Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 6098786, 17 pages
https://doi.org/10.1155/2020/6098786

Research Article

Energy-Efficient Mobile Edge Computing: Three-Tier Computing under Heterogeneous Networks

Yongsheng Pei, Zhangyou Peng, Zhenling Wang, and Haojia Wang

The Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Zhangyou Peng; zypeng@shu.edu.cn

Received 26 December 2019; Revised 17 February 2020; Accepted 18 February 2020; Published 2 April 2020

Academic Editor: Manuel Fernandez-Veiga

Mobile edge computing (MEC) is a promising technique to meet the demands of computing-intensive and delay-sensitive applications by providing computation and storage capabilities in close proximity to mobile users. In this paper, we study energy-efficient resource allocation (EERA) schemes for hierarchical MEC architecture in heterogeneous networks. In this architecture, both small base station (SBS) and macro base station (MBS) are equipped with MEC servers and help smart mobile devices (SMDs) to perform tasks. Each task can be partitioned into three parts. The SMD, SBS, and MBS each perform a part of the task and form a three-tier computing structure. Based on this computing structure, an optimization problem is formulated to minimize the energy consumption of all SMDs subject to the latency constraints, where radio and computation resources are considered jointly. Then, an EERA mechanism based on the variable substitution technique is designed to calculate the optimal workload distribution, edge computation capability allocation, and SMDs' transmit power. Finally, numerical simulation results demonstrate the energy efficiency improvement of the proposed EERA mechanism over the baseline schemes.

1. Introduction

Driven by the rapid development of Internet of Things and mobile Internet, many novel applications are emerging [1]. However, most of these applications are computing-intensive and delay-sensitive, e.g., augmented reality, face recognition, and healthcare [2]. Running these applications locally is very challenging for smart mobile devices (SMDs) when ensuring users' quality of experience (QoE) because of the limited resources of SMDs. How to complete the applications while guaranteeing users' QoE becomes the focus of academic and industrial communities. Mobile edge computing (MEC) is a promising technique to solve this problem, which endows the radio access network with computation and storage capabilities. In order to improve users' QoE, MEC helps SMDs complete applications by performing some tasks in the edge nodes of networks, which reduces the latency and energy consumption of task execution thanks to the close proximity of edge nodes to SMDs [3, 4].

Extensive research on MEC has been conducted from many perspectives, e.g., single-server MEC models and multiserver MEC models. Regarding the single-server MEC models, much work has been done, e.g., single-user models [5–9] and multiuser models [10–15]. For a single-user MEC model, the authors in [5] considered a binary computation offloading model and derived a data consumption rate threshold that decided to offload the whole task or execute the entire task locally. Based on that work, for further reducing the energy consumption of SMDs, partial offloading was introduced into the single-user model. The task was partitioned into two parts, one of which was offloaded [6, 7]. Considering the stochastic arrival of tasks, the optimal task scheduling policy was derived to minimize the weighted sum of the energy consumption and latency [8]. In addition, the energy harvesting technique was incorporated into the MEC model and the Lyapunov optimization-based dynamic computation offloading algorithm was proposed in [9]. For a multiuser MEC model, to satisfy the requirements of as many users as possible in a channel environment with wireless interference, the multiuser offloading system was formulated as a game and analyzed to admit a Nash equilibrium [10]. Considering inelastic computation tasks and non-negligible task execution durations, the authors in [11] proposed an energy-efficient resource allocation schemes. To deal with the arbitrary arrival of tasks in multiuser MEC system, tasks scheduling techniques were utilized in [12, 13]. To reduce the redundant execution of the same tasks and minimize the energy consumption, the storage resource of the base station was utilized in [14]. For further improving users' QoE, wireless power transfer was added into the multiuser MEC model and an access point energy minimization problem was formulated [15].

Regarding the multiserver MEC models, many edge cloud architectures are emerging, e.g., flat edge cloud architectures [16–19] and hierarchical edge cloud architectures [20–22]. In the flat edge cloud architectures, MEC servers are located at the same tier. In the hierarchical edge cloud architectures, MEC servers are located at different tiers. And MEC servers in different tiers have distinct computation and storage capabilities [3, 23]. For a flat edge cloud architecture, geography information of SMDs and MEC servers was used to reduce the task execution delays in [16]. Considering maximizing the revenue of service providers, resources from different service providers were centralized to create a resource pool and the revenue was allocated by using core and Shapley values [17]. To minimize the communication latency, a cloudlet selection model based on mixed integer linear programming was developed in [18]. Furthermore, by utilizing the idle computing resources of vehicles, the authors in [19] proposed a decentralized framework named Autonomous Vehicular Edge to increase the computational capabilities of vehicles. For a hierarchical edge cloud architecture, a three-tier MEC model was built on the basis of LTE-advanced mobile backhaul network [20]. For improving the cost efficiency of network operators, the authors in [21] took the cost disparity of the edge tiers into account. Under a three-tier MEC model, the Stackelberg game was used to allocate the limited computing resources of edge severs to the data service subscribers [22].

Combined with heterogeneous networks, the hierarchical MEC was further studied. The small base station (SBS) and macro base station (MBS) are equipped with MEC servers to serve SMDs. Particularly, in [24], offloading decisions and radio resource were optimized jointly for minimizing the system energy cost. Then, the framework was developed further. SBSs were endowed with computing capabilities. And a resource allocation problem for minimizing the energy consumption of mobile users and MEC servers was formulated [25]. Based on the heterogeneous network powered by hybrid energy, user association and resource allocation were optimized for maximizing the network utility [26]. Considering the variability of mobile devices' capabilities and user preferences, offloading decisions and resource allocation were optimized for maximizing system utility [27]. In addition, a novel information-centric heterogeneous network framework was designed and a virtual resource allocation problem was formulated in [28].

1.1. Motivations and Contributions. Hierarchical architectures of edge servers have an advantage over flat architectures in serving the peak loads [23, 29]. In addition, under the three-tier MEC architectures, previous studies focused on the system construction [20–22] and maximization of the system utility [26–28]. However, it is also important how to allocate computation and communication resource energy efficiently under a three-tier MEC architecture to improve users' QoE. In this paper, we investigate a multiuser three-tier computing model under heterogeneous networks. The SBS integrated with relatively small computation capability and MBS integrated with great computation capability jointly execute tasks. Based on this hierarchical MEC model, an energy-efficient resource allocation (EERA) scheme is proposed. In EERA, the computation and radio resources are optimized jointly for minimizing the energy consumption of all SMDs. The main contributions of this paper are summarized as follows:

(1) Based on heterogeneous networks, we establish a three-tier computing model, including local computing, SBS computing, and MBS computing. An energy-efficient optimization problem is formulated. Workload placement strategy, transmit power, and computation capability allocation are optimized to minimize SMDs' energy consumption under task delay constraints.

(2) We propose an EERA scheme based on the variable substitution technique. In this scheme, the optimal workload distribution and computation capability allocation are first obtained. Then, the optimal SMDs' transmit power is derived through the variable substitution.

(3) Numerical simulation experiments are conducted. Simulation results are presented to validate that EERA outperforms other baseline schemes and effectively reduces the SMDs' energy consumption.

1.2. Organization. The rest of this paper is organized as follows. In Section 2, the three-tier computing model is presented and the energy-efficient optimization problem is formulated. In Section 3, EERA based on the variable substitution technique is proposed, where workload distribution in three-tier, computation capability allocation from SBS and SMDs' transmit power are optimized jointly to minimize SMDs' energy consumption. Numerical results are provided in Section 4, and conclusions are presented in Section 5.

2. System Model and Problem Formulation

As shown in Figure 1, SBS and MBS are equipped with MEC servers and help SMDs perform tasks. SMDs, SBS, and MBS execute tasks together and establish a three-tier computing architecture. In the first tier, there is K SMDs and the set of SMDs is denoted as $\mathscr{K} = \{1, 2, \cdots, K\}$. The processing capability of k-SMD ($k \in \mathscr{K}$) is denoted as $f_{k,l}$ cycles/s. In the second tier, the SBS has the limited computation capability denoted as F cycles/s. In the third tier, we assume that the MBS has infinite computational resources and its execution latency is negligible [9, 30]. In addition, the backhaul link time delay between SBS and MBS is proportional to the transfer data size and the proportion coefficient is denoted as ϕ [24]. We assume that each user has one SMD and each

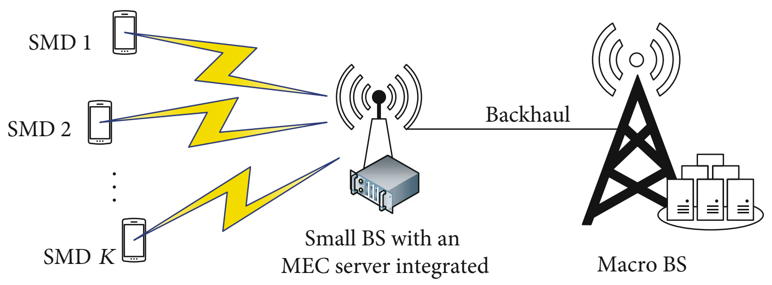


Figure 1: Multiuser task execution in three-tier computing architecture.

SMD has one task. We only consider the case that SBS can transfer data to MBS and SMDs cannot offload tasks to MBS directly [24, 25]. Moreover, SMDs occupy orthogonal wireless channels. The k-SMD has the task denoted as $A_k(D_k, C_k, T_k)$. The task A_k containing D_k bits needs to be completed in time T_k. Each bit needs C_k cycles. We assume the task belongs to data-partitioned oriented tasks [6], which can be segmented arbitrarily, such as virus scan task and GZip task. The task can be executed separately in three tiers, i.e., SMDs, SBS, and MBS (Specially, in virus scan, the files can be partitioned into three parts. Then, each tier can scan a part of the total files in parallel. Finally, the results of three tiers are combined and the final result is obtained.) $\boldsymbol{\alpha}_k = [\alpha_{k,l}, \alpha_{k,s}, \alpha_{k,m}] (0 \le \alpha_{k,l}, \alpha_{k,s}, \alpha_{k,m} \le 1)$ is set as the workload distribution. $\alpha_{k,l}$, $\alpha_{k,s}$, and $\alpha_{k,m}$ denote the proportion of k-SMD workload, SBS workload, and MBS workload, respectively. We assume that the computation results are so small that the time delay from SBS and MBS to SMDs can be ignored [15, 30, 31].

2.1. Local Computing and Transmitting Model

2.1.1. Local Computing Model. The number of bits needed to be processed locally is $\alpha_{k,l} D_k$ and thus, it needs $\alpha_{k,l} D_k C_k$ cycles. The latency of local computing is denoted as $t_{k,l}^{\text{comp}}$ and obtained as

$$t_{k,l}^{\text{comp}} = \frac{\alpha_{k,l} D_k C_k}{f_{k,l}}. \tag{1}$$

We consider a low voltage task execution model and the energy consumed by one CPU cycle is denoted as ε given by

$$\varepsilon = \kappa f_{k,l}^2, \tag{2}$$

where κ is a constant related to capacitance coefficient [15]. Then, the computing energy consumed locally is written as

$$E_{k,l}^{\text{comp}} = \alpha_{k,l} D_k C_k \varepsilon = \alpha_{k,l} D_k C_k \kappa f_{k,l}^2, \tag{3}$$

where $E_{k,l}^{\text{comp}}$ denotes the k-SMD energy consumption of local computing.

2.1.2. Local Transmitting Model. The transmitting channel between SMDs and SBS is assumed as Rayleigh channels [6]. We assume that the coherence time is larger than the task deadline T_k, i.e., the channel gain is invariant during the task execution [31]. The channel gain is denoted as g_k, and the task offloading rate can be obtained as

$$r_k = B \log_2 \left(1 + \frac{p_{k,tx} g_k}{N_0}\right), \tag{4}$$

where r_k, B, $p_{k,tx}$, and N_0 denote k-SMD's transmit rate, channel bandwidth, transmit power, and white Gaussian noise power, respectively. The k-SMD's transmit power cannot exceed the maximum transmit power $p_{k,tx}^{\max}$. $\mathbf{p}_{tx}$ denotes the SMDs' transmit power vector, which is expressed as $[p_{1,tx}, p_{2,tx}, \cdots, p_{K,tx}]$. The task offloaded to MBS needs to be transferred to SBS first. Thus, the offloading time of k-SMD $t_{k,l}^{\text{trans}}$ is obtained as

$$t_{k,l}^{\text{trans}} = \frac{(\alpha_{k,s} + \alpha_{k,m}) D_k}{r_k}. \tag{5}$$

The offloading energy consumption is the product of the offloading time and transmit power as

$$E_{k,l}^{\text{trans}} = t_{k,l}^{\text{trans}} p_{k,tx} = \frac{(\alpha_{k,s} + \alpha_{k,m}) D_k}{r_k} p_{k,tx}. \tag{6}$$

2.2. Computation Model

2.2.1. SBS Computing Model. SBS has limited computation capabilities because of its limited volume compared with MBS. k-SMD has a priority β_k from the telecom operator, which decides the portion of SBS computation capability allocated to k-SMD. The SBS computation ability of k-SMD is denoted as $f_{k,s}$ cycles/s. $\mathbf{f}_s = [f_{1,s}, f_{2,s}, \cdots, f_{K,s}]$ denotes SBS computation capability allocation vector of K SMDs. And the following limitations exist:

$$\begin{gathered} 0 \le f_{k,s} \le \beta_k F, \\ \sum_{k=1}^{K} \beta_k = 1. \end{gathered} \tag{7}$$

The SBS workload from k-SMD is $\alpha_{k,s}D_k$, and the number of its computation cycles is $\alpha_{k,s}D_kC_k$. The time delay of SBS execution is obtained as

$$t_{k,s}^{\text{comp}} = \frac{\alpha_{k,s}D_kC_k}{f_{k,s}}. \tag{8}$$

The total delay of SBS computing is made up of offloading delay and execution delay, which is given by

$$t_{k,s} = t_{k,l}^{\text{trans}} + t_{k,s}^{\text{comp}}. \tag{9}$$

2.2.2. MBS Computing Model. The backhaul link delay $t_{k,m}^{\text{trans}}$ is proportional to the transfer data size, i.e., the transfer delay between SBS and MBS is calculated as

$$t_{k,m}^{\text{trans}} = \phi D_k \alpha_{k,m}. \tag{10}$$

The MBS execution latency can be ignored. Therefore, the delay of MBS computing $t_{k,m}$ is the sum of offloading delay and backhaul link delay as

$$t_{k,m} = t_{k,l}^{\text{trans}} + t_{k,m}^{\text{trans}}. \tag{11}$$

2.3. Problem Formulation. Based on equations (3) and (6), the energy consumption of k-SMD E_k, which consists of computing consumption and transmitting consumption, is written as

$$E_k = E_{k,l}^{\text{comp}} + E_{k,l}^{\text{trans}}. \tag{12}$$

The task of k-SMD is executed parallel in three-tier (local devices, SBS, and MBS), and thus, the execution delay t_k is obtained as

$$t_k = \max\left\{t_{k,l}^{\text{comp}}, t_{k,s}, t_{k,m}\right\}. \tag{13}$$

The energy-efficient problem under tasks delay constraints is formulated as

$$\mathbf{P1}: \min_{\boldsymbol{\alpha}_k,\mathbf{p}_{tx},\mathbf{f}_s} \sum_{k=1}^{K} E_k, \tag{14a}$$

$$\text{s.t. } t_k \le T_k, \tag{14b}$$

$$0 \le f_{k,s} \le \beta_k F, \tag{14c}$$

$$\sum_{k=1}^{K} \beta_k = 1, \tag{14d}$$

$$\alpha_{k,l} + \alpha_{k,s} + \alpha_{k,m} = 1, \tag{14e}$$

$$0 \le \alpha_{k,l}, \alpha_{k,s}, \alpha_{k,m} \le 1, \tag{14f}$$

$$0 \le p_{k,tx} \le p_{k,tx}^{\max}, \tag{14g}$$

$$\forall k \in \mathscr{K}, \tag{14h}$$

where (14b) means that the delay needs to meet the demand. (14c) indicates that the SBS computation capability allocated to k-SMD cannot exceed the maximum allocation frequency. (14e) denotes that the sum workload of the local device, SBS, and MBS needs to be equal to the total task load of k-SMD.

3. Problem Solution

In this section, for gaining some engineering insights, an EERA scheme based on the variable substitution technique [6, 32] is proposed to solve problem **P1**. Firstly, we fix $\mathbf{p}_{tx}$ and find the optimal workload distribution $\boldsymbol{\alpha}_k^*$ and SBS computation capability allocation $\mathbf{f}_s^*$ by minimizing $\sum_{k=1}^{K} E_k$. Then, we use $\boldsymbol{\alpha}_k^*$ and $\mathbf{f}_s^*$ to find the optimal transmit power $\mathbf{p}_{tx}^*$.

According to equations (3), (6), and (12), E_k can be rewritten as

$$E_k = \alpha_{k,l}D_kC_k\kappa f_{k,l}^2 + \frac{(\alpha_{k,s} + \alpha_{k,m})D_k}{r_k} p_{k,tx}. \tag{15}$$

Substituting equation (14e) into (15), E_k can be written as

$$E_k = \alpha_{k,l}\left(D_kC_k\kappa f_{k,l}^2 - \frac{D_k p_{k,tx}}{r_k}\right) + \frac{D_k p_{k,tx}}{r_k}. \tag{16}$$

3.1. Problem Decomposition. Fixing transmission power $\mathbf{p}_{tx}$, problem **P1** is simplified to problem **P2**, where the second term of equation (16) is fixed and can be eliminated.

$$\mathbf{P2}: \min_{\boldsymbol{\alpha}_k,\mathbf{f}_s} \sum_{k=1}^{K} \alpha_{k,l}\left(D_kC_k\kappa f_{k,l}^2 - \frac{D_k p_{k,tx}}{r_k}\right), \tag{17a}$$

$$\text{s.t. } t_k \le T_k, \tag{17b}$$

$$0 \le f_{k,s} \le \beta_k F, \tag{17c}$$

$$\alpha_{k,l} + \alpha_{k,s} + \alpha_{k,m} = 1, \tag{17d}$$

$$\sum_{k=1}^{K} \beta_k = 1, \tag{17e}$$

$$0 \le \alpha_{k,l}, \alpha_{k,s}, \alpha_{k,m} \le 1, \tag{17f}$$

$$\forall k \in \mathscr{K}, \tag{17g}$$

where transmit power vector $\mathbf{p}_{tx}$ is fixed. Substituting the solution of problem **P2** into equation (15) and optimizing $\mathbf{p}_{tx}$ by minimizing $\sum_{k=1}^{K} E_k$, we formulate problem **P3** as

$$\mathbf{P3}: \min_{\mathbf{p}_{tx}} \sum_{k=1}^{K} \alpha_{k,l}D_kC_k\kappa f_{k,l}^2 + (\alpha_{k,s} + \alpha_{k,m})D_k \frac{p_{k,tx}}{r_k}, \tag{18a}$$

$$\text{s.t. } t_k \le T_k, \tag{18b}$$

$$0 \le p_{k,tx} \le p_{tx}^{\max}, \tag{18c}$$

$$\forall k \in \mathscr{K}. \tag{18d}$$

3.2. Energy-Efficient Resource Allocation Scheme. We define the transmission energy consumption per bit as v_k, which is obtained as

$$v_k = \frac{p_{k,tx}}{r_k}. \tag{19}$$

Lemma 1. *v_k increases monotonically with the increase of $p_{k,tx}$.*

Proof. See Appendix A.

Define e_k as

$$e_k = \alpha_{k,l}\left(D_k C_k \kappa f_{k,l}^2 - \frac{D_k p_{k,tx}}{r_k}\right). \tag{20}$$

Lemma 2. *Based on Lemma 1, e_k changes with v_k as follows:*

(1) $v_k > \kappa C_k f_{k,l}^2$, e_k decreases monotonically with the increase of $\alpha_{k,l}$.

(2) $v_k < \kappa C_k f_{k,l}^2$, e_k increases monotonically with the increase of $\alpha_{k,l}$.

(3) $v_k = \kappa C_k f_{k,l}^2$, e_k is independent of $\alpha_{k,l}$.

Proof. The derivative of e_k is $(\mathrm{d}e_k/\mathrm{d}\alpha_{k,l}) = D_k C_k \kappa f_{k,l}^2 - (D_k p_{k,tx}/r_k)$. When $v_k > \kappa C_k f_{k,l}^2$, $(\mathrm{d}e_k/\mathrm{d}\alpha_{k,l}) < 0$ and e_k decreases monotonically with the increase of $\alpha_{k,l}$. The second case and the third case can be proved by the same way as the first case.

Based on Lemma 1 and Lemma 2, we can judge whether problem **P1** has a solution or not and get Lemma 3.

Lemma 3. *Problem **P1** is feasible.*

Proof. See Appendix B.

Remark 4. When $v_k > \kappa C_k f_{k,l}^2$, i.e., the energy consumed per bit by offloading is more than the energy consumed per bit by local execution. More bits will be processed in the local device to save energy. That is why $\mathrm{e_k}$ decreases monotonically with the increase of $\alpha_{k,l}$. In the second case of Lemma 2, $v_k < \kappa C_k f_{k,l}^2$, i.e., the energy consumed per bit by offloading is less than the energy consumed per bit by local execution. More bits will be processed by offloading to save energy. That is why e_k increases monotonically with the increase of $\alpha_{k,l}$.

Remark 5. According to Lemma 1, v_k increases monotonically with the increase of $p_{k,tx}$. From equation (4), a larger r_k is due to a larger $p_{k,tx}$ and a larger $p_{k,tx}$ induces a larger r_k. Wherefore, the larger is r_k, the larger is v_k. According to equation (15), when v_k becomes larger, E_k becomes larger. Thus, E_k increases with the increase of r_k, i.e., the energy consumption of SMDs increases with the increase of r_k. In other words, the SMD will consume more energy when having a higher offloading rate.

Substituting equation (13) into inequality (14b), we get

$$\begin{cases} t_{k,l}^{\text{comp}} \le T_k, \\ t_{k,s} \le T_k, \\ t_{k,m} \le T_k. \end{cases} \tag{21}$$

In order to simplify problem **P2**, $t_{k,s}$ and $t_{k,m}$ are compared and then, problem **P2** becomes problem **P2.1** and problem **P2.2**.

When $t_{k,s} \ge t_{k,m}$, i.e., the delay of SBS computing is larger than MBS computing, problem **P2** becomes problem **P2.1**, which is written as

$$\mathbf{P2.1}: \min_{\boldsymbol{\alpha}_k, \mathbf{f}_s} \sum_{k=1}^{K} \alpha_{k,l}\left(D_k C_k \kappa f_{k,l}^2 - \frac{D_k p_{k,tx}}{r_k}\right), \tag{22a}$$

$$\text{s.t.}\, t_{k,l}^{\text{comp}} \le T_k, \tag{22b}$$

$$t_{k,s} \le T_k, \tag{22c}$$

$$t_{k,s} \ge t_{k,m}, \tag{22d}$$

$$(17\text{c}) - (17\text{g}). \tag{22e}$$

When $t_{k,s} < t_{k,m}$, i.e., the delay of MBS computing is larger than that in SBS computing, problem **P2** becomes problem **P2.2**, which can be written as

$$\mathbf{P2.2}: \min_{\boldsymbol{\alpha}_k, \mathbf{f}_s} \sum_{k=1}^{K} \alpha_{k,l}\left(D_k C_k \kappa f_{k,l}^2 - \frac{D_k p_{k,tx}}{r_k}\right), \tag{23a}$$

$$\text{s.t.}\ t_{k,l}^{\text{comp}} \le T_k, \tag{23b}$$

$$t_{k,m} \le T_k, \tag{23c}$$

$$t_{k,s} < t_{k,m}, \tag{23d}$$

$$(17\text{c}) - (17\text{g}). \tag{23e}$$

According to Lemma 2, three cases are dealt with, respectively, to solve problem **P1**.

(1) $v_k > \kappa C_k f_{k,l}^2$: when the energy consumed per bit by offloading is more than the energy consumed per bit by local execution, the following derivations exist.

Lemma 6. *Both problems **P2.1** and **P2.2** have the same optimal local task load $\alpha_{k,l}^*$ as*

$$\alpha_{k,l}^* = \frac{f_{k,l} T_k}{D_k C_k}. \tag{24}$$

Proof. From inequalities (22b) and (23b), $\alpha_{k,l} \le (f_{k,l} T_k / D_k C_k)$ is obtained. In the light of the first case of Lemma 2, e_k decreases monotonously with the increase of $\alpha_{k,l}$. Wherefore, we take $\alpha_{k,l}^* = (f_{k,l} T_k / D_k C_k)$.

Remark 7. According to equation (24), the local workload is related to local computation ability and the task delay constraint. Larger local computation ability brings a larger local workload. In order to save energy, SMDs will process as many bits as possible locally if the processing latency meets the task delay constraint. Looser delay constraint brings the SMD a larger local workload. Looser delay constraint means that the local device has more time to execute the task and thus process more bits locally to save energy.

Lemma 8. *Define $\alpha^*_{k,s}$, $\alpha^*_{k,m}$, and $f^*_{k,s}$ as the optimal SBS workload, MBS workload, and computation ability allocated from SBS, respectively. When $v_k > \kappa C_k f^2_{k,l}$, both problem **P2.1** and problem **P2.2** have*

$$\alpha^*_{k,s} = \frac{\phi f^*_{k,s}}{C_k}\alpha^*_{k,m}, \tag{25}$$

$$f^*_{k,s} = \beta_k F. \tag{26}$$

Proof. See Appendix C.

Remark 9. According to equation (25), $\alpha^*_{k,s}$ is related to backhaul link delay coefficient ϕ and the computation ability $f^*_{k,s}$ allocated from SBS. When much SBS computation ability is allocated to k-SMD or backhaul link delay is large, the SBS workload will be large. In other words, the task will be executed prior in SBS unless MBS execution costs less time.

When $v_k > \kappa C_k f^2_{k,l}$, based on Lemma 6 and Lemma 8, the solution of problem **P2** can be obtained as Theorem 10.

Theorem 10. *The optimal workload distribution α^*_k and the optimal allocation of SBS computation ability $\mathbf{f}^*_s$ can be obtained as*

$$f^*_{k,s} = \beta_k F, \tag{27}$$

$$\begin{cases} \alpha^*_{k,l} = \dfrac{f_{k,l}T_k}{D_k C_k}, \\ \alpha^*_{k,s} = \dfrac{\phi f^*_{k,s}\left(D_k C_k - f_{k,l}T_k\right)}{C_k D_k\left(C_k + \phi f^*_{k,s}\right)}, \\ \alpha^*_{k,m} = \dfrac{D_k C_k - f_{k,l}T_k}{D_k\left(C_k + \phi f^*_{k,s}\right)}. \end{cases} \tag{28}$$

Proof. Substituting equations (24)–(26) into equation (17d), the optimal allocation of SBS computation ability and the optimal workload distribution can be obtained.

In the light of Remark 5, the optimal transmission rate r^*_k can be calculated by Lemma 11 and then, problem P3 can be solved.

Lemma 11. *Problem **P2.1** and problem **P2.2** have the same optimal transmission rate r^*_k as*

$$r^*_k = \frac{\left(1-\alpha^*_{k,l}\right)D_k}{T_k - \alpha^*_{k,s}D_k C_k / f^*_{k,s}}. \tag{29}$$

Proof. According to inequalities (C.3) and (C.9), we choose the lower boundary of $r_k$as r^*_k for saving energy. Considering Lemma 8, r^*_k of problems **P2.1** and **P2.2** are same and equation (29) is obtained.

Then, substituting equation (29) into equation (4), we attain the optimal solution of problem **P3** by Theorem 12.

Theorem 12. *The optimal transmission power $p^*_{k,tx}$ is given by*

$$p^*_{k,tx} = \frac{N_0}{g_k}\left(2^{\frac{\left(1-\alpha^*_{k,l}\right)D_k}{B\left(T_k-\alpha^*_{k,s}D_kC_k/f^*_{k,s}\right)}} - 1\right). \tag{30}$$

Remark 13. As can be seen from equation (30), smaller $\alpha^*_{k,l}$ and larger $\alpha^*_{k,s}$ induce larger k-SMD's transmission power $p^*_{k,tx}$. When the proportion of the task executed locally is small, the offloading rate should be large enough to meet the task delay constraint, which results in large transmission power. Similarly, larger $\alpha^*_{k,s}$ means more bits will be processed in SBS and means a larger offloading rate, which accounts for larger transmit power.

(2) $v_k < \kappa C_k f^2_{k,l}$: when the energy consumed by offloading per bit is less than the energy consumed by local execution per bit, offloading will be prior to local execution for saving energy, i.e., smaller $\alpha_{k,l}$ will be better for saving energy.

Considering problem **P2.1**, we have the optimal local workload as Lemma 14.

Lemma 14. *The optimal $\alpha_{k,l}$ of problem **P2.1** can be given by*

$$\alpha_{k,l} = 1 - \frac{T_k r_k}{D_k} + \frac{\alpha_{k,s}C_k r_k}{f_{k,s}}. \tag{31}$$

Proof. We have $\alpha_{k,l} \geq 1 - \left(T_k r_k / D_k\right) + \left(\alpha_{k,s}C_k r_k / f_{k,s}\right)$ by substituting equations (5), (8), and (9) into inequality (22c). Smaller $\alpha_{k,l}$ leads to less energy consumption of k-SMD. Therefore, we take $\alpha_{k,l} = 1 - \left(T_k r_k / D_k\right) + \left(\alpha_{k,s}C_k r_k / f_{k,s}\right)$.

Similarly to Lemma 14, we obtain the optimal local workload of problem **P2.2** as Lemma 15 using inequality (23c).

Lemma 15. *The optimal $\alpha_{k,l}$ of problem **P2.2** can be calculated as*

$$\alpha_{k,l} = 1 - \frac{T_k r_k}{D_k} + \phi\alpha_{k,m}r_k. \tag{32}$$

Lemma 16. *When $v_k < \kappa C_k f^2_{k,l}$, the optimal MBS workload $\alpha^*_{k,m}$ and SBS workload $\alpha^*_{k,s}$ have*

$$\alpha^*_{k,m} = \frac{C_k}{\phi f_{k,s}}\alpha^*_{k,s}. \tag{33}$$

Proof. Considering problem **P2.1**, we obtain $t_{k,s} \geq t_{k,m}$, where $\alpha_{k,s} \geq (\phi f_{k,s}/C_k)\alpha_{k,m}$ is attained according to equations (9) and (11). From equation (31), smaller $(\alpha_{k,s}/f_{k,s})$ will be better for saving energy. Thus, we take $\alpha_{k,s} = (\phi f_{k,s}/C_k)\alpha_{k,m}$. Considering problem **P2.2**, we get $\alpha_{k,m} > (C_k/\phi f_{k,s})\alpha_{k,s}$ from $t_{k,s} < t_{k,m}$.

According to equation (32), smaller $\alpha_{k,m}$ brings smaller $\alpha_{k,l}$ and saves more energy. In addition, $\alpha_{k,m}$ can approach $(C_k/\phi f_{k,s})\alpha_{k,s}$ as much as possible because of the continuity of $\alpha_{k,m}$. Hence, we obtain $\alpha^*_{k,m} = (C_k/\phi f_{k,s})\alpha^*_{k,s}$.

Remark 17. There always exists $\alpha^*_{k,s} = (\phi f_{k,s}/C_k)\alpha^*_{k,m}$ whether v_k is larger than $\kappa C_k f^2_{k,l}$ or not. It indicates that the energy consumed by offloading per bit has nothing to do with the relation between $\alpha^*_{k,s}$ and $\alpha^*_{k,m}$. The relation depends on the computation ability allocated from SBS and transfer delay of backhaul link, i.e, the distribution of workload between SBS and MBS is decided jointly by the computation ability allocated from SBS and MBS time cost.

Remark 18. Based on Lemma 14, Lemma 15, and Lemma 16, we easily find that problem **P2.1** and problem **P2.2** have the same optimal $\alpha_{k,l}$. In other words, the optimal workload of local devices $\alpha^*_{k,l}$ is independent of the workload distribution between SBS and MBS.

Remark 19. In the second case of Lemma 2, problem **P2.1** and problem **P2.2** have the same optimal local workload $\alpha^*_{k,l}$ and same relation between $\alpha^*_{k,s}$ and $\alpha^*_{k,m}$. Therefore, according to equation (17d), problem **P2.1** and problem **P2.2** have the same optimal solution about α^*_k.

Based on Remark 19, the solution of problem **P2** can be obtained by Theorem 20.

Theorem 20. *When $v_k < \kappa C_k f^2_{k,l}$, the optimal computation ability allocation from SBS $f^*_{k,s}$ and the optimal workload distribution α^*_k among SMDs, SBS, and MBS can be attained as*

$$f^*_{k,s} = \beta_k F, \tag{34}$$

$$\begin{cases} \alpha^*_{k,l} = 1 - \dfrac{T_k r_k}{D_k} + \dfrac{C_k T_k \phi r_k^2}{D_k\left(\phi C_k r_k + C_k + \phi f^*_{k,s}\right)}, \\ \alpha^*_{k,s} = \dfrac{T_k r_k \phi f^*_{k,s}}{D_k\left(\phi C_k r_k + C_k + \phi f^*_{k,s}\right)}, \\ \alpha^*_{k,m} = \dfrac{T_k r_k C_k}{D_k\left(\phi C_k r_k + C_k + \phi f^*_{k,s}\right)}. \end{cases} \tag{35}$$

Proof. Substituting equations (31) and (33) into equation (17d), the optimal workload distribution α^*_k can be obtained. In addition, from equation (31), $\alpha_{k,l}$ decreases with the increase of $f_{k,s}$. A larger $f_{k,s}$ brings a smaller $\alpha_{k,l}$ and saves more energy. Thus, we take $f^*_{k,s} = \beta_k F$.

```
Input: error $e$, start point $p^{start}_{k,tx}$, end point $p^{end}_{k,tx}$
Output: $p^*_{k,tx}$
Initialization:
    $p^{start}_{k,tx} = 0$
    $p^{end}_{k,tx} = v_k^{-1}(\kappa C_k f^2_{k,l})$
    $p^{mid}_{k,tx} = \frac{p^{start}_{k,tx} + p^{end}_{k,tx}}{2}$
    $d = p^{end}_{k,tx} - p^{start}_{k,tx}$
while $d > e$ do
    if $Q(p^{mid}_{k,tx} - e/2) \geq Q(p^{mid}_{k,tx} + e/2)$ then
        $p^{start}_{k,tx} = p^{mid}_{k,tx}$
    else
        $p^{end}_{k,tx} = p^{mid}_{k,tx}$
    $d = p^{end}_{k,tx} - p^{start}_{k,tx}$
    $p^{mid}_{k,tx} = \frac{p^{start}_{k,tx} + p^{end}_{k,tx}}{2}$
return $p^{mid}_{k,tx}$
```

Algorithm 1: Binary search for $p^*_{k,tx}$.

Considering problem **P3**, we substitute equations (34) and (35) into E_k and get the optimal transmit power $p^*_{k,tx}$ as Theorem 21.

Theorem 21. *When $v_k < \kappa C_k f^2_{k,l}$, the optimal transmission power $p^*_{k,tx}$ is*

$$p^*_{k,tx} = \min\left\{p, v_k^{-1}\left(\kappa C_k f^2_{k,l}\right)\right\}, \tag{36}$$

where $(dQ/dp_{k,tx})|_{p_{k,tx}=p} = 0$ and v_k^{-1} denotes the inverse function of $v_k Q$ is defined as

$$Q = \frac{T_k p_{k,tx} - \kappa C_k f^2_{k,l} T_k r_k}{\phi C_k r_k + C_k + \phi f_{k,s}}. \tag{37}$$

Proof. See Appendix D.

It is difficult to solve v_k^{-1} and $(dQ/dp_{k,tx})|_{p_{k,tx}=p} = 0$. Hence, some tools are used to get the optimal transmission power $p^*_{k,tx}$. In the first step, we use MATLAB to get the maximum transmission power $p^{end}_{k,tx}$ from $p^{end}_{k,tx} = v_k^{-1}(\kappa C_k f^2_{k,l})$. In the second step, we use the binary search technique to search the optimal transmit power $p^*_{k,tx}$ between 0 and $p^{end}_{k,tx}$ for minimizing Q. The variables e, d, and $p^{mid}_{k,tx}$ denote the search error, search interval, and interval midpoint, respectively. The search is not stopped until $d < e$. The detailed search process is summarized in Algorithm 1.

(3) $v_k = \kappa C_k f^2_{k,l}$: when $v_k = \kappa C_k f^2_{k,l}$, i.e., $e_k = 0$, $\alpha_{k,l}$ cannot change e_k. In this case, the energy consumed per bit by local execution equals the energy consumed per bit by offloading. Offloading cannot reduce energy consumption of task execution. We choose to execute tasks in local devices or the entire

```
Step 1: According to Theorem 10 and Theorem 12, calculate
$\boldsymbol{\alpha}_k^*$, $\mathbf{f}_s^*$ and $\mathbf{p}_{tx}^*$.
Step 2: Based on equation (19), compute $v_k^*$ by substituting
the results of Step 1.
Step 3:
if $v_k^* < \kappa C_k f_{k,l}^2$ then
    recompute $\boldsymbol{\alpha}_k^*$ $f_{k,s}^*$ and $p_{k,tx}^*$ according to Theorem 20
and Theorem 21.
else if $v_k^* = \kappa C_k f_{k,l}^2$ then
    recompute $\boldsymbol{\alpha}_k^*$ $f_{k,s}^*$ and $p_{k,tx}^*$ according to Theorem 22.
```

Algorithm 2: The Main Process of the Energy-Efficient Resource Allocation Scheme

offloading to minimize execution latency and get the following Theorem 22.

Theorem 22. *(1) When $t_{k,l}^{all} \le t_{k,off}^*$, the task will be executed entirely by the local device and have*

$$f_{k,s}^* = 0,$$
$$p_{k,tx}^* = 0,$$
$$\begin{cases} \alpha_{k,l}^* = 1, \\ \alpha_{k,s}^* = 0, \\ \alpha_{k,m}^* = 0. \end{cases} \tag{38}$$

(2) When $t_{k,l}^{all} > t_{k,off}^*$, the task will be executed entirely by offloading and have

$$f_{k,s}^* = \beta_k F,$$
$$p_{k,tx}^* = v_k^{-1}\left(\kappa C_k f_{k,l}^2\right),$$
$$\begin{cases} \alpha_{k,l}^* = 0, \\ \alpha_{k,s}^* = \dfrac{\phi f_{k,s}^*}{C_k + \phi f_{k,s}^*}, \\ \alpha_{k,m}^* = \dfrac{C_k}{C_k + \phi f_{k,s}^*}. \end{cases} \tag{39}$$

In (1) and (2), the latency of local execution entirely is denoted as $t_{k,l}^{all}$ and $t_{k,l}^{all} = (D_k C_k / f_{k,l})$. The minimum offloading latency is denoted as $t_{k,off}^*$ and $t_{k,off}^* = (D_k / B\log_2(1 + (v_k^{-1}(\kappa C_k f_{k,l}^2) g_k / N_0))) + (\phi D_k C_k / C_k + \phi f_{k,s}^*)$.

Proof. See Appendix E.

By now, the optimal solution of problem **P1** is given by the theorems and the procedure is described in Algorithm 2.

3.3. Analysis of Special Cases. From the first four theorems, we not only consider energy minimization but also consider the delay constraint. That is why we still allocate resources when we know the case with the least energy consumption.

In Theorem 22, we only consider the latency. In this case, energy consumed per bit by offloading equals the energy consumed per bit by local execution, i.e., the offloading will not reduce energy consumption of the task execution. We cannot use the offloading to reduce SMDs' energy consumption. However, we can choose the solution with the least delay to try to improve users' QoE. Wherefore, we choose to execute the task either locally or remotely according to the latencies of the task execution in the local device and offloading.

4. Numerical Results

In this section, numerical results are given to evaluate the performances of the proposed EERA scheme, as compared to the following baseline schemes.

(i) Local Computing Only: all SMDs perform their own tasks by only local computing

(ii) Full Offloading: all SMDs accomplish their own tasks by fully offloading

(iii) Computing without MBS: the tasks are performed only by local devices and SBS server. Resource allocation for minimizing all SMDs' energy consumption only takes place on local devices and the SBS server

Some parameters are set as follows unless stated otherwise. The tasks models of all SMDs are set to be identical, i.e., $D_k = 10$ kbits [15], $C_k = 1000$ cycles/bit [15], and $T_k = 2$ ms $(k \in \mathcal{K})$ [9]. The local computation capability $f_{k,l}$ equals 5×10^8 cycles/s [33]. The energy coefficient of local computation κ is 10^{-28} [15]. The maximum transmission power $p_k^{\max}$ is 0.1 watts [6]. The computation capability of the SBS server is 8×10^9 cycles/s [34]. The backhaul time delay coefficient ϕ is set to be 1.25×10^{-8} sec/bit [24]. We consider a Rayleigh fading channel model, and the channel gain $g_k = \lambda \bar{g}_k$. λ is an independent exponential random

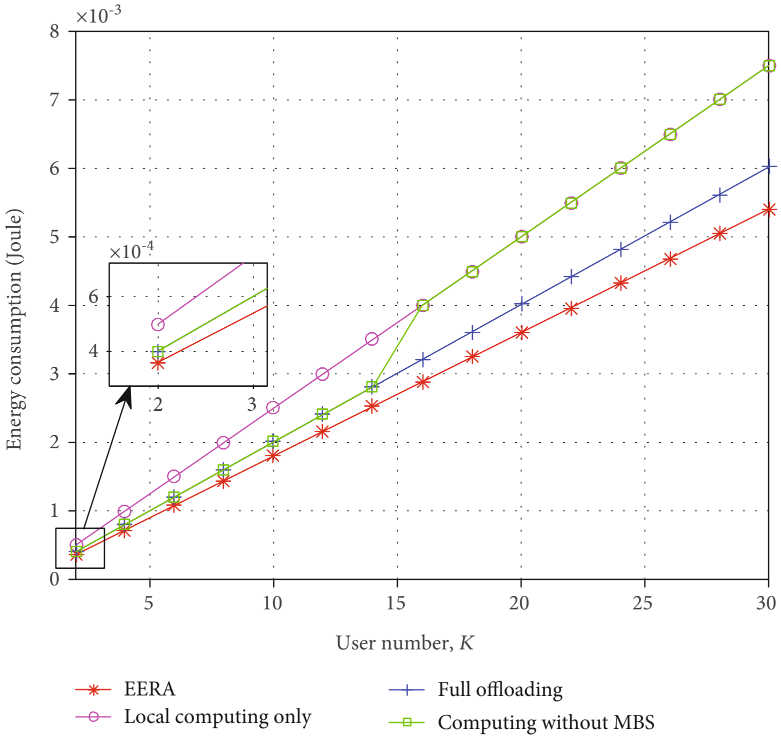


Figure 2: Energy consumption versus user number.

variable of unit mean. $\bar{g}_k$ follows the free-space path loss model as

$$\bar{g}_k = A_d \left(\frac{3 \times 10^8}{4\pi f_c d_k} \right)^{d_e}, \tag{40}$$

where $A_d = 4.11$ denotes the antenna gain, $f_c = 915$ MHz denotes the carrier frequency, $d_k = 18\,\text{m}$ denotes the distance from the SBS to k-SMD [6], and $d_e = 2.8$ denotes the path loss exponent. The channel bandwidth B is 2 MHz [31].

4.1. Performances of EERA. In this subsection, we analyze the performances of EERA compared with local-computing-only, full-offloading, and computing-without-MBS. Figures 2–5 present the energy consumption of SMDs under different conditions. It is shown that the proposed EERA achieves the lowest energy consumption among those four methods.

Figure 2 plots the sum energy consumption of all SMDs versus the user number K. It is shown that the energy consumption by all the schemes increases as the user number grows. Besides, the energy consumption of computing-without-MBS is close to full-offloading when the user number is less than 15 while close to local-computing-only when the user number is greater than 15. The reason is that the computation resource that each user obtains from the SBS server becomes less as the user number increases. And SMDs process more bits locally for meeting the tasks' deadline. It is also observed that EERA outperforms the other schemes. This is because EERA has more computation capacity thanks to the MBS server. And lower execution latency gives more time to offload computation bits.

Figure 3 depicts the sum energy consumption of all SMDs versus the computation tasks size D. It is shown that the energy consumption by the four schemes rises with the computation task size growth. When the computation task size is small, the energy consumption of computing-without-MBS is less than that of local-computing-only and more than that of full-offloading. When the computation task size is large, computing-without-MBS is close to full offloading. It indicates that the number of local computation bits decreases with the computation task size increase under the task latency constraints. The energy consumption of EERA is the least among these methods. In addition, the gap between EERA and full-offloading is gradually widening when the computation task size is less than 9.45 kbits and narrowing when the computation task size is greater than 9.45 kbits. The reason is that offloading consumes less energy and EERA processes more bits by offloading when the computation task size is small. To meet the task latency demand, more bits are offloaded when the computation task size grows.

Figure 4 shows the sum energy consumption versus the channel bandwidth B. As we can see, with the increase of the channel bandwidth, the energy consumption by local-computing-only remains invariant while other schemes decrease. The reason is that local-computing-only has nothing to do with offloading. However, other schemes can reduce transmit power owing to a bigger bandwidth under

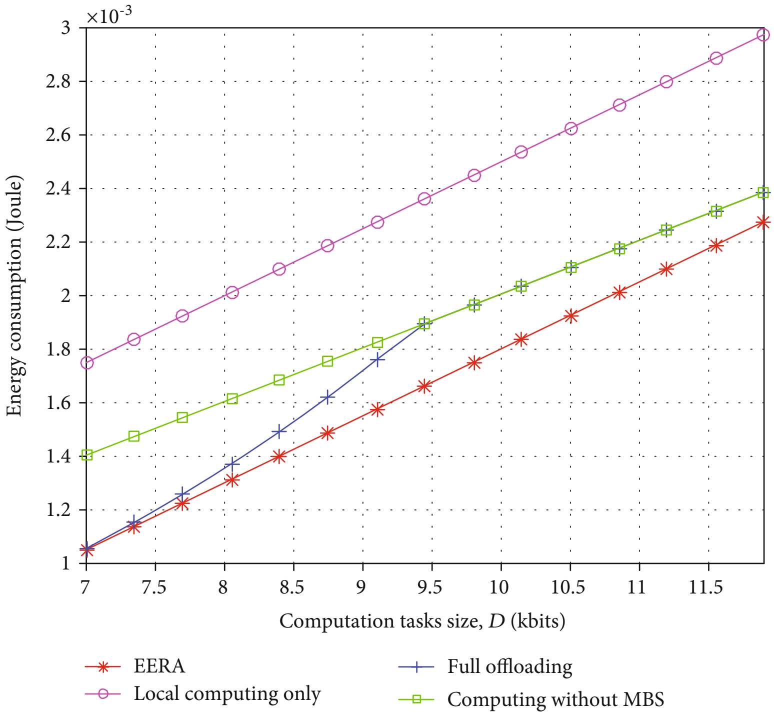


Figure 3: Energy consumption versus computation task size.

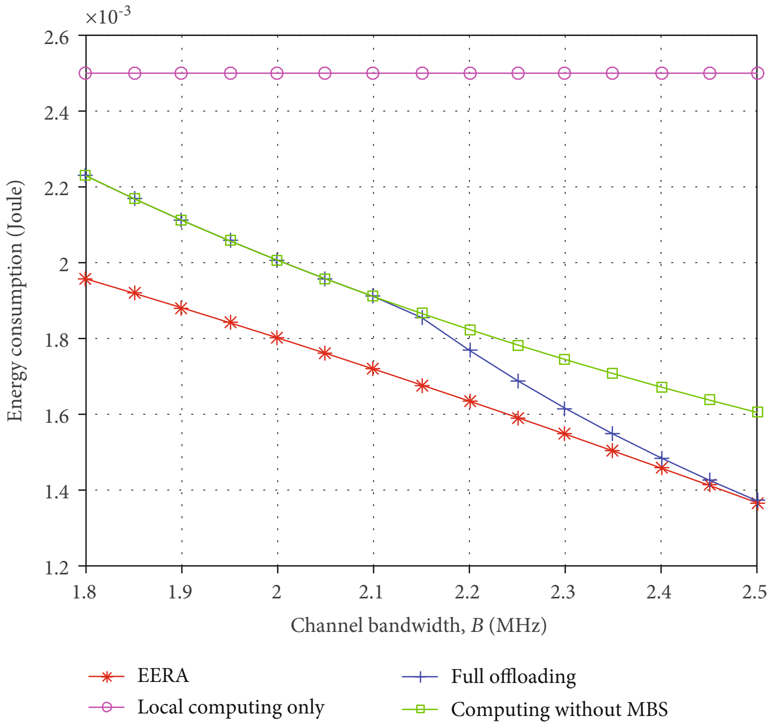


Figure 4: Energy consumption versus the channel bandwidth.

time delay constraints. The gap between full-offloading and computing-without-MBS is widening as the channel bandwidth grows. Full-offloading has more computation capability than computing-without-MBS and has lower execution latency, which leaves more time for offloading and lowers the transmit power. EERA is gradually close

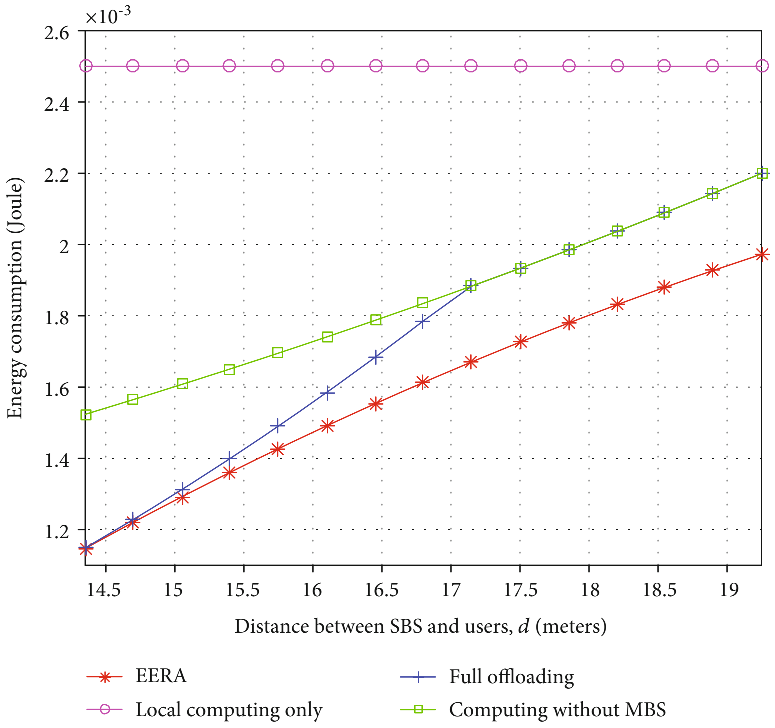


Figure 5: Energy consumption versus the distance from the SBS to users.

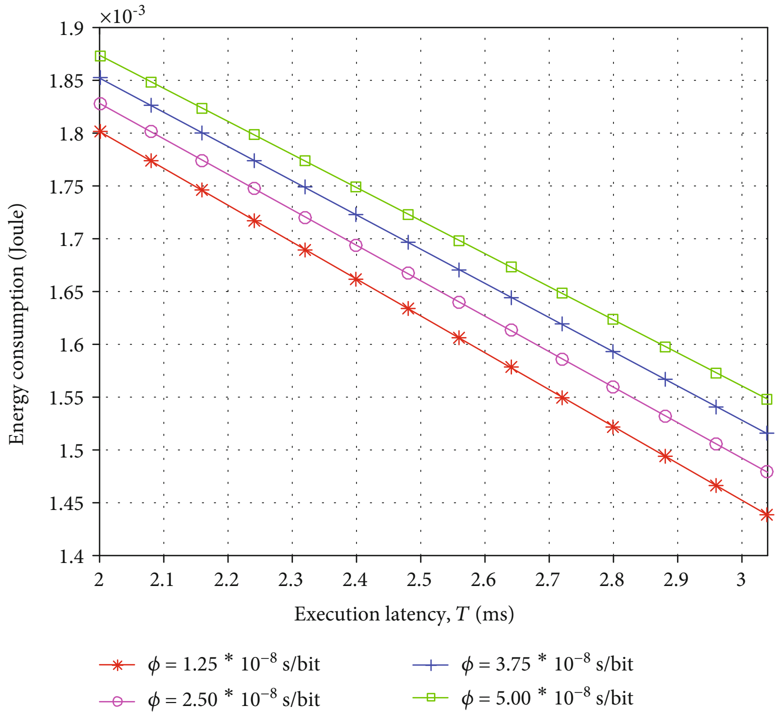
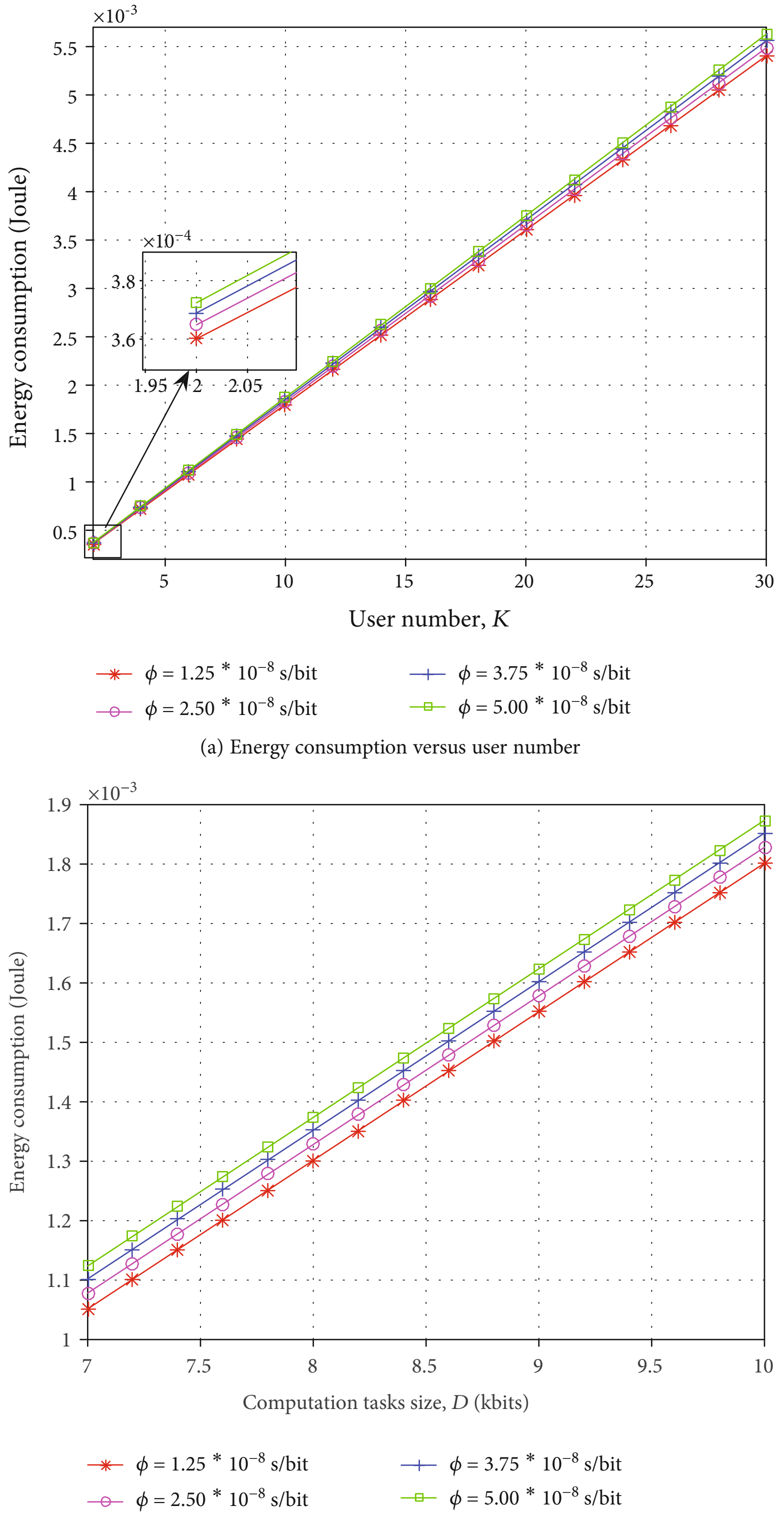


Figure 6: Energy consumption versus the execution latency.

to full-offloading as the channel bandwidth rises. It indicates that EERA processes more bits by offloading when the channel bandwidth is widening.

Figure 5 shows the sum energy consumption versus distance from the SBS to users. It is observed that these schemes except local-computing-only rise when the distance becomes



(a) Energy consumption versus user number

(b) Energy consumption versus computation tasks size

FIGURE 7: Effect of the backhaul time delay coefficient ϕ on energy consumption.

larger. Similar to Figure 4, local-computing-only has nothing to do with the communication distance. Longer distance leads to a larger path loss, which needs high transmit power to meet the time delay constraint. It is shown that the energy consumption by EERA is less than computing-without-MBS. That is because the existence of the MBS server lowers the execution latency and the transmit power. Moreover, the gap between EERA and full-offloading is widening. It illustrates that the offloading bit number becomes less owing to the longer communication distance.

4.2. Impacts of Backhaul Time Delay Coefficient ϕ.

In this subsection, we analyze the energy consumption with respect to the backhaul time delay coefficient in different conditions, e.g., the varying latency constraint, the varying user number, and the varying computation task size.

Figure 6 plots the sum energy consumption of all SMDs in different backhaul time delay coefficients ϕ versus the execution latency constraints. It is shown that the energy consumption decreases as the execution latency increases. The reason is that more time will be used to offload. And the low transmit power is allowed when the execution latency constraints relax.

Figure 7(a) shows the energy consumption versus user number under different ϕ. Figure 7(b) depicts the energy consumption versus computation task size given different ϕ. Combined with Figure 6, it is observed that a larger backhaul time delay coefficient results in larger energy consumption with the rise of the execution latency, user number, and computation task size. The reason is that a larger backhaul time delay coefficient increases the execution time and reduces the offloading time. Thus, transmit power increases for satisfying the task latency constraints.

5. Conclusion

In this paper, we investigated resource allocation mechanisms for three-tier MEC architecture in heterogeneous networks. We considered that both MBS and SBS are integrated with MEC servers and are combined with local devices to form a three-tier computing architecture. Each task from SMDs can be divided into three parts. SMDs, SBS, and MBS perform a part of the task, respectively. We formulated an optimization problem to minimize all SMDs' energy consumption under the time delay constraints. To improve the efficiency of resource allocation, we proposed an EERA mechanism based on the variable substitution technique, which jointly optimized the computation and radio resources. The optimal workload placement strategy among SMDs, SBS, and MBS was derived. And the optimal computation capability allocation and SMDs' transmit power were obtained. Finally, numerical simulation results are presented. Compared with the benchmark schemes, the proposed EERA scheme can reduce the SMDs' energy consumption significantly.

Appendix

A. Proof of Lemma 1

Substituting equation (4) into equation (19), we rewrite v_k as

$$v_k = \frac{p_{k,tx}}{r_k} = \frac{p_{k,tx}}{B\log_2\left(1 + \left(p_{k,tx}g_k/N_0\right)\right)}. \tag{A.1}$$

The derivative of r_k with respect to transmit power $p_{k,tx}$ is denoted as r_k' and it can be calculated as

$$r_k' = \frac{\mathrm{d}r_k}{\mathrm{d}p_{k,tx}} = \frac{Bg_k}{\left(N_0 + p_{k,tx}g_k\right)\ln 2}. \tag{A.2}$$

Based on equation (A.2), the derivative of v_k is

$$\begin{aligned}\frac{\mathrm{d}v_k}{\mathrm{d}p_{k,tx}} &= \frac{r_k - r_k' p_{k,tx}}{p_{k,tx}^2}\\ &= \frac{B\log_2\left(1 + \left(p_{k,tx}g_k/N_0\right)\right) - Bg_k p_{k,tx}/\left(N_0 + p_{k,tx}g_k\right)\ln 2}{p_{k,tx}^2}.\end{aligned} \tag{A.3}$$

Define Z as

$$Z = B\log_2\left(1 + \frac{p_{k,tx}g_k}{N_0}\right) - \frac{Bg_k p_{k,tx}}{\left(N_0 + p_{k,tx}g_k\right)\ln 2}. \tag{A.4}$$

Furthermore, we get the derivative of Z as

$$\begin{aligned}\frac{\mathrm{d}Z}{\mathrm{d}p_{k,tx}} &= \frac{Bg_k}{\left(N_0 + p_{k,tx}g_k\right)\ln 2} - \frac{Bg_k N_0}{\left(N_0 + p_{k,tx}g_k\right)^2\ln 2}\\ &= \frac{Bp_{k,tx}g_k^2}{\left(N_0 + p_{k,tx}g_k\right)^2\ln 2}.\end{aligned} \tag{A.5}$$

Obviously, $(\mathrm{d}Z/\mathrm{d}p_{k,tx}) \geq 0$ and Z increases with the increase of $p_{k,tx}$. In addition, $Z = 0$ when $p_{k,tx} = 0$. Thus, $Z \geq 0$ exists. Then, we have $(\mathrm{d}v_k/\mathrm{d}p_{k,tx}) \geq 0$ and v_k increases monotonically with the increase of $p_{k,tx}$. The Proof is completed.

B. Proof of Lemma 3

The energy consumption should be semipositive, i.e., $E_k \geq 0$ always holds. According to Lemma 2, we have the following three cases:

(1) In the first case of Lemma 2, i.e., $(p_{k,tx}/r_k) > \kappa C_k f_{k,l}^2$.

From equation (16), we assume

$$\mathrm{E}_k = \alpha_{k,l}\left(D_k C_k \kappa f_{k,l}^2 - \frac{D_k p_{k,tx}}{r_k}\right) + \frac{D_k p_{k,tx}}{r_k} \geq 0. \tag{B.1}$$

Then, we obtain

$$\alpha_{k,l} \leq \frac{D_k p_{k,tx}/r_k}{D_k p_{k,tx}/r_k - D_k C_k \kappa f_{k,l}^2}. \tag{B.2}$$

Define M as

$$M = \frac{D_k p_{k,tx}/r_k}{D_k p_{k,tx}/r_k - D_k C_k \kappa f_{k,l}^2}. \tag{B.3}$$

It is easy to get $M \geq 1$. According to inequality (14f), $\alpha_{k,l} \in [0, 1]$, i.e., $\alpha_{k,l} \leq M$ always holds.

(2) In the second case of Lemma 2, i.e., $(p_{k,tx}/r_k) < \kappa C_k f_{k,l}^2$. It is similar to the first case and there exists $\alpha_{k,l} \in [0, 1]$.

(3) In the third case of Lemma 2, i.e., $(p_{k,tx}/r_k) = \kappa C_k f_{k,l}^2$. It is obvious that $E_k \geq 0$.

Based on above (1), (2), and (3), problem **P1** is feasible. The Proof is completed.

C. Proof of Lemma 8

(1) Problem **P2.1**

Substituting equations (5), (8), and (9) into inequality (22c), we obtain

$$\begin{gathered} t_{k,l}^{\text{trans}} + t_{k,s}^{\text{comp}} \leq T_k, \\ \frac{(\alpha_{k,s} + \alpha_{k,m})D_k}{r_k} + \frac{\alpha_{k,s} D_k C_k}{f_{k,s}} \leq T_k. \end{gathered} \tag{C.1}$$

According to equation (17d), we substitute $\alpha_{k,s} + \alpha_{k,m}$ for $1 - \alpha_{k,l}$ and get

$$\frac{(1 - \alpha_{k,l})D_k}{r_k} + \frac{\alpha_{k,s} D_k C_k}{f_{k,s}} \leq T_k. \tag{C.2}$$

Then, get the inequality about r_k as

$$r_k \geq \frac{(1 - \alpha_{k,l})D_k}{T_k - \alpha_{k,s} D_k C_k / f_{k,s}}. \tag{C.3}$$

In the light of Lemma 1 and Remark 5, smaller v_k induces smaller E_k and smaller r_k induces smaller v_k. Wherefore, r_k should better be small to save energy. $(\alpha_{k,s}/f_{k,s})$ should better be small to make the lower boundary of r_k small. From inequality (17c), we take $f_{k,s}^* = \beta_k F$.

Considering $t_{k,s} \geq t_{k,m}$, from equations (9) and (11), we get

$$t_{k,l}^{\text{trans}} + t_{k,s}^{\text{comp}} \geq t_{k,l}^{\text{trans}} + t_{k,m}^{\text{trans}}. \tag{C.4}$$

Eliminating $t_{k,l}^{\text{trans}}$, we rewrite the equation (C.4) as

$$\begin{gathered} t_{k,s}^{\text{comp}} \geq t_{k,m}^{\text{trans}}, \\ \frac{\alpha_{k,s} D_k C_k}{f_{k,s}} \geq \phi D_k \alpha_{k,m}, \\ \alpha_{k,s} \geq \frac{\phi f_{k,s}}{C_k} \alpha_{k,m}. \end{gathered} \tag{C.5}$$

We take $\alpha_{k,s} = (\phi f_{k,s}/C_k)\alpha_{k,m}$ for getting small $(\alpha_{k,s}/f_{k,s})$.

(2) Problem **P2.2**

Substituting equation (11) into inequality (23c), we get

$$t_{k,l}^{\text{trans}} + t_{k,m}^{\text{trans}} \leq T_k. \tag{C.6}$$

From equations (5) and (10), we rewrite (C.6) as

$$\frac{(\alpha_{k,s} + \alpha_{k,m})D_k}{r_k} + \phi D_k \alpha_{k,m} \leq T_k. \tag{C.7}$$

Based on equation (17d), we substitute $\alpha_{k,s} + \alpha_{k,m}$ for $1 - \alpha_{k,l}$ and get

$$\frac{(1 - \alpha_{k,l})D_k}{r_k} + \phi D_k \alpha_{k,m} \leq T_k. \tag{C.8}$$

And thus, the lower boundary of r_k can be obtained as

$$r_k \geq \frac{(1 - \alpha_{k,l})D_k}{T_k - \phi D_k \alpha_{k,m}}. \tag{C.9}$$

We take $\alpha_{k,m}$ as small as possible to make r_k small for saving energy. Considering $t_{k,s} < t_{k,m}$, from equations (9) and (11), we get

$$t_{k,l}^{\text{trans}} + t_{k,s}^{\text{comp}} < t_{k,l}^{\text{trans}} + t_{k,m}^{\text{trans}}, \tag{C.10}$$

where we obtain the lower boundary of $\alpha_{k,m}$ as

$$\alpha_{k,m} > \frac{\alpha_{k,s} C_k}{\phi f_{k,s}}. \tag{C.11}$$

According to the continuity of $\alpha_{k,m}$, $\forall \delta > 0$, there always exists a $\alpha_{k,m}$ that makes $0 < \alpha_{k,m} - (\alpha_{k,s} C_k / \phi f_{k,s}) < \delta$. Thus, we take $\alpha_{k,m} = (C_k / \phi f_{k,s}) \alpha_{k,s}$. In addition, $\alpha_{k,m}$ decreases with the increase of $f_{k,s}$. From inequality (17c), we take $f^*_{k,s} = \beta_k F$.

Given above cases (1) and (2), both problems **P2.1** and **P2.2** have $\alpha^*_{k,m} = (C_k / \phi f^*_{k,s}) \alpha^*_{k,s}$ and $f^*_{k,s} = \beta_k F$. The Proof is completed.

D. Proof of Theorem 21

Based on Theorem 20, we substitute $\alpha^*_{k,l}$ into equation (16) and get

$$\begin{aligned} E_k &= \left(1 - \frac{T_k r_k}{D_k} + \frac{\phi T_k C_k {r_k}^2}{D_k \left(\phi C_k r_k + C_k + \phi f_{k,s}\right)}\right) \\ &\quad \times \left(D_k C_k \kappa f^2_{k,l} - \frac{D_k p_{k,tx}}{r_k}\right) + \frac{D_k p_{k,tx}}{r_k} \\ &= D_k C_k \kappa f^2_{k,l} + \frac{T_k p_{k,tx} - \kappa C_k f^2_{k,l} T_k r_k}{\phi C_k r_k + C_k + \phi f_{k,s}} \left(C_k + \phi f_{k,s}\right). \end{aligned} \tag{D.1}$$

For simplifying equation (D.1) and getting the optimal transmission power $p^*_{k,tx}$, we define Q as

$$Q = \frac{T_k p_{k,tx} - \kappa C_k f^2_{k,l} T_k r_k}{\phi C_k r_k + C_k + \phi f_{k,s}}. \tag{D.2}$$

In equation (D.1), a smaller Q induces a smaller E_k. Thus, we will try to minimize Q by optimizing $p_{k,tx}$.

Furthermore, for simplifying the expression of Q, we set $A = C_k + \phi f_{k,s}$ and $D = \kappa C_k f^2_{k,l} T_k$. Thus, Q can be rewritten as

$$Q = \frac{T_k p_{k,tx} - D r_k}{\phi C_k r_k + A}. \tag{D.3}$$

Then, the derivative of Q can be calculated as

$$\frac{\mathrm{d}Q}{\mathrm{d}p_{k,tx}} = \frac{A T_k + T_k \phi C_k r_k - \left(AD + \phi C_k T_k p_{k,tx}\right) r'_k}{\left(\phi C_k r_k + A\right)^2}. \tag{D.4}$$

Define M as

$$M = A T_k + T_k \phi C_k r_k - \left(AD + \phi C_k T_k p_{k,tx}\right) r'_k. \tag{D.5}$$

The second derivative of r_k is computed as

$$\frac{\mathrm{d}^2 r_k}{\mathrm{d}{p_{k,tx}}^2} = -\frac{B g^2_k}{\left(N_0 + p_{k,tx} g_k\right)^2 \ln 2}. \tag{D.6}$$

Obviously, the second derivative of r_k is negative. The derivative of M is obtained as

$$\frac{\mathrm{d}M}{\mathrm{d}p_{k,tx}} = -\left(AD + \phi C_k T_k p_{k,tx}\right) r''_k. \tag{D.7}$$

In the light of equation (D.6), equation (D.7) shows that the first derivative of M is positive. Hence, M increases monotonously with the increase of $p_{k,tx}$. When $p_{k,tx} = 0$,

$$M(0) = A T_k - \frac{A D B g_k}{N_0 \ln 2} = A T_k \left(1 - \frac{\kappa C_k f^2_{k,l} B g_k}{N_0 \ln 2}\right). \tag{D.8}$$

For simplifying expressions, define $a = (\kappa C_k f^2_{k,l} B g_k / N_0 \ln 2)$ and there exists two cases according to the value of a.

(1) When $M(0) \geq 0$, i.e., $a \leq 1$, this case does not exist. We prove this case by contradiction in the following. Firstly, suppose this case is feasible, then, we have

$$\begin{aligned} \frac{\kappa C_k f^2_{k,l} B g_k}{N_0 \ln 2} &\leq 1, \\ \kappa C_k f^2_{k,l} &\leq \frac{N_0 \ln 2}{B g_k}. \end{aligned} \tag{D.9}$$

According to $v_k < \kappa C_k f^2_{k,l}$, we get

$$v_k < \kappa C_k f^2_{k,l} \leq \frac{N_0 \ln 2}{B g_k}. \tag{D.10}$$

From equations (4) and (19), we obtain

$$\begin{aligned} \frac{p_{k,tx}}{r_k} &< \frac{N_0 \ln 2}{B g_k}, \\ \frac{p_{k,tx}}{B \log_2\left(1 + \left(p_{k,tx} g_k / N_0\right)\right)} &< \frac{N_0 \ln 2}{B g_k}, \\ \frac{p_{k,tx} g_k}{N_0} &< \ln\left(1 + \frac{p_{k,tx} g_k}{N_0}\right). \end{aligned} \tag{D.11}$$

For simplifying the expression, we set $x = (p_{k,tx} g_k / N_0)$ and $\mathrm{y} = x - \ln(1 + x)$. Then, the first derivative of y is given by

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{x}{1 + x}, \tag{D.12}$$

where $x = (p_{k,tx} g_k / N_0) \geq 0$ that makes $(\mathrm{d}y/\mathrm{d}x) \geq 0$. Thus, y increases with the increase of x. Moreover, $y|_{x=0} = 0$. Wherefore, $\mathrm{y} \geq 0$ and $x \geq \ln(1 + x)$, i.e., $(p_{k,tx} g_k / N_0) \geq \ln(1 + (p_{k,tx} g_k / N_0))$. It is in conflict to inequality (D.11). Thus, this case does not exist

(2) When $M(0) < 0$, i.e., $a > 1$, we define p that makes $(\mathrm{d}Q/\mathrm{d}p_{k,tx})|_{p_{k,tx}=p} = 0$. Q decreases when $p_{k,tx} \in (0, p)$ and

increases when $p_{k,tx} \in (p, +\infty)$. According to $v_k < \kappa C_k f_{k,l}^2$, we have

$$p_{k,tx}^* = \begin{cases} p, & \frac{p}{r_k} < \kappa C_k f_{k,l}^2, \\ v_k^{-1}\left(\kappa C_k f_{k,l}^2\right), & \frac{p}{r_k} \geq \kappa C_k f_{k,l}^2. \end{cases} \tag{D.13}$$

Wherefore, from Lemma 1, $p_{k,tx}^* = \min\{p, v_k^{-1}(\kappa C_k f_{k,l}^2)\}$. The Proof is completed

E. Proof of Theorem 22

When tasks are executed entirely by local devices, substituting $\alpha_{k,l} = 1$ into equation (1), we obtain the execution latency $t_{k,l}^{\text{all}}$ as

$$t_{k,l}^{\text{all}} = \frac{D_k C_k}{f_{k,l}}. \tag{E.1}$$

When tasks are offloaded entirely, according to $v_k = \kappa C_k f_{k,l}^2$, the transmission power is $v_k^{-1}(\kappa C_k f_{k,l}^2)$. Substitute it into equation (4), and get the offloading rate as

$$r_k = B \log_2\left(1 + \frac{v_k^{-1}\left(\kappa C_k f_{k,l}^2\right) g_k}{N_0}\right). \tag{E.2}$$

Then, the offloading latency $t_{k,l}^{\text{trans}}$ can be obtained as $t_{k,l}^{\text{trans}} = (D_k/r_k)$. Substituting $t_{k,l}^{\text{trans}}$ into equations (9) and (11), we get the offloading latency $t_{k,\text{off}}$ as

$$t_{k,\text{off}} = \max\left\{t_{k,s}, t_{k,m}\right\}. \tag{E.3}$$

We decompose equation (E.3) into two cases, i.e., $t_{k,s} \geq t_{k,m}$ and $t_{k,s} < t_{k,m}$.

When $t_{k,s} \geq t_{k,m}$, i.e., the delay from SBS is larger than the delay from MBS, $t_{k,\text{off}} = t_{k,s}$ and minimizing $t_{k,\text{off}}$ becomes minimizing $t_{k,s}$. From equation (17d), we obtain

$$\alpha_{k,s} \geq \frac{\phi f_{k,s}}{C_k + \phi f_{k,s}}. \tag{E.4}$$

According to equation (8), a smaller $\alpha_{k,s}$ brings a smaller $t_{k,s}$ and thus a smaller $t_{k,\text{off}}$. Wherefore, we take

$$\alpha_{k,s} = \frac{\phi f_{k,s}}{C_k + \phi f_{k,s}}. \tag{E.5}$$

Substituting equation (E.5) into equation (17d), we get

$$\alpha_{k,m} = \frac{C_k}{C_k + \phi f_{k,s}}. \tag{E.6}$$

When $t_{k,s} < t_{k,m}$, interestingly, it is similar to $t_{k,s} \geq t_{k,m}$ and we also get equations (E.5) and (E.6). Based on the above derivations, we attain that $t_{k,\text{off}}$ is smallest when $t_{k,s} = t_{k,m}$.

Substituting equations (E.5) and (E.6) into equation (E.3), we obtain the optimal latency of total offloading $t_{k,\text{off}}^*$ as

$$t_{k,\text{off}}^* = \frac{D_k}{B\log_2\left(1 + \left(v_k^{-1}\left(\kappa C_k f_{k,l}^2\right) g_k / N_0\right)\right)} + \frac{\phi D_k C_k}{C_k + \phi f_{k,s}}, \tag{E.7}$$

where we take $f_{k,s} = \beta_k F$ to minimize $t_{k,\text{off}}^*$.

Wherefore, (1) when $t_{k,l}^{\text{all}} \leq t_{k,\text{off}}^*$, we choose entire local execution to minimize the execution latency, i.e., $\alpha_{k,l}^* = 1$, $\alpha_{k,s}^* = 0$, $\alpha_{k,m}^* = 0$, $p_{k,tx}^* = 0$, and $f_{k,s}^* = 0$; (2) when $t_{k,l}^{\text{all}} > t_{k,\text{off}}^*$, we choose entire offloading, i.e, $\alpha_{k,l}^* = 0$, $\alpha_{k,s}^* = ((\phi f_{k,s})/(C_k + \phi f_{k,s}))$, $\alpha_{k,m}^* = ((C_k)/(C_k + \phi f_{k,s}))$, $p_{k,tx}^* = v_k^{-1}(\kappa C_k f_{k,l}^2)$, and $f_{k,s}^* = \beta_k F$.

Data Availability

The data used to support the findings of this study are available from Yongsheng Pei (peiys@shu.edu.cn).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[4] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for internet of things: a primer," *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, 2018.

[5] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[6] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: partial computation offloading using dynamic voltage scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 1–4282, 2016.

[7] L. Li, Z. Kuang, and A. Liu, "Energy efficient and low delay partial offloading scheduling and power allocation for MEC," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

[8] T. Q. Thinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 1–3584, 2017.

[9] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[10] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[11] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-efficient resource allocation for multi-user mobile edge computing," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–7, Singapore, Singapore, December 2017.

[12] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington, DC, USA, December 2016.

[13] X. Wang, Y. Cui, Z. Liu, J. Guo, and M. Yang, "Optimal resource allocation for multi-user MEC with arbitrary task arrival times and deadlines," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

[14] Y. Cui, W. He, C. Ni, C. Guo, and Z. Liu, "Energy-efficient resource allocation for cache-assisted mobile edge computing," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 640–648, Singapore, Singapore, October 2017.

[15] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2018.

[16] R. Yu, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, and D. H. K. Tsang, "Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 72–84, 2018.

[17] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685–2700, 2013.

[18] L. Liu and Q. Fan, "Resource allocation optimization based on mixed integer linear programming in the multi-cloudlet environment," *IEEE Access*, vol. 6, pp. 24533–24542, 2018.

[19] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10660–10675, 2017.

[20] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: an auction-based profit maximization approach," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2082–2091, 2017.

[21] E. El Haber, T. M. Nguyen, and C. Assi, "Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3407–3421, 2019.

[22] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IOT fog networks: a joint optimization approach combining stackelberg game and matching," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, 2017.

[23] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, April 2016.

[24] K. Zhang, Y. Mao, S. Leng et al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[25] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12313–12325, 2018.

[26] Q. Han, B. Yang, G. Miao, C. Chen, X. Wang, and X. Guan, "Backhaul-aware user association and resource allocation for energy-constrained hetnets," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 580–593, 2017.

[27] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2017.

[28] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11339–11351, 2017.

[29] Y. Lan, X. Wang, C. Wang, D. Wang, and Q. Li, "Collaborative computation offloading and resource allocation in cache-aided hierarchical edge-cloud systems," *Electronics*, vol. 8, no. 12, p. 1430, 2019.

[30] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, 2016.

[31] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, 2018.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, U.K., 2004.

[33] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[34] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017.