q_i^S(t) + V\eta_{EE}(t)]\right\} \\
&\quad - \mathbb{E}\left\{\sum_{i\in\mathcal{M}} D_i^L(t)[Q_i^L(t) + V\eta_{EE}(t)]\right\}
\end{aligned} \tag{22}$$

where B is a positive constant and satisfies

$$B \ge \sum_{i\in\mathcal{M}} \left\{D_i^R(t)^2 + D_i^B(t)^2 + \frac{1}{2}D_i^S(t)^2 + \frac{1}{2}D_i^L(t)^2\right\}.$$

Proof. Please refer to Appendix A. □

From Lemma 1, we formulate four individual subproblems to minimize the right-hand side (R.H.S) of (22). The four subproblems are task offloading decision problem, subchannel and transmit power allocation problem, computation resource allocation problem at edge cloud and local computation resource scheduling problem. In the following, we will solve these subproblems sequentially.

A. Subproblem 1: Task Offloading Decision Problem

In this subsection, we determine the proportion of tasks executed locally and at edge cloud by minimizing the third term of R.H.S of (22), i.e.,

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{M}} \Bigg[c_i(t)^2 A_i(t) + c_i(t)(Q_i^R(t) - Q_i^L(t) - A_i(t)) \\
& \qquad\qquad + (Q_i^L(t) + \frac{A_i(t)}{2})\Bigg] A_i(t) \\
\text{s.t.} \quad & 0 \le c_i(t) \le 1, \quad \forall i \in \mathcal{M}, t \in \mathcal{T}.
\end{aligned} \tag{23}$$

From the structure of (23), we obverse that the task offloading proportion of each SMD is independent, hence (23) can be optimized separately. By minimizing $c_i(t)^2 A_i(t) + c_i(t)(Q_i^R(t) - Q_i^L(t) - A_i(t)) + \left(Q_i^L(t) + \frac{A_i(t)}{2}\right)$ with constraint C1, we can obtain that

$$c_i(t) = \begin{cases} 1, & Q_i^R(t) \le Q_i^L(t) - A_i(t) \\ 0, & Q_i^R(t) \ge Q_i^L(t) + A_i(t) \\ \frac{Q_i^L(t) + A_i(t) - Q_i^R(t)}{2A_i(t)}, & \text{otherwise.} \end{cases} \tag{24}$$

This is because that if $Q_i^R(t) \le Q_i^L(t) - A_i(t)$, Q_i^R has much fewer queue backlogs than Q_i^L; hence all $A_i(t)$ tasks are offloaded to edge cloud. If $Q_i^R(t) \ge Q_i^L(t) + A_i(t)$, the queue backlogs at SMD i are much fewer than that at edge cloud and thus all $A_i(t)$ tasks are executed locally. Besides, if $Q_i^L(t) - A_i(t) < Q_i^R(t) < Q_i^L(t) + A_i(t)$, $A_i(t)$ tasks are allocated to Q_i^L and Q_i^R so that the queue lengths of Q_i^L and Q_i^R are equal. In summary, computation tasks are offloaded to Q_i^R or Q_i^L so as to minimize the difference between $Q_i^R(t)$ and $Q_i^L(t)$.

B. Subproblem 2: Subchannel and Transmit Power Allocation Problem

As mentioned in Section III-D, user association rules have been widely studied. Matching game is an efficient technique to determine SMD-SRRH pairs in a distributed way [26], [27]. In this paper, we follow the former studies and apply the many-to-one matching game to the user association process. In this subsection, we assume that the user association is given and focus on subchannel and transmit power allocation problem. From (22), we extract the terms about $\mathbf{X}(t), \mathbf{P}(t)$ and formulate the subchannel and transmit power allocation problem as following:

$$\begin{aligned}
\min_{\mathbf{X(t)},\mathbf{P(t)}} \quad & \sum_{i\in\mathcal{M}} \{V\omega_i x_{ij}^k(t) p_i^T(t) T \\
& \quad -[Q_i^R(t) - Q_i^B(t)] D_i^R(\mathbf{X}(t), \mathbf{P}(t))\} \\
\text{s.t.} \quad & \text{C5} - \text{C8}.
\end{aligned} \tag{25}$$

Subproblem 2 is a mixed integer nonlinear programming, which is NP-hard. Matching game and geometric programming are promising methods to decrease the computational complexity through distributed implementation. Matching game also considers the competitive, distributed and selfish nature of mobile networks. Hence we propose a two-stage distributed algorithm to solve this problem. More specifically, subchannel allocation is determined by two-side swap matching game. Based on the matching results, the transmit power of each SMD is determined by geometric programming.

1) Stage 1: Subchannel Allocation: Because each SMD is only allocated one subchannel and every subchannel is assigned to at most one SMD, the subchannel allocation can be model as a one-to-one matching game. Denote the associated SMD set of SRRH j as $\mathcal{M}_j, \forall j \in \mathcal{N}$. We first give the definition of matching game for subchannel allocation.

Definition 3. *The matching game Ω for subchannel allocation is defined as a one-to-one bidirectional mapping between two disjoint sets of players, $\mathcal{M}_j$ and $\mathcal{K}$, such that:*

- $\Omega(i) \subseteq \mathcal{K} \cup \{\emptyset\}$ *and* $|\Omega(i)| \leq 1, \forall i \in \mathcal{M}_j$;
- $\Omega(k) \subseteq \mathcal{M}_j \cup \{\emptyset\}$ *and* $|\Omega(k)| \leq 1, \forall k \in \mathcal{K}$;
- $\{k\} = \Omega(i) \leftrightarrow \{i\} = \Omega(k), \forall i \in \mathcal{M}_j, \forall k \in \mathcal{K}$.

$|\Omega(\cdot)|$ is the cardinality of matching outcome set $\Omega(\cdot)$. Given the associated SRRH, every SMD prefers the subchannel with the highest SINR so as to acquire high offloading rate. Thus we define the SINR on subchannel k as the preference function of SMD i:

$$\psi_i^{SA}(k) \quad = \quad \Upsilon_{ij}^k(\mathbf{P}(t)). \tag{26}$$

To decrease energy cost and improve task completion rate, a subchannel prefers to select the SMD with a higher received signal strength, larger queue length and lower aggregated interference to other SRRHs. Hence subchannel k of SRRH j ranks SMDs according to the following preference function:

$$\psi_k^{SA}(i) = [Q_i^R(t) - Q_i^B(t)]R_i^k(t) - V\omega_i p_i^T(t) - \sum_{i' \in \mathcal{M}\backslash\{i\}} [Q_{i'}^R(t) - Q_{i'}^B(t)]\sigma_{i'}^k p_i^T(t) h_{ij'}^k(t), \tag{27}$$

where $R_i^k(t) = \sum_{j\in\mathcal{N}} W_s \log_2[1 + \Upsilon_{ij}^k(\mathbf{P}(t))]$, j' is the associated SRRH of i', $\sigma_{i'}^k$ is referred to as *interference price* [33], which indicates the marginal decrease about uplink transmission rate of SMD $i' \in \mathcal{M}\backslash\{i\}$ caused by the unit increase of inter-cell interference. $\sigma_{i'}^k$ is calculated by

$$\begin{aligned}\sigma_{i'}^k &= -\frac{\partial R_{i'}^k(t)}{\partial I_{i'}^k(t)} \\ &= \frac{W_s p_{i'}^T(t) h_{i'j'}^k(t)}{\ln 2(I_{i'}^k(t) + n_0 W_s + p_{i'}^T(t) h_{i'j'}^k(t))(I_{i'}^k(t) + n_0 W_s)},\end{aligned}$$

where $I_{i'}^k(t) = \sum_{i\in\mathcal{M}\backslash\{i'\}} p_i^T(t) h_{ij'}^k(t)$ is the aggregated interference received by SRRH j'.

Duo to the dense deployment of SRRHs, the preference list of an SMD is affected by other SMD-subchannel matching results, hence the preference relationships between SMDs and SRRHs will vary in the matching process, which is referred to as *externalities* [34]. Considering the externalities in subchannel allocation matching, deferred acceptance algorithm [35] cannot be applied directly, because the matching stability is no longer guaranteed. We introduce the two-side exchange stability [34] and obtain the subchannel allocation by swap matching.

Definition 4. *Give a matching Ω with $\Omega(i) = k, \Omega(i') = k'$, the matching Ω is two-side exchange stable if and only if there does not exist a swap-blocking pair (i, i') such that:*

(a) $\forall s \in \{i, i', k, k'\}, \psi_s^{SA}(\Omega_i^{i'}) \geq \psi_s^{SA}(\Omega)$ *and*
(b) $\exists s \in \{i, i', k, k'\}, \psi_s^{SA}(\Omega_i^{i'}) > \psi_s^{SA}(\Omega)$,
where, $\Omega_i^{i'} = \{\Omega \setminus \{(i,k),(i',k')\}\} \cup \{(i,k'),(i',k)\}$ *is a swap matching.*

This definition implies that a pair of SMDs are permitted to exchange assigned subchannels only when at least one SMD's or subchannel's utility is improved. To achieve the two-side exchange stable matching, we propose a subchannel allocation algorithm, which is implemented in each small cell. The details of the subchannel allocation algorithm are summarized in Alg. 1. Denote $\mathcal{M}_{\text{um}}^{\text{SA}}$ as the set of unmatched SMDs. RL^k is the set of SMDs requesting for subchannel k. After initializing $\mathcal{M}_{\text{um}}^{\text{SA}}$ and RL^k, lines 1-12 are to obtain an initial SMD and subchannel matching by deferred acceptance algorithm. $\mathcal{M}^k$ is the set of SMDs which are assigned subchannel k. Next, SMD pairs exchange matched subchannels until there are not swap-blocking pairs (lines 13-21). Finally, we can obtain a subchannel allocation solution $\mathbf{X}(t)$ with two-side exchange stable.

2) Stage 2: Transmit Power Control: Given subchannel allocation variables $\bar{\mathbf{X}}(t)$, the transmit power control of SMDs in $\mathcal{M}^k$ can be obtained by solving the following problem.

$$\begin{aligned}\min_{\mathbf{P(t)}} \quad & \sum_{i\in\mathcal{M}^k} \left\{V\omega_i T p_i^T(t) + [Q_i^B(t) - Q_i^R(t)] D_i^R(\mathbf{P}(t))\right\} \\ \text{s.t.} \quad & 0 \leq p_i^T(t) \leq P_i^{\max}, \quad \forall i \in \mathcal{M}^k, t \in \mathcal{T}.\end{aligned} \tag{28}$$

Because of the inter-cell interference, problem (28) is nonconvex and intractable by centralized algorithms. We resort to the geometric programming [36], and propose a distributed iterative method. Firstly, we transformed the problem (28) into a convex optimization problem. In previous subchannel matching game, interference coordination is taken into account. That is, when assigning one subchannel to an SMD, each SRRH decreases the interference to other SRRHs as far as possible. Hence we mainly consider the transmit power control in high SINR conditions. With the high SINR regime $\left(\Upsilon_{ij}^k(\bar{\mathbf{X}}(t), \mathbf{P}(t)) >> 1\right)$, $D_i^R(t)$ can be approximated by $W_s T \log_2\left[\Upsilon_{ij}^k(\bar{\mathbf{X}}(t), \mathbf{P}(t))\right]$.

Besides, in order to decouple the interference among SMDs which are allocated the same subchannel, we introduce additional auxiliary variables and equality constraints. In a distributed strategy, we assume that each SMD i can estimate inter-cell interference $I_i^k = \sum_{m\in\mathcal{M}^k\backslash\{i\}} e^{\tilde{p}_m^T(t)} h_{mj}^k(t)$ and define a new variable $\tilde{I}_i^k(t) = \ln(I_i^k(t))$. In addition, changing variable $p_i^T(t)$ by $\tilde{p}_i^T(t) = \ln p_i^T(t)$, the problem (28) can be

Algorithm 1 Matching Game-Based Subchannel Allocation Algorithm

```
Input: $Q_i^R(t), Q_i^B(t), \mathbf{P}(t)$
Output: $\mathbf{X}(t)$
  Initialization : $\mathcal{M}_{\text{um}}^{\text{SA}} = \mathcal{M}_j$, $\text{RL}^k = \emptyset, \forall k \in \mathcal{K}$.
  Each SMD in $\mathcal{M}_j$ constructs the preference list $\text{PL}_i^{\text{SA}}$ in a descending order according to (26);
  repeat
      Each SMD in $\mathcal{M}_{\text{um}}^{\text{SA}}$ sends an association request to the first sunchannel $k$ in $\text{PL}_i^{\text{SA}}$;
      for $k = 1 \ldots K$ do
          Update $\text{RL}^k$ and resort $\text{RL}^k$ in a descending order according to $\psi_k^{SA}(i)$;
          $\mathcal{M}_{\text{um}}^{\text{SA}} \leftarrow \mathcal{M}_{\text{um}}^{\text{SA}} \cup \text{RL}^k \setminus \text{RL}^k[1]$;
          for $i \in \text{RL}^k \setminus \text{RL}^k[1]$ do
              $\text{PL}_i^{\text{SA}} \leftarrow \text{PL}_i^{\text{SA}} \setminus \{k\}$;
          end for
          $\text{RL}^k \leftarrow \text{RL}^k[1]$;
      end for
  until $|\mathcal{M}_{\text{um}}^{\text{SA}}| = 0$ or $\text{PL}_i^{\text{SA}} = \emptyset, \forall i \in \mathcal{M}_{\text{um}}^{\text{SA}}$
  for each $i \in \mathcal{M}_j$ do
      if $i \in \text{RL}^k$ then
          $x_{ij}^k = 1$;
      end if
      Search for each $i' \in \mathcal{M} \setminus \mathcal{M}^k$;
      if $(i, i')$ satisfies the conditions in Define 4 then
          $\Omega \leftarrow \Omega_i^{i'}$.
      end if
  end for
  return $\mathbf{X}(t)$
```

reformulated as follows.

$$
\begin{aligned}
\min_{\tilde{\mathbf{P}}(\mathbf{t})} \quad & \sum_{i \in \mathcal{M}^k} \Bigg\{ V\omega_i e^{\tilde{p}_i^k(t)} + [Q_i^R(t) - Q_i^B(t)] \times \\
& \qquad W_s \log_2 \left[\frac{e^{-\tilde{p}_i^k(t)}}{h_{ij}^k(t)} \left(n_0 W_s + e^{\tilde{I}_i^k(t)} \right) \right] \Bigg\} \\
\text{s.t.} \quad & 0 \le e^{\tilde{p}_i^k(t)} \le P_i^{\max}, \quad \forall i \in \mathcal{M}^k, t \in \mathcal{T} \\
& I_i^k = e^{\tilde{I}_i^k}, \quad \forall i \in \mathcal{M}^k, t \in \mathcal{T}.
\end{aligned} \tag{29}
$$

This problem is a convex optimization problem according to the convex property of log-sum-exp function [37]. Next, we derive $\tilde{p}_i^T(t)$ and $\tilde{I}_i^k(t)$ via dual decomposition. Since problem (29) is convex and satisfies Slater's condition, the duality gap is zero. Based on KKT necessary conditions, we can determine the optimal transmit power $p_i^{T^*}(t)$ and the auxiliary variable $\tilde{I}_i^k(t)$ as following:

$$
\begin{aligned}
& p_i^{T^*}(t) = e^{\tilde{p}_i^T(t)} \\
& = \left[\frac{\left(Q_i^R(t) - Q_i^B(t)\right) W_s}{\ln 2 \left(V\omega_i + \nu_i(t) + \sum_{m \in \mathcal{M}^k \setminus \{i\}} \xi_m(t) h_{in}^k(t) \right)} \right]^+,
\end{aligned} \tag{30}
$$

$$
e^{\tilde{I}_i^k(t)} = \left[\frac{(Q_i^R(t) - Q_i^B(t)) W_s}{\xi_i(t) \ln 2} - n_0 W_s \right]^+. \tag{31}
$$

The Lagrange multipliers $\nu_i(t)$ and consistency prices $\xi_i(t)$ can be updated via subgradient method:

$$
\nu_i^{l+1}(t) = \left[\nu_i^l(t) + \theta_{\nu_i} (e^{\tilde{p}_i^T(t)} - P_i^{\max}) \right]^+, \tag{32}
$$

$$
\xi_i^{l+1}(t) = \xi_i^l(t) + \theta_{\xi_i} \left[\sum_{m \in \mathcal{M}^k \setminus \{i\}} h_{mn}^k(t) e^{\tilde{p}_m^T(t)} - e^{\tilde{I}_i^k(t)} \right], \tag{33}
$$

where $\theta_{\nu_i}^l, \theta_{\xi_i}^l$ are diminishing step sizes which satisfy the square summable but not summable rule [38] and l represents the iterative times.

C. Subproblem 3: Computation Resource Allocation Problem at Edge Cloud

The computation resources assigned to SMDs at vBBU and vMES can be obtained by solving the following subproblem:

$$
\begin{aligned}
\min_{\mathbf{f^B}, \mathbf{f^S}} \quad & \sum_{i \in \mathcal{M}} \Big\{ V\omega_G \kappa^G [f_i^B(t)^3 + f_i^S(t)^3] - \\
& \quad [Q_i^B(t) - Q_i^S(t)] \frac{f_i^B(t)}{\beta} - [Q_i^S(t) + V\eta_{EE}(t)] \frac{f_i^S(t)}{\gamma_i(t)} \Big\} \\
\text{s.t.} \quad & \text{C3} - \text{C4}.
\end{aligned} \tag{34}
$$

It can be verified that subproblem (34) is a convex problem. However, due to coupling constraints C3, it is complex to obtain the optimal solution based on KKT conditions. We resort to dual decomposition method to solve this problem with a distributed method. Denote $V\omega_G\kappa^G$ by Z and the partial Lagrangian of (34) can be written as

$$
\begin{aligned}
& L(\mathbf{f^B}(t), \mathbf{f^S}(t), \mu(t)) \\
& = \sum_{i \in \mathcal{M}} \left\{ Z f_i^B(t)^3 + \left[\mu(t) - \frac{Q_i^B(t) - Q_i^S(t)}{\beta} \right] f_i^B(t) \right\} \\
& \quad + \sum_{i \in \mathcal{M}} \left\{ Z f_i^S(t)^3 + \left[\mu(t) - \frac{Q_i^S(t) + V\eta_{EE}(t)}{\gamma_i(t)} \right] f_i^S(t) \right\} \\
& \quad - \mu(t) F_G^{\max} \\
& = L[\mathbf{f^B}(t), \mu(t)] + L[\mathbf{f^S}(t), \mu(t)] - \mu(t) F_G^{\max},
\end{aligned}
$$

where $\mathbf{f^B}(t) = \{f_1^B(t), \cdots, f_M^B(t)\}$, $\mathbf{f^S}(t) = \{f_1^S(t), \cdots, f_M^S(t)\}$ are computation resource allocation vectors at vBBU pool and vMES. $\mu(t) \ge 0$ is the lagrangian multiplier. The dual function is

$$
g(\mu(t)) = \inf_{\mathbf{f^B}(t)} L[\mathbf{f^B}(t), \mu(t)] + \inf_{\mathbf{f^S}(t)} L[\mathbf{f^S}(t), \mu(t)]. \tag{35}
$$

As subproblem (34) is convex and Slater's condition holds, strong duality can be guaranteed. Hence we can obtain the optimal computation resource allocation at edge cloud by solving the following dual problem.

$$
\max_{\mu(t)} \quad g(\mu(t)), \qquad \text{s.t.} \quad \mu(t) \ge 0.
$$

Given $\mathbf{f^S}(t)$ and dual variables $\mu(t)$, we first derive the minimum value of $L[\mathbf{f^B}(t), \mu(t)]$. We can find that if $Q_i^B(t) - Q_i^S(t) - \mu\beta < 0$, $L[\mathbf{f^B}(t), \mu(t)]$ is monotonically increasing with $f_i^B(t)$, thus the optimal solution is obtained at $f_i^{B^*}(t) =$

0. if $Q_i^B(t) - Q_i^S(t) - \mu\beta \geq 0$, the optimal $f_i^{B^*}(t)$ can be calculated through $\frac{\partial L(\mathbf{f^B}(t),\mu(t))}{\partial f_i^B(t)} = 0$, we can obtain $f_i^{B^*}(t) = \sqrt{\frac{Q_i^B(t)-Q_i^S(t)-\mu\beta}{3\beta Z}}$. Let $I_i^B(t) = [Q_i^B(t) - Q_i^S(t) - \mu\beta, 0]^+$, $f_i^{B^*}(t)$ can be written as

$$f_i^{B^*}(t) \quad = \quad \sqrt{\frac{I_i^B(t)}{3\beta Z}}. \tag{36}$$

Similar to the solution of $f_i^{B^*}(t)$, the optimal computation resource assigned to SMD i at vMES can be derived as

$$f_i^{S^*}(t) \quad = \quad \sqrt{\frac{I_i^S(t)}{3\gamma_i(t)Z}}, \tag{37}$$

where $I_i^S(t) = [Q_i^S(t) + V\eta_{EE}(t) - \mu\gamma_i(t), 0]^+$.

To solve the master optimization problem about $\mu(t)$, we adopt projected subgradient method to update the dual variable.

$$\mu_{l+1}(t) = \left\{ \mu_l(t) + \alpha_l(t) \left[\sum_{i \in \mathcal{M}} [f_i^B(t) + f_i^S(t)] - F_G^{\max} \right] \right\}^+, \tag{38}$$

where $\alpha_l(t)$ is the diminishing step size in l-th iteration and satisfy the square summable but not summable rule [38].

D. Subproblem 4: Local Computation Resource Scheduling Problem

The optimal local CPU-cycle frequency of each SMD can be obtained by solving:

$$\begin{aligned} \min_{\mathbf{f^L}} \quad & \sum_{i \in \mathcal{M}} \left\{ V\omega_i\kappa_i^L f_i^L(t)^3 - [Q_i^L(t) + V\eta_{EE}(t)] \frac{f_i^L(t)}{\gamma_i(t)} \right\} \\ \text{s.t.} \quad & 0 \leq f_i^L(t) \leq f_i^{\max}, \quad \forall i \in \mathcal{M}, t \in \mathcal{T}. \end{aligned} \tag{39}$$

It is observed that the local computation resource scheduling problem (39) can be decoupled with respect to each SMD. The optimal local computation frequencies can be obtained according to the stationary point and boundary values $f_i^{\max}$.

$$f_i^{L^*}(t) \quad = \quad \min\left\{ \sqrt{\frac{Q_i^L(t) + V\eta_{EE}(t)}{3V\omega_i\kappa_i^L\gamma_i(t)}}, f_i^{\max} \right\}. \tag{40}$$

In above, we have sequentially solved four subproblems. Next, we develop a dynamic task offloading and resource allocation scheme, which is referred to as DTORA algorithm. The details of DTORA algorithm are shown in Alg.2.

E. Discussions

1) Implementation Issues: The proposed DTORA algorithm is implemented in a semi-distributed way. Firstly, the task offloading decision is made by each SMD individually according to (24). Secondly, the subchannel allocation is based on matching game. SRRHs interact with SMDs independently without global information. After the subchannel allocation, the transmit power is determined according to each SMD's queue state information, channel gain and power budget. Thirdly, the computation resource allocation at edge cloud is based on Lagrangian dual decomposition by which edge cloud assigns computation resource to SMDs according to queue state information and computation intensity. Fourthly, the local computation resource scheduling is implemented by each SMD according to (40). From above analysis, we can find that the task offloading and resource allocation decisions are made by SMDs, SRRHs, and edge cloud based on local and interactive information. Hence, the proposed DTORA algorithm is semi-distributed and has less reporting overhead compared with centralized methods.

Algorithm 2 Dynamic Task Offloading and Resource Allocation Algorithm (DTORA)

```
Input: $\mathcal{M}, \mathcal{N}, \mathcal{K}$
Output: $\mathbf{c}(t), \mathbf{X}(t), \mathbf{P}(t), \mathbf{f^B}(t), \mathbf{f^S}(t), \mathbf{f^L}(t)$
 1: for each time slot $t$ do
 2:     Determine optimal task offloading decision $\mathbf{c}(t)$ according to (24);
 3:     Obtain subchannel allocation $\mathbf{X}(t)$ via Alg. 1;
 4:     Obtain SMDs' transmit power $\mathbf{P}(t)$ via (30)-(33);
 5:     Calculate the computation resource allocation $\mathbf{f^B}(t), \mathbf{f^S}(t)$ at edge cloud according to (36)-(38);
 6:     Obtain optimal local CPU-cycle frequencies $\mathbf{f^L}(t)$ via (40);
 7:     Update $Q_i^L(t), Q_i^R(t), Q_i^B(t), Q_i^S(t)$ and $\eta_{EE}(t)$ by (1), (2), (3), (4) and (17);
 8:     Set $t = t + 1$
 9: end for
10: return $\mathbf{c}(t), \mathbf{X}(t), \mathbf{P}(t), \mathbf{f^B}(t), \mathbf{f^S}(t), \mathbf{f^L}(t)$
```

2) Computational complexity: The computational complexity of the proposed DTORA algorithm mainly consists of five parts, i.e., lines 2-6 in Alg. 2. We first analyze the computation overhead of each part and then the total computational complexity is given. Since the task offloading decision is made by each SMD according to (24), its computation overhead is linear with the number of SMDs, i.e., $O(M)$. In Alg. 1, we first obtain an initial subchannel allocation by deferred acceptance algorithm of which the computation overhead is $O(KM)$ [39]. Then we execute swap matching processes to obtain a stable subchannel allocation. Consider the worse case that each SMD exchange the assigned subchannel with all other SMDs. The number of exchange is $\frac{M(M-1)}{2}$. Thus, the computation overhead of subchannel allocation is $O(KM) + O\left(\frac{M(M-1)}{2}\right) = O(KM + M^2)$. Given the number of iterations I_{pc}^k of transmit power control on subchannel k, the computation overhead is $O(|\mathcal{M}^k| I_{pc}^k)$. Since $\sum_{k \in \mathcal{K}} I_{pc}^k$ is a constant, the computation overhead of transmit power control is $O(M)$. Similar to transmit power control, the computation overhead of computation resource allocation at edge cloud is also $O(M)$. Because each SMD calculate the local CPU-cycle frequency $f_i^L(t)$ via (40), the computation overhead of local computation resource scheduling is $O(M)$. Therefore, the computational complexity of the proposed DTORA algorithm is $O(M) + O(KM + M^2) + O(M) + O(M) + O(M) = O(KM + M^2)$.

3) Limited Fronthaul: This work considers the case of the unlimited fronthaul where the computation tasks are transmitted to the edge cloud without queueing at SRRHs. When considering the limited fronthaul, we need to add task buffers at SRRHs to model the impact of limited fronthal capacity on the task offloading and design the corresponding fronthaul scheduling algorithm. In specific, we need place task buffers $Q_i^F(t)$ at SRRHs to storage the tasks transmitted from $Q_i^R(t)$. Similar to Lemma 1, it's straightforward to derive the upper bound of Lyapunov drift-plus-penalty with limited fronthaul. By minimizing the upper bound of Lyapunov drift-plus-penalty, the fronthaul scheduling problem can be formulated. This problem is a linear programming problem and can be solved optimally. In addition, the solution of subchannel and transmit power allocation problem also needs to be adjusted when adding task buffers at SRRHs. Since computation tasks pulled out from $Q_i^R(t)$ are sent to $Q_i^F(t)$ instead of $Q_i^B(t)$, $Q_i^B(t)$ in subchannel allocation algorithm (Alg.1) and transmit power control algorithm ((30)-(31)) should be replaced by $Q_i^F(t)$ correspondingly.

VI. Theoretic Analysis

In this section, we analyze the performance of the proposed DTORA algorithm, including the queue stability, the upper bounds of energy efficiency and sum queue length.

To derive these theoretical results, we first give some necessary boundedness assumptions. Assume that $E_A(t), D_i^L(t), D_i^S(t)$ satisfy the following practical boundedness properties:

$$\begin{aligned} &E_A^{\min} \le \mathbb{E}\left\{E_A(t)\right\} \le E_A^{\max}, \\ &D_{\min}^L \le \mathbb{E}\left\{\sum_{i\in\mathcal{M}} D_i^L(t)\right\} \le D_{\max}^L, \\ &D_{\min}^S \le \mathbb{E}\left\{\sum_{i\in\mathcal{M}} D_i^S(t)\right\} \le D_{\max}^S, \end{aligned} \tag{41}$$

where $E_A^{\min}$, $E_A^{\max}$, $D_{\min}^L$, $D_{\max}^L$, $D_{\min}^S$, $D_{\max}^S$ are finite constants. Next, we introduce a Lemma to show that there exist stationary and randomized policies by which task offloading and resource allocation scheme is determined independently in each time slot and the optimal energy efficiency can be obtained arbitrarily closely.

Lemma 2. *Suppose problem (16) is feasible, the system satisfies boundedness assumptions (41), λ is strictly interior to network capacity region Λ, i.e. for a positive value ϵ, and $\lambda+\epsilon$ is still in Λ, then, for all slot t and any $\delta > 0$, there exists an independent, stationary and randomized task offloading and resource allocation algorithm, which satisfies:*

$$\begin{aligned} &\mathbb{E}\left\{\hat{E}_A(t)\right\} \le \mathbb{E}\left\{\sum_{i\in\mathcal{M}} [\hat{D}_i^L(t) + \hat{D}_i^S(t)]\right\} (\eta_{EE}^* + \delta) \\ &\mathbb{E}\left\{\hat{D}_i^L(t) - A_i(t)|\mathbf{\Theta}(t)\right\} = \mathbb{E}\left\{\hat{D}_i^L(t) - A_i(t)\right\} \ge \epsilon \\ &\mathbb{E}\left\{\hat{D}_i^R(t) - A_i(t)|\mathbf{\Theta}(t)\right\} = \mathbb{E}\left\{\hat{D}_i^R(t) - A_i(t)\right\} \ge \epsilon \\ &\mathbb{E}\left\{\hat{D}_i^B(t) - \hat{D}_i^R(t)|\mathbf{\Theta}(t)\right\} = \mathbb{E}\left\{\hat{D}_i^B(t) - \hat{D}_i^R(t)\right\} \ge \epsilon \\ &\mathbb{E}\left\{\hat{D}_i^S(t) - \hat{D}_i^B(t)|\mathbf{\Theta}(t)\right\} = \mathbb{E}\left\{\hat{D}_i^S(t) - \hat{D}_i^B(t)\right\} \ge \epsilon \end{aligned} \tag{42}$$

where $\hat{E}_A(t), \hat{D}_i^L(t), \hat{D}_i^R(t), \hat{D}_i^B(t), \hat{D}_i^S(t)$ are resulting values under the independent, stationary and randomized algorithm, η_{EE}^* is the optimal value of (16).

Proof. The detailed proof for Lemma 2 is omitted for brevity as a similar proof can be found in [20]. □

Based on Lemma 2 and boundedness assumptions (41), we demonstrate the queue stability and derive the tradeoff between energy efficiency and service delay in the following theorem.

Theorem 1. *Suppose problem (16) is feasible, $\mathbb{E}\{L(\mathbf{\Theta}(0))\} \le \infty$ and λ is strictly interior to network capacity region Λ, then, for all slot $t \in \mathcal{T}$ and any $V > 0, \epsilon > 0$, the proposed DTORA algorithm satisfies the following properties.*

(a) All queues are mean rate stable.

(b) The energy efficiency η_{EE} is upper bounded by

$$\eta_{EE} \le \eta_{EE}^* + \frac{B+C}{V(D_{min}^L + D_{min}^S)}, \tag{43}$$

where C is a constant gap between the solution obtained by DTORA algorithm and the infimum of the R.H.S of (22).

(c) The average sum queue length is upper bounded by

$$\bar{Q}^A \le \frac{B + C + V\left[\eta_{EE}^*(D_{max}^L + D_{max}^S) - E_A^{min}\right]}{\epsilon}. \tag{44}$$

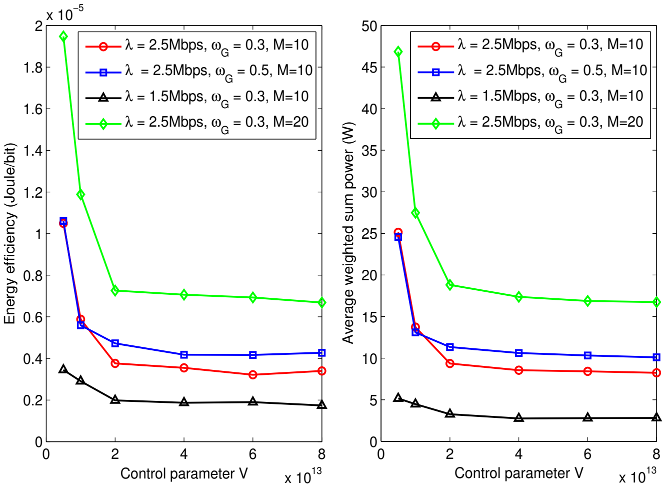
Proof. Please refer to Appendix B. The proof for the existence of constant gap C is given in Appendix C. □

VII. Simulation Results

In this section, we first evaluate the performance of the proposed DTORA algorithm with various system parameters, then three task offloading and resource allocation algorithms are compared with the DTORA in terms of energy efficiency and delay performance.

A. Simulation Setup

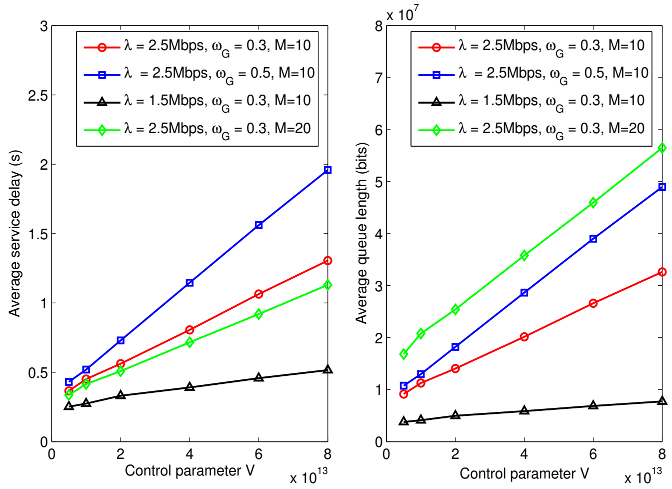
We simulate a dense C-RAN with 15 SRRHs and multiple SMDs randomly located over a 500×500 m^2 square area [16]. There are 4 subchannels and the bandwidth of each subchannel is 5 MHz [26]. The uplink path-loss model from SMDs to SRRHs is assumed to be $127 + 30\log_{10} d$ [16], where d is in kilometers. The small-scale channel power gains are modeled as i.i.d Rayleigh fading with unit mean value [40]. The maximum transmit power of SMDs is 0.5 W [22]. Furthermore, the total computation resource of edge cloud is 40 GHz and the maximum computation capacity of each SMD is randomly assigned from the set {0.5, 0.8, 1} GHz [25], [26]. When $W_s = 5$ MHz and each SMD has one antenna, β is 198 cycles/bit, which can be calculated according to [12]. We assume that each SMD has the same average task arrival rate λ. The weight factor $\omega_i = 1 - \omega_G$, for all $i \in \mathcal{M}$. The remaining simulation parameters are set as follows: $\kappa_i^L = \kappa^G = 10^{-27}$Watt $\cdot$ s^3/cycle^3, $\gamma_i(t) = 737.5$ cycles/bit, for all $i \in \mathcal{M}, t \in \mathcal{T}$, and $n_0 = -174$ dBm/Hz [22].



(a) Energy efficiency.

(b) Average weighted sum power.

Fig. 3: Energy efficiency / average weighted sum power v.s. control parameter V.



(a) Average service delay.

(b) Average queue length.

Fig. 4: Average service delay / average queue length v.s. control parameter V.

B. Impacts of System Parameters on Energy Efficiency and Service Delay

We first evaluate the energy efficiency and service delay versus system parameters including control parameter V, average task arrival rate λ, weight factor ω_G, the number of SMD M. Then, the varieties of computation resource allocation and queue length are illustrated.

Fig.3 illustrates the evolution of energy efficiency and average weighted sum power with control parameter V. It can be observed that η_{EE} converges to optimal energy efficiency value η^*_{EE} at a descent speed of $O[1/V]$, which conforms to Theorem 1. Given weight factor ω_G, energy efficiency and average weighted sum power increase as the growth of task arrival rate and the number of SMDs. This is due to the fact that larger task arrival rates and greater number of SMDs will require more computation resource to keep the queue stability, which leads to the increase of average sum power according to (6), (12), (13). Meanwhile, from (5), (11), we can find that the growth rate of energy consumption with computation capacity is higher than the growth rate of completed tasks. Hence energy efficiency also increases.

Fig.4 shows that the average service delay and average sum queue length increase linearly with the control parameter V, which is in conformity with Theorem 1. In addition, given task arrival rate λ and SMD number M, the average service delay and average sum queue length grow with the weight factor ω_G. A larger ω_G implies a greater weight of energy consumption at edge cloud, thus edge cloud reduces the computation resource assigned to SMDs, which leads to the increase of average sum queue length and average service delay. Intuitively, a lower task arrival rate results in the decrease of average service delay and average sum queue length. In addition, when network size increases from $M = 10$ to 20, the average queue length increases as a result of the larger aggregated task arrival rate. However, the increase of average service delay is small, which is because more local computation resource is available to process queue backlogs. These comparisons also indicate that the proposed scheme is scalable and can adapt to various network sizes and task arrival rates.

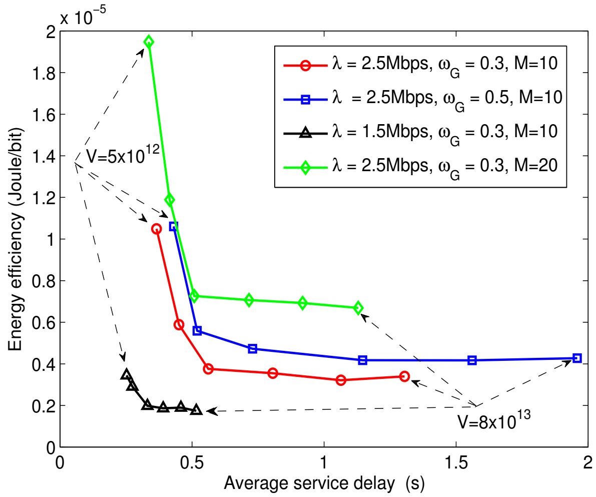
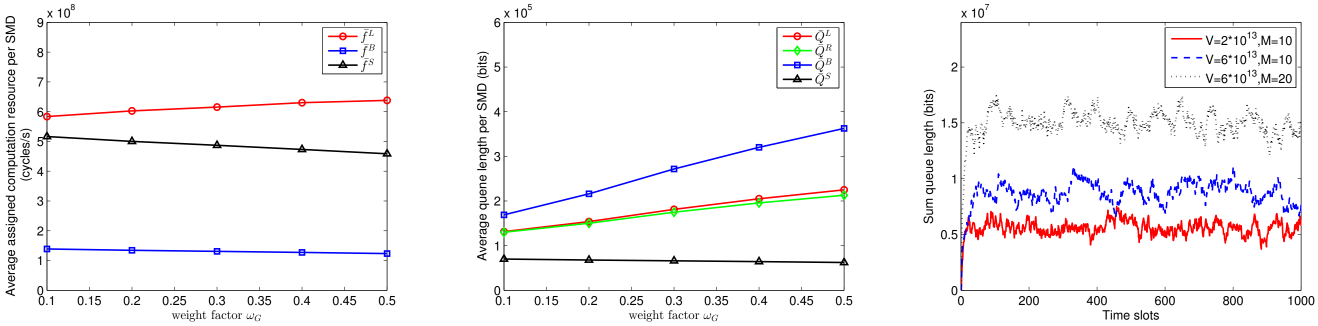


Fig. 5: Energy efficiency v.s. average service delay.

In Fig.5, it is observed that the energy efficiency decreases with the increase of service delay, which verifies the $[O(1/V), O(V)]$ tradeoff in Theorem 1. For a given service delay, when the task arrival rate, the number of SMD and the weigh factor ω_G increase, more local computation resource is scheduled to compensate the computation capacity of edge cloud. Because the weight of local energy consumption is larger than that of edge cloud, which leads to higher energy efficiency. The tradeoff relationship can provide guidelines for balancing energy and delay performance by properly tuning the control parameter V. Specifically, when the delay constraint of computation tasks is stringent, parameter V can be tuned to a small value to reduce the average service delay, otherwise a large V can be set to improve the energy efficiency performance. Thus we can adjust parameter V to satisfy different quality of service requirements for various computation tasks.



(a) Average computation resource per SMD v.s. ω_G.

(b) Average queue length per SMD v.s. ω_G.

(c) Sum queue length v.s. time slots.

Fig. 6: Average computation resource / average queue length per SMD v.s. ω_G, and sum queue length v.s. time slots.

To illustrate the adjustment of computation resource allocation with weight factor ω_G, we plot the varieties of average computation resource allocation ($\bar{f}^L, \bar{f}^B, \bar{f}^S$,) and average queue length ($\bar{Q}^L, \bar{Q}^R, \bar{Q}^B, \bar{Q}^S$,) per SMD with $V = 6\times10^{13}$, $\lambda = 2.5$ Mbps, $M = 10$ in Fig. 6(a) and Fig. 6(b). As can be seen, with the increase of ω_G, edge cloud decreases the CPU-cycle frequencies, and thus SMDs adjust local CPU-cycle frequencies to supplement the reduced computation resource at edge cloud. In addition, since the assigned computation resources at vBBU and vMES reduce with ω_G, more computation tasks are executed locally, leading to the increase of local queue length in Fig. 6(b). The larger value of ω_G is, the less computation resource allocated to SMD at vBBU, and the queue length at vBBU increases correspondingly. Meanwhile, the backlogs of computation tasks at uplink transmission buffers increase due to the decrease of computation speed at vBBU, which is supported by (2) and (30).

To better demonstrate the stability of task queues, we illustrate the sum queue length along with time slots in Fig. 6(c). The variations of sum queue length are compared under different control parameter V and SMD number M. It can be observed that the sum queue length of task buffers first increases rapidly and then stabilizes at fixed values. Furthermore, the stable queue length increases with control parameter V, that is because a larger V increases the weight of energy efficiency. Besides, the sum queue length grows with the number of SMDs, which results from the fact that more computation tasks are offloaded to edge cloud.

To illustrate the benefits of matching game-based subchannel allocation in Alg.1, we have compared the performances of energy efficiency and service delay between matching game-based subchannel allocation and burst force-based subchannel allocation in Fig.7. The burst force-based subchannel allocation method is an exhaustive method which can obtain the optimal subchannel allocation solution. From Fig.7, we can find the subchannel allocation solution obtained by matching game method is very close to the optimal solution obtained by burst force method. The computational complexity of matching game-based subchannel allocation method is $O(KM + M^2)$. However, the computational complexity of burst force method is $O(K^M)$, which is much higher than the matching game-based method.

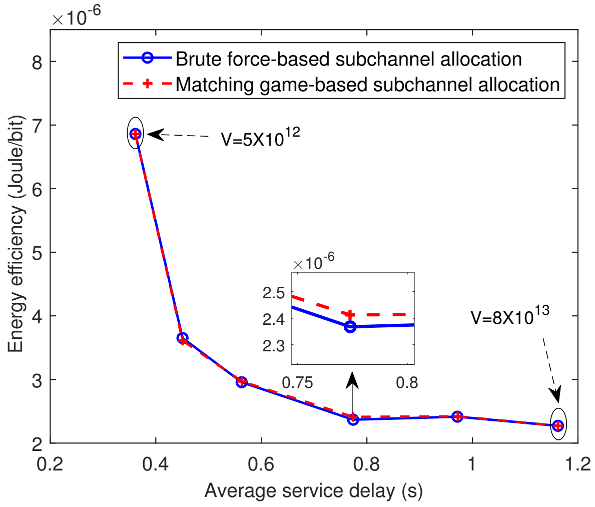


Fig. 7: Energy efficiency v.s. average service delay with $\lambda = 2.5$Mbps, $\omega_G = 0.3$, and $M = 7$.

To validate the assumption of high SINR conditions, we illustrate the empirical cumulative distribution functions (CDFs) of SINR Υ_{ij}^k after user association and subchannel allocation processes. The empirical CDF, also called empirical distribution function, is an unbiased estimator for the CDF. It can be obtained from the samples of the underlying CDF. In the simulations, we record the SINR values of all SMDs during 1000 time slots and calculate the empirical CDFs of SINR as shown in Fig.8. It can be observed that when the number of SMDs are 10, 30, and 50, the probabilities of ($\Upsilon_{ij}^k >= 20$ dB) are 0.997, 0.983, and 0.946 respectively, which are all close to 1. That is, the high SINR conditions hold in the proposed transmit power control algorithm.

In Fig.9, we evaluate the average running time of the proposed algorithm versus the number of subchannels and the number of SMDs. The DTORA algorithm is run on a computer equipped with Intel Core i5-8265, 3.9GHz processor and 16GB RAM memory. From Fig.9, it can be observed that the average running time increases linearly with the number of subchannels and increases with the number of SMDs at quadratic rates, which are consistent with the theoretical analysis. Therefore, the DTORA algorithm can be completed in polynomial time and it's able to be implemented online.

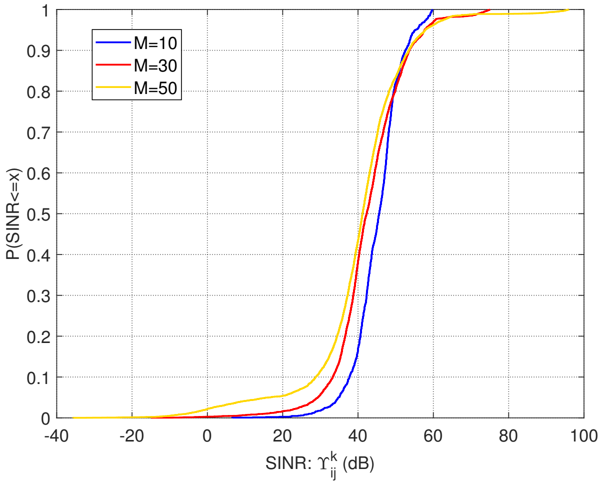


Fig. 8: The empirical CDFs of SINR Υ_{ij}^k after subchannel allocation with $V = 6 \times 10^{13}, \lambda = 1.5\text{Mbps}, \omega_G = 0.3$.

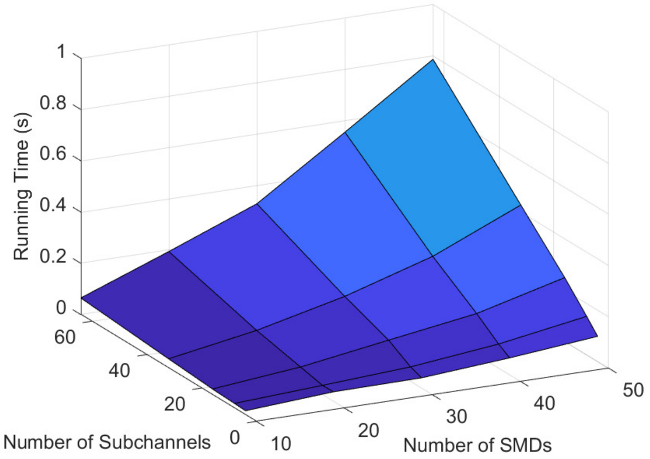


Fig. 9: The running time v.s. the number of SMDs and the number of subchannels.

C. Comparisons with Other Task Offloading and Resource Allocation Policies

In this section, we compare the proposed DTORA algorithm with three baseline policies: local computing only, computation resource allocation with the separated BBU pool and MEC servers and random offloading.

- Local computing only: All computation tasks are executed locally, i.e., $c_i(t) = 0, \forall i \in \mathcal{M}, t \in \mathcal{T}$. The local computation resource allocation is the same as DTORA.
- Separated BBU and MES: The task offloading decision, subchannel and transmit power allocation, and local computation resource scheduling are the same as DTORA algorithm. The BBU pool and mobile edge computing servers (MES) are deployed separately and can not share computation resources. The computation capacities of BBU pool and MES are $F_B^{\max}$ and $F_S^{\max}$ which satisfy $F_B^{\max} = F_S^{\max} = 20\text{GHz}$.
- Random offloading: Task offloading decision $c_i(t)$ is selected from uniform distribution on $[0,1], \forall i \in \mathcal{M}, \forall t \in \mathcal{T}$. Each SMD is allocated a subchannel from $[0, K]$ with a probability of $\frac{1}{K}$. The transmit power of SMD i is set from uniform distribution on $[0, P_i^{\max}]$. Local CPU-cycle frequencies are scheduled from uniform distribution on $[0, f_i^L(t)], \forall i \in \mathcal{M}, \forall t \in \mathcal{T}$. Computation resources allocated to vBBU and vMES are random and the sum of assigned computation resources does not exceed $F_G^{\max}$.

In Figs.10-11, we evaluate the performance of energy efficiency and service delay versus task arrival rate λ under four task offloading and resource allocation schemes. When $\lambda < 2$ Mbps, the computation resource at separated MEC servers is sufficient, thus energy efficiency and service delay are the same under DTORA and Separated BBU and MES algorithms. When $\lambda \geq 2$ Mbps, the computation resources of MES under Separated BBU and MES scheme are all in use and the computation frequency of MES reaches the maximum value $F_S^{\max}$, i.e., 20GHz. But in the proposed DTORA scheme, the computation resources assigned to vMES are elastic based on NFV and can be greater than $F_S^{\max}$, which decreases the queue backlogs and average server delay. Because more computation resources are scheduled to process offloaded tasks, the energy efficiency of DTORA algorithm is a little higher than the Separated BBU and MES scheme. In comparison with Separated BBU and MES algorithm, DTORA decreases the service delay by 56% at $\lambda = 3$ Mbps with a slight increase of energy efficiency. Hence when the arrival rate of computation tasks is lower than 2Mbps, DTORA and Separated BBU and MES schemes can be applied indiscriminately. When the task arrival rate is larger than 2 Mbps and the computation tasks are delay-sensitive such as automated driving services and real-time video analysis, DTORA scheme is more suitable than Separated BBU and MES scheme. When the task arrival rate is larger than 2 Mbps and the computation tasks are energy-sensitive such as data collection and analysis of energy-hungry IoT devices, Separated BBU and MES scheme can be applied.

The energy efficiency under Local computing only policy grows rapidly with task arrival rate λ and then converges to a constant, because the local computation resource is used up when $\lambda \geq 1.5$ Mbps, which leads to the increase of service delay. Compared with Local computing only policy, DTORA can significantly improve the performances about energy efficiency and service delay. The energy efficiency under Random offloading policy first decreases and then keeps stable at a constant. That is due to the fact that with the increase of task arrival rate, the computation resource is fully utilized. With the talk arrival rate $\lambda = 3$ Mbps, the proposed algorithm could reduce the energy consumption and average service delay by 59% and 57%, respectively, compared with the random offloading scheme.

To show the better energy and delay tradeoff of the DTORA scheme, we define the weighted sum of energy consumption and average delay as a unified metric. Inspired by the previous works [41], [42], the weighted sum of energy consumption and average delay is defined by $\varphi \bar{E}_A + (1-\varphi)\bar{Q}^A / \sum_{i=1}^{M} \lambda_i$, where φ is the weight factor of energy consumption. It's clear that DTORA algorithm outperforms Local computing only scheme and Random offloading scheme under various weight factors. Although DTORA scheme has a little higher

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2967502, IEEE Internet of Things Journal

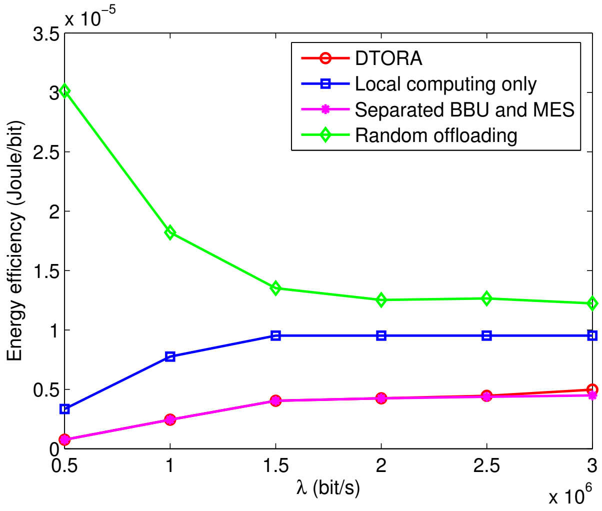


Fig. 10: Energy efficiency v.s. task arrival rate.

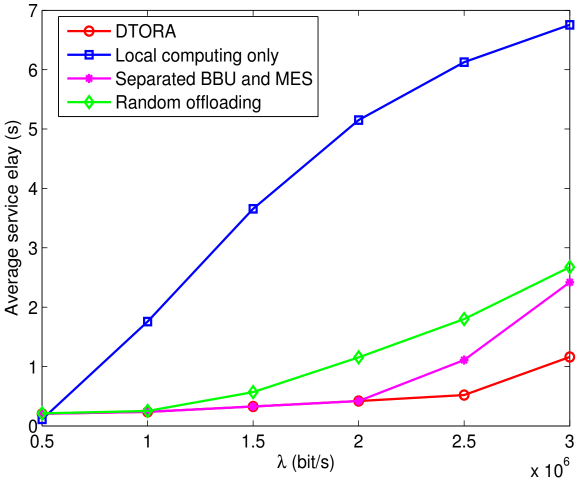


Fig. 11: Average service delay v.s. task arrival rate.

energy consumption than Separated BBU and MES scheme when $\lambda \geq 2$ Mbps, DTORA scheme can adjust the weight factor φ to obtain a better tradeoff of energy consumption and average delay. Fig.12 depicts the comparisons between DTORA scheme and baseline schemes by sampling φ as 0.8. It can be observed that DTORA scheme achieves lower weighted sum of energy consumption and average delay than Separated BBU and MES scheme when $\lambda \geq 2$ Mbps. For instance, when $\lambda = 3$ Mbps, the weighted sum of energy consumption and average delay of DTORA algorithm is 54% lower than that of Separated BBU and MES scheme.

VIII. Conclusion

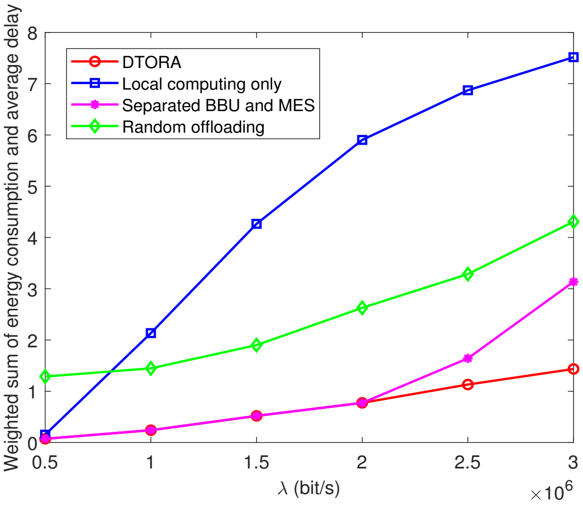


Fig. 12: The weighted sum of energy consumption and average delay v.s. task arrival rate

In this paper, we investigated the dynamic task offloading and resource allocation for MDC-RAN. An online semi-distributed optimization framework was proposed to handle task offloading decision, subchannel and transmit power allocation, computation resource scheduling at edge cloud and SMDs. We theoretically analyzed network stability and the tradeoff between network energy efficiency and average service delay. Extensive simulations demonstrated the impacts of system parameters on energy efficiency and service delay. Simulation results also corroborate the superior performance of proposed task offloading and resource management scheme. In specific, the proposed algorithm could reduce the energy consumption and average service delay by 59% and 57%, respectively, compared with the random offloading scheme with the talk arrival rate $\lambda = 3$ Mbps.

Appendix A
Proof for Lemma 1

By squaring both sides of queue evolution equation (2), the following inequality can be obtained:

$$\begin{aligned} Q_i^R(t+1)^2 &= (\max\{Q_i^R(t) - D_i^R(t), 0\})^2 + [c_i(t)A_i(t)]^2 \\ &\quad + 2\max\{Q_i^R(t) - D_i^R(t), 0\}c_i(t)A_i(t) \\ &\leq Q_i^R(t)^2 + [c_i(t)A_i(t)]^2 + D_i^R(t)^2 \\ &\quad + 2Q_i^R(t)[c_i(t)A_i(t) - D_i^R(t)]. \end{aligned} \tag{45}$$

Subtracting $Q_i^R(t)^2$ from both sides of (45), we have

$$\begin{aligned} Q_i^R(t+1)^2 - Q_i^R(t)^2 &\leq [c_i(t)A_i(t)]^2 + D_i^R(t)^2 \\ &\quad + 2Q_i^R(t)[c_i(t)A_i(t) - D_i^R(t)]. \end{aligned} \tag{46}$$

In the same way, we can derive:

$$\begin{aligned} Q_i^B(t+1)^2 - Q_i^B(t)^2 &\leq D_i^R(t)^2 + D_i^B(t)^2 \\ &\quad + 2Q_i^B(t)[D_i^R(t) - D_i^B(t)], \end{aligned} \tag{47}$$

$$\begin{aligned} Q_i^S(t+1)^2 - Q_i^S(t)^2 &\leq D_i^B(t)^2 + D_i^S(t)^2 \\ &\quad + 2Q_i^S(t)[D_i^B(t) - D_i^S(t)], \end{aligned} \tag{48}$$

$$\begin{aligned} Q_i^L(t+1)^2 - Q_i^L(t)^2 &\leq [(1 - c_i(t))A_i(t)]^2 + D_i^L(t)^2 \\ &\quad + 2Q_i^L(t)[(1 - c_i(t))A_i(t) - D_i^L(t)]. \end{aligned} \tag{49}$$

Summing up (46)−(49) over $i \in \mathcal{M}$ and dividing by 2 yield

$$\begin{aligned}
&L(\mathbf{\Theta}(t+1) - L(\mathbf{\Theta}(t)) \\
&\leq \sum_{i \in \mathcal{M}} \{D_i^R(t)^2 + D_i^B(t)^2 + \frac{1}{2}D_i^S(t)^2 + \frac{1}{2}D_i^L(t)^2 \\
&\quad + \frac{1}{2}[c_i(t)^2 + (1-c_i(t))^2]A_i(t)^2\} \\
&\quad - \sum_{i \in \mathcal{M}} \{Q_i^R(t)[D_i^R(t) - c_i(t)A_i(t)] \\
&\quad + Q_i^B(t)[D_i^B(t) - D_i^R(t)] + Q_i^S(t)[D_i^S(t) - D_i^B(t)] \\
&\quad + Q_i^L(t)[D_i^L(t) - (1-c_i(t))A_i(t)]\}. \qquad (50)
\end{aligned}$$

Taking conditional expectations to (50) and subtracting $V\mathbb{E}\{E_A(t) - \eta_{EE}(t)D_A(t)|\mathbf{\Theta}(t)\}$, we can obtain (22) in Lemma 1.

Appendix B
Proof for Theorem 1

(a) Due to the suboptimality of DTORA algorithm, the drift-plus-penalty function satisfies the following inequality according to *C-additive approximation* [20],

$$\begin{aligned}
&\Delta\mathbf{\Theta}(t) + V\mathbb{E}\{E_A(t) - \eta_{EE}(t)D_A(t))|\mathbf{\Theta}(t)\} \\
&\leq B + C + V\mathbb{E}\left\{\hat{E}_A(t) - \eta_{EE}(t)\hat{D}_A(t)|\mathbf{\Theta}(t)\right\} \\
&\quad - \mathbb{E}\{\sum_{i \in \mathcal{M}} Q_i^R(t)[\hat{D}_i^R(t) - \hat{c}_i(t)A_i(t)]|\mathbf{\Theta}(t)\} \\
&\quad - \mathbb{E}\{\sum_{i \in \mathcal{M}} Q_i^B(t)[\hat{D}_i^B(t) - \hat{D}_i^R(t)]|\mathbf{\Theta}(t)\} \\
&\quad - \mathbb{E}\{\sum_{i \in \mathcal{M}} Q_i^S(t)[\hat{D}_i^S(t) - \hat{D}_i^B(t)]|\mathbf{\Theta}(t)\} \\
&\quad - \mathbb{E}\{\sum_{i \in \mathcal{M}} Q_i^L(t)[\hat{D}_i^L(t) - (1-\hat{c}_i(t))A_i(t)]|\mathbf{\Theta}(t)\}, \\
&\qquad (51)
\end{aligned}$$

where $\hat{E}_A(t), \hat{c}_i(t), \hat{D}_i^R(t), \hat{D}_i^B(t), \hat{D}_i^S(t), \hat{D}_i^L(t)$ are corresponding values under any alternative task offloading and resource allocation algorithm referred to Lemma 2. Substituting (42) into (51), we have

$$\begin{aligned}
&\Delta\mathbf{\Theta}(t) + V\mathbb{E}\{E_A(t) - \eta_{EE}(t)D_A(t))|\mathbf{\Theta}(t)\} \\
&\leq B + C + V\mathbb{E}\left\{\hat{D}_A(t)\right\}(\eta_{EE}^* + \delta) \\
&\quad - V\mathbb{E}\left\{\eta_{EE}(t)\hat{D}_A(t)\right\} \\
&\quad - \epsilon\sum_{i \in \mathcal{M}}[Q_i^R(t) + Q_i^B(t) + Q_i^S(t) + Q_i^L(t)]. \qquad (52)
\end{aligned}$$

Since $Q_i^L(t) \geq 0, Q_i^R(t) \geq 0, Q_i^B(t) \geq 0, Q_i^S(t) \geq 0$, taking a limit as $\delta \rightarrow 0$, summing (52) over $t \in \{0, 1, \cdots, \tau-1\}$ and based on the law of telescoping sums, we obtain

$$\begin{aligned}
&\mathbb{E}\{L(\mathbf{\Theta}(\tau))\} - \mathbb{E}\{L(\mathbf{\Theta}(0))\} \\
&\quad + V\sum_{t=0}^{\tau-1}\mathbb{E}\{E_A(t) - \eta_{EE}(t)D_A(t)|\mathbf{\Theta}(t)\} \\
&\leq \tau(B+C) + V\tau\eta_{EE}^*\mathbb{E}\left\{\hat{D}_A(t)\right\} \\
&\quad - V\mathbb{E}\left\{\hat{D}_A(t)\right\}\sum_{t=0}^{\tau-1}\eta_{EE}(t). \qquad (53)
\end{aligned}$$

According to (19) and (41), rearranging (53), we have

$$\begin{aligned}
\mathbb{E}\{[Q_i^L(\tau)]^2\} &\leq 2\tau[B + C + V\eta_{EE}^*(D_{\text{max}}^L + D_{\text{max}}^S)] \\
&\quad - 2V\tau\eta_{EE}^{\text{min}}(D_{\text{min}}^L + D_{\text{min}}^S) \\
&\quad - 2V\tau[E_A^{\text{min}} - \eta_{EE}^{\text{max}}(D_{\text{max}}^L + D_{\text{max}}^S)] \\
&\quad + 2\mathbb{E}\{L(\mathbf{\Theta}(0))\}. \qquad (54)
\end{aligned}$$

Based on the fact that $\mathbb{E}\{[Q_i^L(\tau)]^2\} \geq \left(\mathbb{E}\{|Q_i^L(\tau)|\}\right)^2$, dividing by τ, and take a limit as $\tau \rightarrow \infty$, we obtain

$$\lim_{\tau \rightarrow +\infty}\frac{\mathbb{E}\{|Q_i^L(\tau)|\}}{\tau} = 0. \qquad (55)$$

Thus queues $Q_i^L(t)$ are mean rate stable. In a similar way, we can prove that queues $Q_i^R(t), Q_i^B(t), Q_i^S(t)$ are also mean rate stable.

(b) Due to $\mathbb{E}\{L(\mathbf{\Theta}(\tau))\} \geq 0$, divided by $V\tau$, inequality (53) can be rewritten as

$$\begin{aligned}
&\frac{1}{\tau}\sum_{t=0}^{\tau-1}\mathbb{E}\{E_A(t) - \eta_{EE}(t)D_A(t)\} \\
&\leq \frac{(B+C)}{V} + \eta_{EE}^*\mathbb{E}\left\{\hat{D}_A(t)\right\} \\
&\quad - \frac{\sum_{t=0}^{\tau-1}\eta_{EE}(t)}{\tau}\mathbb{E}\left\{\hat{D}_A(t)\right\} + \frac{\mathbb{E}\{L(\mathbf{\Theta}(0))\}}{V\tau}. \qquad (56)
\end{aligned}$$

Taking a limit as $\tau \rightarrow \infty$,

$$\frac{(B+C)}{V} + (\eta_{EE}^* - \eta_{EE})\mathbb{E}\left\{\hat{D}_A(t)\right\} \geq 0. \qquad (57)$$

Rearranging (57) and referring to boundedness assumptions (41), we have

$$\eta_{EE} \leq \eta_{EE}^* + \frac{B+C}{V(D_{\text{min}}^L + D_{\text{min}}^S)}. \qquad (58)$$

(c) Retaining the last term of (52), taking a limit as $\delta \rightarrow 0$ and summing over $t \in \{0, 1, \cdots, \tau-1\}$ yield

$$\begin{aligned}
&\mathbb{E}\{L(\mathbf{\Theta}(\tau))\} - \mathbb{E}\{L(\mathbf{\Theta}(0))\} \\
&\quad + V\sum_{t=0}^{\tau-1}\mathbb{E}\{E_A(t) - \eta_{EE}(t)D_A(t)\} \\
&\leq \tau(B+C) + V\tau\eta_{EE}^*\mathbb{E}\left\{\hat{D}_A(t)\right\} \\
&\quad - V\mathbb{E}\left\{\hat{D}_A(t)\right\}\sum_{t=0}^{\tau-1}\eta_{EE}(t) \\
&\quad - \epsilon\sum_{t=0}^{\tau-1}\sum_{i=1}^{M}\mathbb{E}\left\{Q_i^R(t) + Q_i^B(t) + Q_i^S(t) + Q_i^L(t)\right\}. \\
&\qquad (59)
\end{aligned}$$

Dividing (59) by $\epsilon\tau$ and taking a limit as $\tau \to \infty$, we obtain

$$\lim_{\tau\to+\infty}\frac{1}{\tau}\sum_{t=0}^{\tau-1}\sum_{i=1}^{M}\mathbb{E}\left\{Q_i^R(t)+Q_i^B(t)+Q_i^S(t)+Q_i^L(t)\right\}$$

$$\leq\frac{1}{\epsilon}\left(B+C+V\eta_{EE}^{*}(D_{\max}^{L}+D_{\max}^{S})\right)-\frac{VE_A^{\min}}{\epsilon} \quad (60)$$

Thus, the average sum queue length satisfies

$$\bar{Q}^A\leq\frac{B+C+V\left[\eta_{EE}^{*}(D_{\max}^{L}+D_{\max}^{S})-E_A^{\min}\right]}{\epsilon}. \quad (61)$$

Appendix C
Proof for the existence of Constant gap C in Theorm 1

To proof the existence of the constant gap C and derive its value, we first analyze the optimality of the proposed DTORA algorithm. The solution obtained by DTORA algorithm is derived by solving four subproblems.

The subproblem 1 and subproblem 4 are task offloading decision problem and local computation resource scheduling problem. The optimal solutions of these two subproblems are solved by (24) and (40) respectively.

The subproblem 3 is computation resource allocation problem at edge cloud. Given the number of iterations L_{sp3} of subgradient method, the solution of subproblem 3 can converge to a suboptimal value [37]. Since the step sizes in subproblem 3 satisfy square summable but not summable rule, the gap $\Delta_{L_{\text{sp3}}}$ between the suboptimal solution to the optimal solution satisfies $\Delta_{L_{\text{sp3}}}\leq\frac{\|\mathbf{f}^0(t)-\mathbf{f}^*(t)\|_2^2+G_{\text{sp3}}^2\sum_{l=1}^{\infty}\mu_l^2(t)}{2\sum_{l=1}^{L_{\text{sp3}}}\mu_l(t)}$, where $\mathbf{f}^0(t)=[\mathbf{f}^{B_0}(t)\quad\mathbf{f}^{S_0}(t)]$ is the initial iterative vector, $\mathbf{f}^*(t)=[\mathbf{f}^{B^*}(t)\quad\mathbf{f}^{S^*}(t)]$ is the optimal solution vector of subproblem 3, G_{sp3} is an upper bound of the norm of $F_G^{\max}-\sum_{i\in\mathcal{M}}[f_i^B(t)+f_i^S(t)]$ for all iterations, $\|\cdot\|_2$ is the operator of euclidean norm. A similar convergence proof can be found in [37].

Similar to the solution of subproblem 3, the solution of transmit power control in subproblem 2 can also coverage to a suboptimal value. Given the number of iterations $L_{\text{sp2.2}}^k$ of on subchannel k, the gap $\Delta_{L_{\text{sp2.2}}}$ between the suboptimal solution to the optimal solution satisfies $\Delta_{L_{\text{sp2.2}}}\leq\sum_{k=1}^{K}\sum_{i=1}^{|\mathcal{M}^k|}\frac{[p_i^{T_0}(t)-p_i^{T^*}(t)]+G_{\text{sp2.2}}^2\sum_{l=1}^{\infty}[\nu_i^l(t)]^2}{2\sum_{l=1}^{L_{\text{sp2.2}}^k}\nu_i^l(t)}$, where $p_i^{T_0}(t)$ is the initial iterative value, $p_i^{T^*}(t)$ is the optimal value of SMD i, $G_{\text{sp2.2}}$ is an upper bound of the norm of $P_i^{\max}-e^{\tilde{p}_i^T(t)}$ for all iterations.

Then, we prove the local optimality of proposed matching game-based subchannel allocation algorithm. Given a matching Ω, we assume (i,i') is a swap-blocking pair and $\Omega(i)=k$, $\Omega(i')=k'$. The set of SMDs that are assigned subchannel k in matching Ω is $\mathcal{M}_\Omega^k$. Denote the transmit power set of SMDs on subchannel k as $\left(\mathbf{P}^k_{-(i,i')}(t),p_i^k(t),p_{i'}^k(t)\right)$. For each SMD $m\in\mathcal{M}_\Omega^k\backslash\{i\}$, the first order Taylor expansion of $R_m^k\left(\mathbf{P}^k_{-(i,i')}(t),p_i^k(t),p_{i'}^k(t)\right)$ about $\left(p_i^T(t),p_{i'}^T(t)\right)=(p_i(t),0)$ can be written as

$$\begin{aligned}&\hat{R}_m^k\left(\mathbf{P}^k_{-(i,i')}(t),p_i^k(t),p_{i'}^k(t)\right)\\&=R_m^k(\mathbf{P}(t))-\sigma_m^k h_{mi}^k(p_i^k(t)-p_i^T(t))\\&\quad-\sigma_m^k h_{mi'}^k(p_{i'}^k(t)-p_{i'}^T(t))\\&=R_m^k(\mathbf{P}(t))-\sigma_m^k h_{mi}^k(p_i^k(t)-p_i(t))\\&\quad-\sigma_m^k h_{mi'}^k p_{i'}^k(t).\end{aligned}$$

Similarly, for each SMD $m\in\mathcal{M}_\Omega^{k'}\backslash\{i'\}$, the first order Taylor expansion of $R_m^{k'}\left(\mathbf{P}^{k'}_{-(i,i')}(t),p_i^{k'}(t),p_{i'}^{k'}(t)\right)$ about $\left(p_i^T(t),p_{i'}^T(t)\right)=(0,p_{i'})$ can be written as

$$\begin{aligned}&\hat{R}_m^{k'}\left(\mathbf{P}^{k'}_{-(i,i')}(t),p_i^{k'}(t),p_{i'}^{k'}(t)\right)\\&=R_m^{k'}(\mathbf{P}(t))-\sigma_m^{k'}h_{mi}^{k'}(p_i^{k'}(t)-p_i^T(t))\\&\quad-\sigma_m^{k'}h_{mi'}^{k'}(p_{i'}^{k'}(t)-p_{i'}^T(t))\\&=R_m^{k'}(\mathbf{P}(t))-\sigma_m^{k'}h_{mi}^{k'}p_i^{k'}(t)-\sigma_m^{k'}h_{mi'}^{k'}(p_{i'}^{k'}(t)-p_{i'}^T(t)).\end{aligned}$$

Then, we have

$$\begin{aligned}&\sum_{m\in\mathcal{M}_\Omega^k\cup\mathcal{M}_\Omega^{k'}}[Q_m^R(t)-Q_m^B(t)]R_m^k(\mathbf{P}(t))\\&=\sum_{m\in\mathcal{M}_\Omega^k\backslash\{i\}}[Q_m^R(t)-Q_m^B(t)]\hat{R}_m^k\left(\mathbf{P}^k_{-(i,i')}(t),p_i(t),0\right)\\&\quad+[Q_i^R(t)-Q_i^B(t)]R_i^k(\mathbf{P}(t))\\&\quad+\sum_{m\in\mathcal{M}_\Omega^{k'}\backslash\{i'\}}[Q_m^R(t)-Q_m^B(t)]\hat{R}_m^{k'}\left(\mathbf{P}^{k'}_{-(i,i')}(t),0,p_{i'}(t)\right)\\&\quad+[Q_{i'}^R(t)-Q_{i'}^B(t)]R_{i'}^{k'}(\mathbf{P}(t))\\&\overset{(a)}{<}\sum_{m\in\mathcal{M}_\Omega^k\backslash\{i\}}[Q_m^R(t)-Q_m^B(t)]\hat{R}_m^k\left(\mathbf{P}^k_{-(i,i')}(t),0,p_{i'}(t)\right)\\&\quad+[Q_{i'}^R(t)-Q_{i'}^B(t)]R_{i'}^k\left(\mathbf{P}^k_{-(i,i')}(t),0,p_{i'}(t)\right)\\&\quad+\sum_{m\in\mathcal{M}_\Omega^{k'}\backslash\{i'\}}[Q_m^R(t)-Q_m^B(t)]\hat{R}_m^{k'}\left(\mathbf{P}^{k'}_{-(i,i')}(t),p_i(t),0\right)\\&\quad+[Q_i^R(t)-Q_i^B(t)]R_i^{k'}\left(\mathbf{P}^{k'}_{-(i,i')}(t),p_i(t),0\right)\\&=\sum_{m\in\mathcal{M}^k_{\Omega_i^{i'}}\cup\mathcal{M}^{k'}_{\Omega_i^{i'}}}[Q_m^R(t)-Q_m^B(t)]R_m^k(\mathbf{P}(t)).\end{aligned}$$

The inequality (a) holds because the preference functions $\psi_k^{SA}(i)$ and $\psi_i^{SA}(k)$ increases after a swap matching. Hence the objective function of subproblem 2 monotonically increases after each swap matching. Since the number of SMDs is finite, the stable matching can be obtained after finite swaps. Thus the matching game-based subchannel allocation can at least converge to a local optimal solution. The gap between the local optimal solution and global optimal solution is denoted as $\Delta_{L_{\text{sp2.1}}}$.

The solutions of subproblems 1 and 4 are optimal. The solutions of subproblem 3 and transmit power control in subproblem 2 are suboptimal. Hence there exists a constant gap $C=\Delta_{L_{\text{sp2.1}}}+\Delta_{L_{\text{sp2.2}}}+\Delta_{L_{\text{sp3}}}$ between the solution obtained by DTORA algorithm and the infimum of the R.H.S of (22).

References

[1] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 14–19, Aug. 2018.

[2] Y. Teng, M. Liu, F. R. Yu, V. C. M. Leung, M. Song, and Y. Zhang, "Resource allocation for ultra-dense networks: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, pp. 1–1, 2019.

[3] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas in Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.

[4] L. Yang, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2762–2773, Dec. 2018.

[5] F. Sun, F. Hou, N. Cheng, M. Wang, H. Zhou, L. Gui, and X. Shen, "Cooperative task scheduling for computation offloading in vehicular cloud," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11 049–11 061, Nov. 2018.

[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quarter 2017.

[7] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.

[8] L. Zhao, J. Wang, J. Liu, and N. Kato, "Routing for crowd management in smart cities: A deep reinforcement learning perspective," *IEEE Commun. Mag.*, vol. 57, no. 4, pp. 88–93, Apr. 2019.

[9] M. Awais, A. Ahmed, S. A. Ali, M. Naeem, W. Ejaz, and A. Anpalagan, "Resource management in multicloud IoT radio access network," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3014–3023, Apr. 2019.

[10] H. Hawilo, M. Jammal, and A. Shami, "Network function virtualization-aware orchestrator for service function chaining placement in the cloud," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 643–655, Mar. 2019.

[11] ETSI, "Cloud RAN and MEC: A perfect pairing," White Paper, Feb. 2018. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp23_MEC_and_CRAN_ed1_FINAL.pdf

[12] K. Wang, K. Yang, H. Chen, and L. Zhang, "Computation diversity in emerging networking paradigms," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 88–94, Feb. 2017.

[13] X. Wang, K. Wang, S. Wu, S. Di, H. Jin, K. Yang, and S. Ou, "Dynamic resource scheduling in mobile edge cloud with cloud radio access network," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 11, pp. 2429–2445, Nov. 2018.

[14] L. Pu, X. Chen, G. Mao, Q. Xie, and J. Xu, "Chimera: An energy-efficient and deadline-aware hybrid edge computing framework for vehicular crowdsensing applications," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 84–99, Feb. 2019.

[15] L. P. Qian, Y. Wu, B. Ji, L. Huang, and D. H. K. Tsang, "HybridIoT: Integration of hierarchical multiple access and computation offloading for IoT-based smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 6–13, Mar. 2019.

[16] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[17] L. Zhang, K. Wang, D. Xuan, and K. Yang, "Optimal task allocation in near-far computing enhanced c-ran for wireless big data processing," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 50–55, Feb. 2018.

[18] Z. Jian, W. Muqing, and Z. Min, "Joint computation offloading and resource allocation in C-RAN with mec based on spectrum efficiency," *IEEE Access*, vol. 7, pp. 79 056–79 068, 2019.

[19] J. Zhang, X. Hu, Z. Ning, E. C. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[20] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[21] M. Peng, Y. Yu, H. Xiang, and H. V. Poor, "Energy-efficient resource allocation optimization for multimedia heterogeneous cloud radio access networks," *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 879–892, May 2016.

[22] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[23] S. Mao, S. Leng, K. Yang, Q. Zhao, and M. Liu, "Energy efficiency and delay tradeoff in multi-user wireless powered mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[24] R. Duan, J. Wang, C. Jiang, Y. Ren, and L. Hanzo, "The transmit-energy vs computation-delay trade-off in gateway-selection for heterogenous cloud aided multi-uav systems," *IEEE Trans. Commun.*, vol. 67, no. 4, pp. 3026–3039, Apr. 2019.

[25] 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.

[26] Q. Pham, T. Leanh, N. H. Tran, B. J. Park, and C. S. Hong, "Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach," *IEEE Access*, vol. 6, pp. 75 868–75 885, 2018.

[27] M. K. Elhattab, M. M. Elmesalawy, T. Ismail, H. H. Esmat, M. M. Abdelhakam, and H. Selmy, "A matching game for device association and resource allocation in heterogeneous cloud radio access networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1664–1667, Aug. 2018.

[28] J. Tang, W. P. Tay, and T. Q. S. Quek, "Cross-layer resource allocation with elastic service scaling in cloud radio access network," *IEEE Trans. Wireless Commun.*, vol. 14, no. 9, pp. 5068–5081, Sep. 2015.

[29] R. Serfozo, *Introduction to Stochastic Networks*. Springer-Verlag, 1999.

[30] W. Dinkelbach, "On nonlinear fractional programming," *Management Science*, vol. 13, no. 7, pp. 492–498, 1967.

[31] Y. Li, M. Sheng, Y. Shi, X. Ma, and W. Jiao, "Energy efficiency and delay tradeoff for time-varying and interference-free wireless networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 5921–5931, Nov. 2014.

[32] K. Wang, W. Zhou, and S. Mao, "Energy efficient joint resource scheduling for delay-aware traffic in cloud-ran," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[33] Y. Yuan, T. Yang, H. Feng, and B. Hu, "An iterative matching-stackelberg game model for channel-power allocation in D2D underlaid cellular networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7456–7471, Nov. 2018.

[34] E. Bodine-Baron, C. Lee, A. Chong, B. Hassibi, and A. Wierman, "Peer effects and stability in matching markets," in *Proc. 4th Symp. Algorithmic Game Theory (SAGT)*, G. Persiano, Ed., Oct. 2011, pp. 117–129.

[35] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[36] M. Chiang, "Geometric programming for communication systems," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 1–2, pp. 1–154, 2005.

[37] L. V. Stephen Boyd, *Convex Optimization*. Cambridge Univ. Press, 2004.

[38] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*,

vol. 2004, pp. 2004–2005, 2003.

[39] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5256–5268, Jun. 2017.

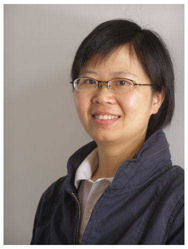
[40] C. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.

[41] Q. Pham and W. Hwang, "Fairness-aware spectral and energy efficiency in spectrum-sharing wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10 207–10 219, Nov. 2017.

[42] X. Zhang and J. Tang, "Power-delay tradeoff over wireless networks," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3673–3684, Sep. 2013.

Qi Zhang received the B.Eng. degree from the School of Electronics and Information, Northwestern Polytechnical University (NPU), Xian, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China. His research interests include mobile edge computing, resource management and optimization.

Lin Gui (M'08) received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2002. Since 2002, she has been at the Institute of Wireless Communication Technology, Shanghai Jiao Tong University, Shanghai, China, where she is currently a Professor. Her current research interests include HDTV and wireless communications.



Fen Hou is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Macau. She received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2008. She worked as a postdoctoral fellow in the Electrical and Computer Engineering at the University of Waterloo and in the Department of Information Engineering at the Chinese University of Hong Kong from 2008 to 2009 and from 2009 to 2011, respectively. Her research interests include resource allocation and scheduling in broadband wireless networks, protocol design and QoS provisioning for multimedia communications in broadband wireless networks, Mechanism design and optimal user behavior in mobile crowd sensing networks and mobile data offloading. She is the recipient of IEEE GLOBECOM Best Paper Award in 2010 and the Distinguished Service Award in IEEE MMTC in 2011. Dr. Fen Hou served as the co-chair in ICCS 2014 Special Session on Economic Theory and Communication Networks, INFOCOM 2014 Workshop on Green Cognitive Communications and Computing Networks (GCCCN), IEEE Globecom Workshop on Cloud Computing System, Networks, and Application (CCSNA) 2013 and 2014, ICCC 2015 Selected Topics in Communications Symposium, and ICC 2016 Communication Software Services and Multimedia Application Symposium, respectively. She currently serves as the Director of Award Board in IEEE ComSoc Multimedia Communications Technical Committee. She also serves as an Associate Editor of IET Communications.

Jiacheng Chen received his Ph.D. degree in information and communications engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018. From Dec. 2015 to Dec. 2016, he was a visiting scholar at BBCR group, University of Waterloo, Canada. Currently, he is an assistant researcher in Peng Cheng Laboratory, Shenzhen, China. His research interests include future network design, 5G/6G network, and resource management.

Shichao Zhu received his B.E. degree from the School of Aeronautics, Northwestern Polytechnical University, Xian, China, in 2016. He is now pursuing his Ph.D degree in the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include data and computation offloading in UAV networks and space-air integrated networks, and the application of AI in wireless networks.

Feng Tian received the B.E. degree in electronic engineering from Northwestern Polytechnical University, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, in 2012 and 2017. He is now an associate professor at Shanghai Engineering Center for Microsatellites, Shanghai, China. His research interests include wireless networking, wireless communications, and satellite communication.