Service-Oriented Energy-Latency Tradeoff for IoT Task Partial Offloading in MEC-Enhanced Multi-RAT Networks

Meng Qin, *Member, IEEE*, Nan Cheng, *Member, IEEE*, Zewei Jing, Tingting Yang, *Member, IEEE*, Wenchao Xu, *Member, IEEE*, Qinghai Yang, *Member, IEEE*, and Ramesh R. Rao, *Fellow, IEEE*

***Abstract*—The upcoming development of the 5G network is envisioned to offer various types of services like virtual reality/augmented reality and autonomous vehicles applications with low-latency requirements in Internet-of-Things (IoT) networks. Mobile-edge computing (MEC) has become a promising solution for enhancing the computation capacity of mobile devices at the edge of the network in a 5G wireless network. Additionally, multiple radio access technologies (multi-RATs) have been verified with the potential in lowering the transmission latency and energy consumption, while improving the Quality of Services (QoS). Benefiting from the cooperation of multi-RATs, large latency-sensitive computing service tasks (L2SC) can be offloaded by different RATs simultaneously, which has great practical significance for data partitioned oriented applications with large task sizes. In this article, to enhance the L2SC offloading services for satisfying low-latency requirements with low energy consumption, we investigate the energy-latency tradeoff problem for partial task offloading in the MEC-enhanced multi-RAT network, considering the limitation of energy and computing in capability-constrained end devices in IoT networks. Specifically, we formulated the L2SC task computation offloading problem to minimize the weighted sum of the latency cost and the energy consumption by jointly optimizing the local computing frequency, task splitting, and transmit power, while guaranteeing the stringent latency requirement and the residual energy constraint. Due to the nonsmoothness and nonconvexity of the formulated problem with high complexity, we convert the tradeoff problem into a smooth biconvex problem and propose an alternate convex search-based algorithm, which can greatly reduce the computational complexity. Numerical simulation results show the effectiveness of the proposed algorithm with various performance parameters.**

Manuscript received March 23, 2020; revised June 8, 2020 and July 28, 2020; accepted August 7, 2020. Date of publication August 12, 2020; date of current version January 22, 2021. This work was supported in part by China Postdoctoral Science Foundation under Grant 2019TQ0210 and Grant 2019M663015; and in part by NSFC under Grant 61971327, Grant 61671062, and Grant 61801365. *(Corresponding author: Tingting Yang.)*

Meng Qin is with School of Electronics and Computer Engineering, Peking University, Beijing 100871, China, and also with Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: mengqin@stu.xidian.edu.cn).

Nan Cheng, Zewei Jing, and Qinghai Yang are with the State Key Laboratory of ISN, School of Telecommunications Engineering and the Collaborative Innovation Center of Information Sensing and Understanding, Xidian University, Xi'an 710071, China (e-mail: nancheng@xidian.edu.cn; zwjing@stu.xidian.edu.cn; qhyang@xidian.edu.cn).

Tingting Yang is with the School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan 523000, China (e-mail: yangtingting820523@163.com).

Wenchao Xu is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong (e-mail: wenchao.xu@polyu.edu.hk).

Ramesh R. Rao is with the CALIT2, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: rrao@ucsd.edu).

Digital Object Identifier 10.1109/JIOT.2020.3015970

***Index Terms*—Internet of Things (IoT), latency-sensitive services, mobile-edge computing (MEC), multiple radio access technology (multi-RAT), partial offloading, resource allocation, task splitting.**

I. Introduction

NOWADAYS, the rapid growth of the computation-intensive and latency-sensitive mobile applications in the Internet-of-Things (IoT) networks, such as face recognition, virtual reality (VR), augmented reality (AR), and intelligent video accelerations, has posed great challenges to the traditional cellular networks and the resource-limited user equipments (UEs) with high energy consumption and high-latency sensitivity [1]–[5]. To address the limitations of UE's battery energy, computation capacity and reduce end-to-end latency, mobile-edge computing (MEC) has been envisioned as a potential computation paradigm in both academia and industry, which can meet the demands of lightweight but latency-sensitive IoT tasks that cannot be provisioned with the current wireless networks [6], [7].

Smart computation applications have recently gained noticeable attention and bring a huge challenge to traditional wireless networks in the near future. The fast-developing IoT technologies bring a large number of devices to the Internet. For L2SC tasks, the latency is a big issue for the L2SC services. With the increasing service applications, the computing offloading has been playing a great role in bringing new experiences to UEs by fulfilling the low-latency requirements, considering the limited resources (communication, computing, storage, etc.) [8]–[12]. Traditional cloud computing can provide powerful computing capability but also introduce additional latency due to cloud architecture and geographical distance. Hence, computing capabilities are transferred to the edge of wireless networks with the aid of the new MEC architecture in IoT networks, which can provide computation ability for the energy-sensitive tasks with lower latency. By pushing the powerful computation and caching capacity at the edge of radio access networks, the less-capable UE can offload parts of its tasks to MEC servers for achieving better service performance and reducing energy consumption. Considering the limitation of computation and energy resources, MEC is gaining great interest from the research

areas. In particular, for L2SC services with large task size and high communication requirements, task computation offloading with MEC servers, will be a key element to bring good experiences to UEs. However, it is still a great challenge to manage MEC resources with performance maximization while considering UEs' dynamic requirements of latency and energy, especially for large task applications.

To cope with the above challenges, research works related to the MEC resource allocation have begun to study the task offloading and resource allocation decisions in wireless networks from two perspectives: 1) binary task offloading [5]–[13] and 2) partial task offloading [14]–[15], respectively. To make the optimization of the latency or battery energy consumption of the application, different task offloading policies are also studied considering the single-user scenario [16], [17] and multiuser scenario with MEC servers [5], [18], [19], which give different research focuses for specific scenarios. However, most of the related research works can only tackle the latency or energy performance limitation, ignoring the internal interactions between the latency and energy especially for actual situations of different UEs. In addition, they only consider the traditional single radio access technology (RAT) scenario, where the UE can only connect to one RAT (such as LTE, WiFi, WiMax, etc.). In such a single-RAT network, the performance gain, such as lower latency and less energy consumption acquired by task offloading can be easily influenced by the poor radio condition. For instance, the WiFi can give efficient supply for the high rate service but is limited in radio coverage and mobility support, while LTE can sustain the user mobility with larger coverage [20], [21]. Therefore, it is essential to integrate these heterogeneous RATs so as to provide a more flexible and steadier radio condition for task offloading.

Multi-RAT service has been recognized as a promising solution for supporting the transmission of the throughput-demanding tasks in IoT scenarios. Multi-RAT service enables one UE (or IoT devices) to maintain multiple simultaneous network paths between the MEC server and the UE by employing different access points (APs) in heterogeneous wireless networks [22]. Moreover, the task stream of the UE can be split into multiple substreams of different sizes which are mapped onto multiple parallel RAT channels adaptively. Since the data rate is a logarithm function in terms of the transmit power, the power consumption increases exponentially with the data rate. In the conventional wireless network with single-RAT, it will cause more energy consumption and latency, especially for computation-intensive tasks with big data size. However, due to the superiorities of parallel transmission and power control in multiple RAT (multi-RAT) networks, much lower energy consumption and latency performance benefit can be achieved for task offloading in MEC-enabled networks. It is worth mentioning that the traffic splitting problems in multi-RAT networks have been investigated widely in [23] and [24]. However, all existing works by now mainly focus on the throughput-oriented objective such that they cannot meet the requirement of stringent latency and low energy consumption for task computation. The MEC server with single access RAT leads to suboptimal utilization of the overall network performance. In 5G wireless network scenarios, UEs may be exposed to the multi-RATs environment in wireless networks. With the added benefits of integrated multiple RAT support (WiFi, LTE, and 5G), the new design of task offloading in multi-RAT wireless networks should be fully addressed while considering the limitations of energy cost and service latency requirements.

Motivated by the above analysis, we consider a task computation offloading MEC architecture, which integrates the multi-RAT technologies in IoT networks for L2SC applications with large task sizes and requirements of low latency. The multi-RAT technologies provide the low-latency transmission while the MEC server supports the efficient task computation for IoT devices. The single-user task offloading is taken as the computation offloading model to illustrate the L2SC application offloading process. Owing to the fact that a lower latency leads to higher energy consumption, there should exist a tradeoff between them. Therefore, we aim to fully exploit the computation resources and reduce energy consumption, making a tradeoff between the two terms for the partial task offloading. Different from the work in the single-RAT networks [25], in this way, not only the offloading fraction of the task for edge computing but also the fractions mapped onto multiple RATs for parallel transmission in the multi-RAT networks, should be determined as well. In addition, aided by the dynamic voltage scaling technique, a proper local computing frequency can also be regulated to balance the latency and energy consumption. The multi-RAT task offloading will be a highly efficient policy for end devices with huge task size and low-latency requirements, which is suitable with bit independence.

In this article, we investigate the energy-efficient task offloading problem for L2SC services in MEC-enhanced multi-RAT wireless networks, considering the limitations of stringent latency and residual battery energy. Specifically, to fully explore the benefits of both MEC and Multi-RAT networks, we formulate the energy-latency tradeoff problem of partial task offloading as the weighted sum of the latency cost and energy consumption optimization problem, by jointly optimizing the local computing frequency, task splitting, and transmit power. In particular, the formulated mathematical problem is nonsmooth and nonconvex, which is quite difficult to solve. Then, we convert the tradeoff problem into a smooth biconvex problem and we propose an alternate convex search (ACS)-based approach, which can greatly reduce the computational complexity and algorithm execution time. The main contributions of this article are outlined as follows.

1) We propose a novel task computation offloading the MEC framework that integrates the multi-RAT technologies in IoT networks, in which the multi-RAT technologies can provide low-latency transmission while the MEC server supports the efficient task computation to satisfy the UEs' Quality-of-Service (QoS) requirements with L2SC services. We also characterize the influences among the UE's battery energy, latency requirements of L2SC services, and the total network cost.
2) We formulated the latency and energy consumption minimization problem by jointly optimizing the local computing frequency, task splitting, and transmit power

while guaranteeing the stringent latency requirement and the residual energy constraint. In particular, due to the nonsmoothness and nonconvexity of the formulated problem with high complexity, we propose an ACS-based approach to solve the problem by converting the tradeoff problem into a smooth biconvex problem, which can greatly reduce the computational complexity and algorithm execution time.

3) We achieved the energy-latency tradeoff performance for partial task offloading with the L2SC services, in which it can provide guidelines for dynamic resource allocation, considering the different QoS requirements for latency-sensitive and energy-sensitive resource allocation. It is practical in real MEC systems. In addition, numerical simulation results are made to demonstrate the effectiveness of our algorithm in terms of execution latency and offloading energy efficiency.

The remainder of this article is organized as follows. The related work is presented in Section II. The network model and problem formulation are given in Sections III and IV, respectively. The proposed ACS-based algorithm and energy-latency tradeoff performance analysis are given in Section V. The proposed algorithm is verified by simulation results in Section VI. Conclusions are drawn in Section VII.

II. Related Work

With the increasing number of powerful mobile devices, many newly emerging mobile applications (such as AR/VR and smart factory) are anticipated to be among the most demanding killer applications in 5G networks, which have made great challenges with high latency and energy requirements. In order to tackle such grand challenges, many researchers make great efforts to meet these demands in both academia and industry by facilitating flexible allocating resources (bandwidth, storage, energy, computing, etc.) for low latency and low energy consumption.

As the computing capacity is important to computing services, it is necessary to migrate tasks from UEs to resourceful edge servers for increasing the computation capacity while saving UE battery energy [26]. For this propose, MEC enables UEs to access edge servers with dynamic fashion for task offloading in [13]–[19] and in [25]–[26]. An alternating direction method of multipliers decomposition technique was developed to make the computing mode selection in [13]. Wang *et al.* [14] investigated the partial computation offloading policy with objectives of energy consumption and latency execution minimization. A reinforcement learning-based offloading scheme was proposed to select the edge device without prior knowledge in [15]. In addition, an energy-efficient computing framework was proposed considering a single-user system in [16]. A joint task offloading and transmit power allocation scheme was proposed in [17]. Ning *et al.* [18] studied the single-user computation offloading problem and considering both the cloud computing and edge computing. In [19], a unified MEC with wireless power transfer was studied to improve the MEC performance in energy. To sum up, energy is a key issue for IoT devices in task offloading processes, so wireless power transfer and energy harvesting technology are considered to prolong the battery life of mobile devices in MEC in [13]–[19]. In addition, the works in [20]–[22] indicate that the multi-RAT technology can improve network resource use efficiency which can be applied in MEC-enabled IoT networks. Furthermore, an energy-aware offloading scheme was proposed with the residual energy of smart devices' battery concept in [25], which is similar to our work. To balance the tradeoff between the high cost of cloud service and the limited capacity of fog, a model-based planning method was proposed to find the optimal resource allocation policy in [26]. To minimize the overall latency of UEs, a jointly offloading optimization algorithm was proposed in nonorthogonal multiple access-enabled networks in [27]. Conclusively, for the L2SC application, energy-efficient offloading of computation will be a key element to bring new experiences to mobile devices by offloading part of the tasks to MEC servers, aiming to reduce the latency. In order to fulfill low latency, the MEC paradigm will play a key role in 5G networking for latency-sensitive computing services such as AR/VR and autonomous vehicles' applications.

Additionally, energy consumption has also been a key issue for capacity-constrained batteries of UEs when offloading computation tasks to MEC servers [28]. To reduce energy consumption, a jointly online task and CPU-cycle frequency allocation algorithm were investigated, considering highly dynamic task arrival and wireless channel states in [29]. A cloud-assisted MEC framework was proposed to meet the energy resource provision and dynamics of service requests in [30]. To support the increasing data traffic with different QoS requirements, a MEC-based dynamic spectrum management approach was proposed in [31]. For better utilizing the energy, Zhang *et al.* [32] investigated the tradeoff between the execution latency and energy consumption with the aid of energy harvesting capability in wireless networks. In particular, the new concept of residual energy of users' battery was introduced that defines as the weighting factor of energy consumption and latency to make energy-efficient offloading decisions. Furthermore, the multiple MEC severs scenario was also investigated to provide cost-effective offloading performance for UEs. Aided by the minority game theory, a distributed MEC server activation computation offloading mechanism was proposed in [33]. An alternating direction method of a multipliers-based algorithm was provided to give computation offloading decisions in [34]. Sun *et al.* [35] further studied the sum of a computation efficiency maximization problem with local computing. According to the above-mentioned works, the binary task offloading schemes were developed in the traditional single RAT scenario. Only communication resources or computing resources are considered for offloading decisions in conventional multi-RAT networks.

In particular, users with powerful end devices can also benefit from offloading the computation tasks to the MEC server with a partial offloading mode [36], [37]. Considering different system design objectives and with the increasing power capacity of end UEs, a partial computation offloading algorithm was proposed by jointly optimizing the transmit power, offloading ratio, and the computation speed of UEs in [14].

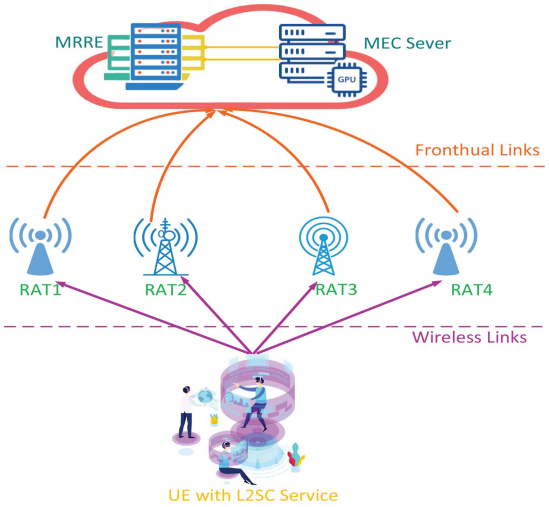


Fig. 1. System model of IoT with MEC-enhanced multi-RAT networks.

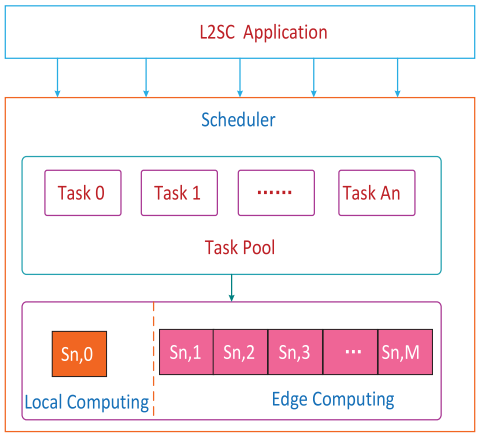


Fig. 2. L2SC task spitting model in multi-RAT networks.

Furthermore, aided with artificial intelligence, a reinforcement learning-based offloading scheme was proposed according to the predicted network information in [15]. The aforementioned works mainly focus on computing offloading with a single RAT scenario. To further make computing offloading in a cost-effective way, the MEC paradigm with multi-RAT scenarios should also be drawing increasing attention.

Since the computation capabilities of MEC servers are relatively stronger than that of end devices, it can provide considerable capabilities to UEs and empower them with powerful resources. In addition, along with powerful artificial intelligence, smart devices with increasing ability are feasible to process one part of L2SC tasks [7], [8]. In particular, the 5G wireless network is envisioned to be a new architecture of tightly integrating multi-RATs, which allows UEs to reap the various benefits offered by each RAT. Hence, the dynamic offloading policy requires careful design to meet the various service requirements for energy and latency in multi-RAT wireless networks.

III. System Model

A. Network Model

We consider an IoT scenario with MEC servers in an uplink multi-RAT network, as shown in Fig. 1, which consists of the set $\mathcal{M} = \{1, \ldots, M\}$ of RATs and one UE n (or IoT devices) with L2SC services, similar with the related works [16], [17]. All RATs are connected with the MEC-enhanced multiradio resource management entity (MRRME) by low-latency fronthaul links. The MRRME provides joint resource management capability for different RATs while the MEC server provides the edge computing service for the UE tasks. We assume that the UE has a computation-intensive task denoted by $A_n(C_n, D_n, T_n^{\max})$ to execute, where C_n, D_n, and $T_n^{\max}$ represent the task size (in bits) [8], the computation workload (namely, the CPU cycles needed for executing this task), and the stringent latency requirement of task A_n, respectively. We also assume that the task A_n is of bit independence and can be partitioned into two chunks of arbitrary size, one of which is executed locally by the UE itself and the other can be offloaded to the MEC server. Moreover, the UE is equipped with the multi-RAT connectivity capability which enables the UE to further split the offloading part of the task to several subchunks and offload them by these RATs independently. It is noted that different RATs generally operate over different spectrum bands (e.g., 2.4 GHz for WiFi with high speed and 1.8–2.3 GHz for LTE networks with larger coverage).[1] Hence, there does not exist cross-RAT interference in this system [38]. From the long timescale perspective, it is reasonable to assume that the channel gain of different RATs is also a random variable and independent identically distributed across time slots. For UE n, the perfect uplink channel state information to all RATs is available for the UE. When the CSI at the RATs is imperfect due to some CSI estimation errors, the task offloading performance in this framework may degrade (however, the imperfect CSI scenario is out of the scope of this article, which can be obtained by channel estimation methods).

Let $\mathbf{p} = [p_{n,m}]_{m\in\mathcal{M}}$ denote the transmit power from the UE to different RATs. The uplink transmit rate between the UE and RAT m can be given by

$$r_{n,m} = B_{n,m} \log\left(1 + p_{n,m}\Gamma_{n,m}\right) \tag{1}$$

where $B_{n,m}$ is the frequency bandwidth allocated by RAT m to the UE and $\Gamma_{n,m} = g_{n,m}/\sigma_{n,m}^2$ is the ratio of the channel gain to noise power (CNR) at the receiver of RAT m.

B. L2SC Task Splitting Model

To reduce the latency and save more energy, the UE will partition one part of its task to offload to the MEC server, as shown in Fig. 2. Furthermore, benefitting from the multi-RAT capability, the UE can offload through multiple RATs in parallel. To this end, the UE needs to further determine the amount of the task to offload to each RAT [39], [40]. Let $0 \le s_{n,0} \le 1$ be the fraction of the task that is executed locally, and $0 \le s_{n,m} \le 1$ be the fraction that is offloaded to RAT m. We collect all fractions of the task by $\mathbf{s} = [s_{n,m}]_{m\in 0,\mathcal{M}}$. In

[1]We assume that the UE can connect to the RAT with no latency. The latency to access channel is not considered in this article.

order to simplify the analysis, we give the assumption that there is full granularity in the data partition. So the application could be partitioned into subsets of any size, which is suitable for L2SC applications in practice. Accordingly, the optimal solution in this article could be served as a performance upper bound of realistic offloading strategies. Then, the sum of all fractions of the task should be equal to 1, i.e.,

$$s_{n,0} + \sum_{m \in \mathcal{M}} s_{n,m} = 1 \tag{2}$$

where $\sum_{m \in \mathcal{M}} s_{n,m}$ is the total of the fractions which need to be offloaded to the MEC server.

C. Energy Consumption and Latency Model

1) Local Computing: We assume that the task has the uniform workload. In other words, the CPU cycles needed for executing each bit of the task are identical. Define $f_n^{l,\min} \leq f_n^l \leq f_n^{l,\max}$ as the local computation ability (namely, the CPU frequency) of UE n. Then, the local computation latency (in seconds) for task A_n can be expressed as

$$t_n^{\text{local}} = \frac{s_{n,0} D_n}{f_n^l} \tag{3}$$

and the energy consumption (in Joule) for the local task computation is

$$e_n^{\text{local}} = \kappa \left(f_n^l\right)^2 s_{n,0} D_n \tag{4}$$

where $\kappa = 10^{-26}$ is the energy coefficient depending on the chip architecture [41].

We note that computation frequency f_n^l influences both the computation latency and energy consumption. Benefitting from the dynamic voltage and frequency scaling (DVFS) technology, the UE can choose an appropriate f_n^l to make a tradeoff of the above two terms.

2) Edge Computing: For the offloading fractions of the UE task, the total latency cost mainly includes three parts: 1) the task input uploading; 2) the task output downloading; and 3) the task execution in the MEC server. Due to the fact that the size of task outcome is much smaller than that of task input, the downloading latency is ignored similar to [25] and [34]. In the multi-RAT networks, the UE can upload its task by different RATs simultaneously. Therefore, the uploading latency between the UE and RAT m can be given by

$$t_{n,m}^{\text{up}} = \frac{s_{n,m} C_n}{r_{n,m}}. \tag{5}$$

Since the uploading process between the UE and different RATs is in parallel, the actual uploading latency for subtask $s_{n,m}$ of task A_n is $t_n^{\text{up}} = \max_{m \in \mathcal{M}} \{t_{n,m}^{\text{up}}\}$.

The task execution latency in the MEC server for task A_n is

$$t_n^c = \frac{\sum_{m \in \mathcal{M}} s_{n,m} D_n}{f_n^c} \tag{6}$$

where f_n^c is the MEC computation frequency allocated by the MEC server to the UE task, which is considered as a fixed value in this article.

Consequently, the total offloading latency for task A_n can be thus given by

$$t_n^{\text{off}} = t_n^{\text{up}} + t_n^c. \tag{7}$$

Based on the uploading latency, the energy consumption for task uploading can be expressed as

$$e_n^{\text{off}} = \sum_{m \in \mathcal{M}} t_{n,m}^{\text{up}} p_{n,m}. \tag{8}$$

Furthermore, since the task A_n has a stringent latency requirement, the total latency for the task should be no larger than $T_n^{\max}$ as

$$t_n^{\text{total}} = \max\left\{t_n^{\text{local}}, t_n^{\text{off}}\right\} \leq T_n^{\max}. \tag{9}$$

Remarks: For the simplicity of analysis, we assume that the MEC server will cache all of the subchunks of the task input until the last subchunk is received, and then it will recombine these subchunks (i.e., $\sum_{m \in \mathcal{M}} s_{n,m}$) and execute by using the CPU frequency f_n^c. In addition, we ignore the downloading latency for our problem keeping in view that computation results have relatively smaller sizes, which has a minimal effect on the latency due to the powerful multi-RAT downlink transmission capacity with high transmission rate. This assumption has also been made in literatures [19], [37].

IV. Problem Formulation

In this article, we aim to address the tradeoff between energy consumption and the latency cost for partial task offloading in multi-RAT networks. To proceed with constructing our objective function, we first introduce a weighting factor $0 \leq w_n \leq 1$, which can be used to adjust the weights of the energy consumption and latency cost according to the UE preference. Moreover, similar to [25], we combine the residual energy effect into the weighting factor w_n. Specifically, the ratio of the residual energy to the battery capacity is denoted by $\gamma_n^E = E_n^R / E_n^{\max}$, where E_n^R and $E_n^{\max}$ are the residual energy and battery capacity for the UE, respectively. Then, the new weighting factor can be defined as $w_n' = w_n \gamma_n^E$. Hence, our objective function can be formulated as the weighted sum of the energy consumption and latency cost, which is given by

$$Z\left(f_n^l, \mathbf{p}, \mathbf{s}\right) = w_n' t_n^{\text{total}} + \left(1 - w_n'\right) \alpha e_n^{\text{total}} \tag{10}$$

where $e_n^{\text{total}} = e_n^{\text{local}} + e_n^{\text{off}}$ and α is the normalizing factor which can help to implement the unitless combination of the two terms. In particular, the normalized weighting factor given in formula (10*) not only allows to be adjusted by the IoT device according to its specific preference but also reflects the service condition of the battery. Since lower latency requires more energy consumption, the more residual energy E_n^R, the more degree of freedom or probability we have to make better latency reduction, so that the optimization target, namely, the weighted latency plus energy consumption can be further depressed. Mathematically, a larger $E_n^R / E_n^{\max}$ will lead to lower latency in the minimization optimization problem.

As a result, the problem of the tradeoff between the energy consumption and latency cost (TECLC) can be formulated as

$$\begin{aligned}
&\min_{\{f_n^l,\mathbf{p},\mathbf{s}\}} Z(f_n^l,\mathbf{p},\mathbf{s})\\
&\text{s.t. C1}: t_n^{\text{total}} \le T_n^{\max}\\
&\quad \text{C2}: e_n^{\text{total}} \le E_n^R\\
&\quad \text{C3}: s_{n,0} + \sum_{m\in\mathcal{M}} s_{n,m} = 1\\
&\quad \text{C4}: 0 \le p_{n,m} \le p_{n,m}^{\max}, m \in \mathcal{M}\\
&\quad \text{C5}: 0 \le s_{n,m} \le 1, m \in \mathcal{M}\cup\{0\}\\
&\quad \text{C6}: f_n^{l,\min} \le f_n^l \le f_n^{l,\max}
\end{aligned} \tag{11}$$

where C1 corresponds to the stringent latency constraint for the UE task, C2 represents that the total energy consumption of the UE should not exceed its residual energy, C3 represents that the sum of all fractions of the UE task should equal to 1, C4 is the transmit power constraint between the UE and different RATs, and C5 and C6 represent the domains for $\mathbf{s}$ and f_n^l, respectively.

The difficulties of solving the TECLC problem mainly lie in the nonconvexity and nonsmoothness of the objective function, the stringent latency constraint C1, and energy consumption constraint C2. To obtain the globally optimal solution is impractical by resorting to the conventional optimization techniques. Instead, we seek for the suboptimal solution in this article. Concretely, we first introduce an auxiliary variable T_n to convert the original TECLC problem to a smooth optimization problem, and then we transform the converted problem to a biconvex problem by variable substitution. Finally, we propose an ACS-based algorithm, which converges to a suboptimal solution to the original TECLC problem.

V. Algorithm Design for Partial Offloading in Multi-RAT Networks

A. Problem Transformation

The TECLC problem can be equivalently converted into a smooth optimization problem by introducing an auxiliary variable T_n that should satisfy $t_n^{\text{total}} \le T_n \le T_n^{\max}$, which can be expressed as

$$\begin{aligned}
&\min_{\{T_n,f_n^l,\mathbf{p},\mathbf{s}\}} w_n' T_n + (1-w_n')\alpha e_n^{\text{total}}\\
&\text{s.t. C2}: e_n^{\text{total}} \le E_n^R\\
&\quad \text{C3}: s_{n,0} + \sum_{m\in\mathcal{M}} s_{n,m} = 1\\
&\quad \text{C4}: 0 \le p_{n,m} \le p_{n,m}^{\max}, m \in \mathcal{M}\\
&\quad \text{C5}: 0 \le s_{n,m} \le 1, m \in \mathcal{M}\cup\{0\}\\
&\quad \text{C6}: f_n^{l,\min} \le f_n^l \le f_n^{l,\max}\\
&\quad \text{C7}: T_n \le T_n^{\max}\\
&\quad \text{C8}: s_{n,0}D_n - f_n^l T_n \le 0\\
&\quad \text{C9}: \frac{s_{n,m}C_n}{r_{n,m}} + \frac{\sum_{m\in\mathcal{M}} s_{n,m}D_n}{f_n^c} \le T_n \ \forall m \in \mathcal{M}.
\end{aligned} \tag{12}$$

Remark: Given that the IoT task can be computed both locally and remotely at the same time, the task completion latency t_n^{total} depends on the larger of the local and remote task latency, which cannot exceed the maximum allowable latency $T_n^{\max}$. We give the connections between them by (9), i.e., $t_n^{\text{total}} = \max\{t_n^{\text{local}}, t_n^{\text{off}}\} \le T_n^{\max}$. In the transformed problem (12), we replace t_n^{total} with T_n in the optimization objective and constraint C1 (C1 was changed to C7 after this operation), and we add two complementary constraints C8 and C9 based on $t_n^{\text{total}} \le T_n$. The transformed problem (12) is intuitively equivalent to the originally formulated problem (11), in which the optimal solutions would be achieved at the boundaries of C8 or C9, or both of them. In other words, the optimal T_n^* obtained from problem (12) satisfies $T_n^* = t_n^{\text{total}}$. This is because if not, we can immediately reduce T_n^* to further degrade the optimization objective. Hence, we must have $T_n^* = t_n^{\text{total}}$.

However, the converted problem (12) is still nonconvex and hard to solve. To solve this problem, we introduce another set of variables $\mathbf{a} = \{a_{n,1}, \ldots, a_{n,M}\}$, which are defined as

$$a_{n,m} = \frac{1}{B_{n,m}\log\left(1+p_{n,m}\Gamma_{n,m}\right)} \ \forall m \in \mathcal{M}. \tag{13}$$

Since the feasible region of the transmit power is $0 \le p_{n,m} \le p_{n,m}^{\max}$ and we can achieve that $a_{n,m} \ge (1/[B_{n,m}\log(1+p_{n,m}^{\max}\Gamma_{n,m})])$. By substituting $\mathbf{a}$ into problem (12), it can be further expressed as

$$\begin{aligned}
&\min_{\{T_n,f_n^l,\mathbf{s},\mathbf{a}\}} Z'\left(T_n,f_n^l,\mathbf{s},\mathbf{a}\right)\\
&\text{s.t. C3}: s_{n,0} + \sum_{m\in\mathcal{M}} s_{n,m} = 1\\
&\quad \text{C5}: 0 \le s_{n,m} \le 1, m \in \mathcal{M}\cup\{0\}\\
&\quad \text{C6}: f_n^{l,\min} \le f_n^l \le f_n^{l,\max}\\
&\quad \text{C7}: T_n \le T_n^{\max}\\
&\quad \text{C10}: E\left(f_n^l,\mathbf{s},\mathbf{a}\right) \le E_n^R\\
&\quad \text{C11}: a_{n,m}s_{n,m}C_n + \frac{\sum_{m\in\mathcal{M}} s_{n,m}D_n}{f_n^c} \le T_n \ \forall m \in \mathcal{M}\\
&\quad \text{C12}: a_{n,m} \ge \frac{1}{B_{n,m}\log\left(1+p_{n,m}^{\max}\Gamma_{n,m}\right)} \ \forall m \in \mathcal{M}
\end{aligned} \tag{14}$$

where

$$\begin{aligned}
Z'\left(T_n,f_n^l,\mathbf{s},\mathbf{a}\right) &= w_n' T_n + \left(1-w_n'\right)\alpha E\left(f_n^l,\mathbf{s},\mathbf{a}\right)\\
E\left(f_n^l,\mathbf{s},\mathbf{a}\right) &= \kappa\left(f_n^l\right)^2 s_{n,0}D_n\\
&\quad + \sum_{m\in\mathcal{M}} a_{n,m}\left(2^{\frac{1}{a_{n,m}B_{n,m}}}-1\right)\frac{s_{n,m}C_n}{\Gamma_{n,m}}.
\end{aligned} \tag{15}$$

Even though the transformed problem (14) is still nonconvex, we observe that if we fix f_n^l and $\mathbf{a}$, problem (14) can be transformed to be a linear programming problem with respect to T_n and $\mathbf{s}$. Likewise, if we fix T_n and $\mathbf{s}$, it can be transformed to be a convex programming problem with respect to f_n^l and $\mathbf{a}$. Therefore, problem (14) can be classified into a biconvex minimization problem [42]. To solve this problem, we propose an effective method to solve the biconvex problems aided by the ACS theory. The main idea is that we alternately optimize the variable blocks when other variable blocks are fixed. The algorithm will not stop until convergence.

B. ACS-Based Algorithm

1) Update T_n and $\mathbf{s}$: Let τ denote the τth iteration. $f_n^{l,\tau-1}$ and $\mathbf{a}^{\tau-1}$ are determined in the $\tau-1$th iteration. Substituting them into problem (14), we can obtain a linear programming problem with respect to T_n and $\mathbf{s}$, which is given by

$$\begin{aligned}
&\min_{\{T_n,\mathbf{s}\}} Z'\left(T_n, f_n^{l,\tau-1}, \mathbf{s}, \mathbf{a}^{\tau-1}\right)\\
&\text{s.t. C3}: s_{n,0} + \sum_{m\in\mathcal{M}} s_{n,m} = 1\\
&\quad \text{C5}: 0 \le s_{n,m} \le 1, m \in \mathcal{M}\cup\{0\}\\
&\quad \text{C7}: T_n \le T_n^{\max}\\
&\quad \text{C8}: s_{n,0}D_n - f_n^{l,\tau-1}T_n \le 0\\
&\quad \text{C10}: E\left(f_n^{l,\tau-1}, \mathbf{s}, \mathbf{a}^{\tau-1}\right) \le E_n^R\\
&\quad \text{C11}: a_{n,m}^{\tau-1}s_{n,m}C_n + \frac{\sum_{m\in\mathcal{M}} s_{n,m}D_n}{f_n^c} \le T_n \ \forall m \in \mathcal{M}.
\end{aligned} \tag{16}$$

This problem (16) is a classic optimization problem which can be solved by many efficient methods, e.g., the simplex method [43], we will not expand the details in this article.

2) Update f_n^l and $\boldsymbol{a}$: We can obtain T_n^τ and $\mathbf{s}^\tau$ by solving the linear programming problem (16). Substituting them into problem (14), we can obtain a convex problem with respect to f_n^l and $\mathbf{a}$, which can be rearranged as

$$\begin{aligned}
&\min_{\{f_n^l,\mathbf{a}\}} E\left(f_n^l, \mathbf{s}^\tau, \mathbf{a}\right)\\
&\text{s.t. C13}: \tilde{f}_n^{l,\min} \le f_n^l \le f_n^{l,\max},\\
&\quad \text{C14}: a_{n,m}^{\min} \le a_{n,m} \le a_{n,m}^{\max} \ \forall m \in \mathcal{M}
\end{aligned} \tag{17}$$

where

$$\begin{aligned}
\tilde{f}_n^{l,\min} &= \max\left\{\frac{s_{n,0}^\tau D_n}{T_n^\tau}, f_n^{l,\min}\right\}\\
a_{n,m}^{\min} &= \frac{1}{B_{n,m}\log\left(1+p_{n,m}^{\max}\Gamma_{n,m}\right)}\\
a_{n,m}^{\max} &= \frac{T_n^\tau}{s_{n,m}^\tau C_n} - \frac{\sum_{m\in\mathcal{M}} s_{n,m}^\tau D_n}{f_n^c s_{n,m}^\tau C_n}.
\end{aligned} \tag{18}$$

Note that we omit the energy constraint C10 in problem (17), this can be explained by the following lemma Lemma 1.

Lemma 1: Problem (17) is equivalent to problem (14) if $T_n = T_n^\tau$ and $\mathbf{s} = \mathbf{s}^\tau$ are the optimal solutions to problem (16).

Proof: When $T_n = T_n^\tau$ and $\mathbf{s} = \mathbf{s}^\tau$ are given, the objective function in problem (14) can be simplified as $E(f_n^l, \mathbf{s}^\tau, \mathbf{a})$ and constraints C3, C5, and C7 can be omitted directly. We then combine constraints C6 and C8 into C13, and C11 and C12 into C14. Since $T_n = T_n^\tau$ and $\mathbf{s} = \mathbf{s}^\tau$ are the optimal solutions to problem (16), we have $E(f_n^{l,\tau-1}, \mathbf{s}^\tau, \mathbf{a}^{\tau-1}) \le E_n^R$. Again, the objective of problem (17) is a lower bound of $E(f_n^{l,\tau-1}, \mathbf{s}^\tau, \mathbf{a}^{\tau-1})$. Therefore, we always have $E(f_n^l, \mathbf{s}^\tau, \mathbf{a}) \le E(f_n^{l,\tau-1}, \mathbf{s}^\tau, \mathbf{a}^{\tau-1}) \le E_n^R$. Hence, constraint C10 is always satisfied in problem (17) and can be omitted. The proof is finished.

It can be seen that problem (17) is separable and can be decoupled into $M+1$ 1-D subproblems. One of them is with respect to f_n^l and given by

$$\begin{aligned}
&\min_{\{f_n^l\}} \kappa\left(f_n^l\right)^2 s_{n,0}^\tau D_n\\
&\text{s.t. C13}: \tilde{f}_n^{l,\min} \le f_n^l \le f_n^{l,\max}.
\end{aligned} \tag{19}$$

The objective of problem (19) is monotonously increasing in the feasible region. Therefore, the optimal solution $f_n^{l,\tau}$ can be determined by

$$f_n^{l,\tau} = \tilde{f}_n^{l,\min}. \tag{20}$$

The other M subproblems are with respect to $\mathbf{a}$, one of which is given by

$$\begin{aligned}
&\min_{\{a_{n,m}\}} a_{n,m}\left(2^{\frac{1}{a_{n,m}B_{n,m}}} - 1\right)\frac{s_{n,m}^\tau C_n}{\Gamma_{n,m}}\\
&\text{s.t. C14}: a_{n,m}^{\min} \le a_{n,m} \le a_{n,m}^{\max}.
\end{aligned} \tag{21}$$

Before solving this subproblem, we first give the following lemma Lemma 2.

Lemma 2: The objective of problem (21) is monotonously decreasing with $a_{n,m}$.

Proof: Let $E_{n,m}^\tau$ represent the objective of problem (21) as

$$E_{n,m}^\tau = a_{n,m}\left(2^{\frac{1}{a_{n,m}B_{n,m}}} - 1\right)H_{n,m}^\tau \tag{22}$$

where $H_{n,m}^\tau = (s_{n,m}^\tau C_n/\Gamma_{n,m})$.

The first-order derivative and the second-order derivative of $E_{n,m}^\tau$ in terms of $a_{n,m}$ can be, respectively, written as

$$\frac{\partial E_{n,m}^\tau}{\partial a_{n,m}} = H_{n,m}^\tau\left[2^{\frac{1}{a_{n,m}B_{n,m}}}\left(1 - \frac{\ln 2}{a_{n,m}B_{n,m}}\right) - 1\right] \tag{23}$$

$$\frac{\partial^2 E_{n,m}^\tau}{\partial a_{n,m}^2} = \frac{2^{\frac{1}{a_{n,m}B_{n,m}}}(\ln 2)^2 H_{n,m}^\tau}{a_{n,m}^3 B_{n,m}^2}. \tag{24}$$

It is obvious that the second-order derivative is always positive and thereby the first-order derivative is a monotonously increasing function with $a_{n,m}$. Moreover, the first-order derivative satisfies $(\partial E_{n,m}^\tau/\partial a_{n,m}) < 0$ as $a_{n,m}$ is small and $\lim_{a_{n,m}\to+\infty}(\partial E_{n,m}^\tau/\partial a_{n,m}) = 0$ as $a_{n,m} \to +\infty$. As a result, the first-order derivative always satisfies $(\partial E_{n,m}^\tau/\partial a_{n,m}) \le 0$. Therefore, the function $E_{n,m}^\tau$ is monotonously decreasing with $a_{n,m}$. The proof is finished.

With Lemma 2, the optimal solution of problem (21) is always obtained at its right boundary, i.e.,

$$a_{n,m}^\tau = a_{n,m}^{\max}. \tag{25}$$

In summary, the details of the proposed ACS-based algorithm is given by Algorithm 1, and the convergence of the proposed algorithm is proofed by Theorem 1.

Theorem 1: The ACS-based algorithm can always converge to a suboptimal solution to the original TECLC problem.

Proof: From each iteration in the ACS-based algorithm, we can get a smaller or equal objective value comparing to the last iteration, namely, $Z'(T_n^\tau, f_n^{l,\tau}, \mathbf{a}^\tau, \mathbf{s}^\tau) \le Z'(T_n^{\tau-1}, f_n^{l,\tau-1}, \mathbf{a}^{\tau-1}, \mathbf{s}^{\tau-1})$. Therefore, as the algorithm continues, it produces a nonincreasing objective sequence. Since $Z'(T_n, f_n^l, \mathbf{a}, \mathbf{s})$ is bounded below by a nonnegative value,

Algorithm 1 ACS-Based Algorithm

Input: Network parameters $\Gamma_{n,m}$, r_n^E, α, etc; convergence tolerance ϵ; iteration index $\tau = 1$.
Output:$f_n^{l,*}$, $\mathbf{p}^*$ and $\mathbf{s}^*$;
Initialize starting variables $f_n^{l,0}$, T_n^0, $\mathbf{a}^0$ and $\mathbf{s}^0$;
repeat
  Update T_n^τ and $\mathbf{s}^\tau$ by solving the linear programming problem (16);
  Update $f_n^{l,\tau}$ according to equation (20);
  Update $\mathbf{a}^\tau$ according to equation (25);
until $|Z'(T_n^\tau, f_n^{l,\tau}, \mathbf{a}^\tau, \mathbf{s}^\tau) - Z'(T_n^{\tau-1}, f_n^{l,\tau-1}, \mathbf{a}^{\tau-1}, \mathbf{s}^{\tau-1})| \leq \epsilon$
Obtain $\mathbf{p}^*$ according to (13);
return $f_n^{l,*}$, $\mathbf{p}^*$ and $\mathbf{s}^*$;

there must exist a large enough T, such that for $\tau \geq T$, $|Z'(T_n^\tau, f_n^{l,\tau}, \mathbf{a}^\tau, \mathbf{s}^\tau) - Z'(T_n^{\tau-1}, f_n^{l,\tau-1}, \mathbf{a}^{\tau-1}, \mathbf{s}^{\tau-1})| \leq \epsilon$ always holds, where ϵ is a small positive number. Thus, it can always converge to a suboptimal solution to problem (14). Since problem (14) is equivalent to the original TECLC problem, the proposed ACS-based algorithm can also converge to a suboptimal solution to the TECLC problem. The proof is finished.

C. Special Case: Local Execution

With the increasing power on smart devices, it is also an efficient way to make on-device data analysis locally with AI applications for achieving low-latency, reduced resource consumption, and privacy protection in the future. Hence, we also analyze one special case (i.e., local execution) to compare with the partial offloading algorithm in this article.

Local Execution: In this case, the UE task A_n is totally executed locally by the UE itself. It means that $s_{n,0} = 1$ and $s_{n,m} = 0, p_{n,m} = 0\ \forall m \in \mathcal{M}$. The TECLC problem can be simplified as

$$\min_{\{f_n^l\}} \frac{w_n' D_n}{f_n^l} + \left(1 - w_n'\right)\alpha\kappa\left(f_n^l\right)^2 D_n$$
$$\text{s.t. C15} : \tilde{\tilde{f}}_n^{l,\min} \leq f_n^l \leq \tilde{f}_n^{l,\max} \tag{26}$$

where $\tilde{\tilde{f}}_n^{l,\min} = \max\{(D_n/T_n^R), f_n^{l,\min}\}$ and $\tilde{f}_n^{l,\max} = \min\{\sqrt{(E_n^R/\kappa D_n)}, f_n^{l,\max}\}$. Note that the objective is differentiable and convex with respect to f_n^l. Hence, the optimal solution is obtained at either of its stationary point or the bounds of the feasible region. By the first-order condition, we can get the stationary point $f_n^{l,s} = \sqrt[3]{([w_n' D_n]/[2(1 - w_n')\alpha\kappa D_n])}$. The optimal solution $f_n^{l,*}$ can be determined by the following equation:

$$f_n^{l,*} = \max\left\{\tilde{\tilde{f}}_n^{l,\min}, \min\left\{\tilde{f}_n^{l,\max}, f_n^{l,s}\right\}\right\}. \tag{27}$$

VI. Simulation Results

In this section, we provide simulation results to evaluate the performance of the proposed partial offloading policy by comparing to the local execution policy. We assume that there are three RATs with the average distance d m to the UE, and we assume that the task arrival $\lambda_n(t) \sim$ Uniform [0, 5] Mb/s. The channel gain models of the three RATs are adopted as the same as [23]. The detailed simulation parameters are given in Table I.[2] In particular, the UE is equipped with the multi-RAT connectivity capability which enables the UE to further split the offloading part of the task to several subchunks and offload them by these RATs independently. It is noted that different RATs generally operate over different spectrum bands (e.g., 2.4 GHz for WiFi and 1.8–2.3 GHz for the cellular network). Hence, there does not exist cross-RAT interference in this system [38]. The perfect uplink channel state information to all RATs is available for the UE. For the partial offloading, we also give the performance comparison between the schemes with different number of RATs.

TABLE I
KEY PERFORMANCE INDICATORS

Parameters	Values
f_n^l	$[0.2, 1]$GHz
f_n^c	2GHz
A_n	1 Megabits, 0.5 Gigacycles, 3 Sec
$\lambda_n(t)$	Uniform $[0, 5]$ Mbps
$B_{n,1}$	30MHz
$B_{n,2}$	20MHz
$B_{n,3}$	10MHz
$p_{n,1}^{max}$	0.15W
Shadowing standard deviation	4dB
Pathloss	$128.1 + 37.6log10(max(d_{Km}))$
Noise Power	-174 dBm/Hz
$p_{n,2}^{max}$	0.1W
$p_{n,3}^{max}$	0.05W
E_n^{max}	5×10^5 Joule
E_n^R	5×10^5 Joule
w_n	0.8
α	1
Number of RATs	3

Fig. 3 shows the energy consumption performance in terms of residual energy with the weighting factor $w_n = 0.8$. It shows that the energy consumption of location execution policy significantly increases with the increasing residual energy. The proposed partial offloading algorithm needs much less energy consumption with increasing residual energy. In addition, energy consumption reduces with the increasing number of RAT wireless networks. Because the proposed algorithm splits the tasks properly and offloads them to different RAT networks considering the different RAT network environments.

Fig. 4 shows the latency performance in terms of residual energy with the weighting factor $w_n = 0.8$. It shows that service latency decreases with the increase of residual energy. The proposed partial offloading algorithm can efficiently reduce the latency compared with the local execution. Furthermore, we can also see that the latency of the proposed algorithm decreases gradually with the increasing number of RATs by taking the benefits of multi-RAT networks. In particular, we can see that there exists the tradeoff performance between the energy consumption and latency cost considering the impact of

[2]We aim to verify with theoretical analysis with numerical simulation based on the current wireless access technologies and reasonable assumptions from a theoretical view, which is not strictly consistent with the real systems due to the high complexity.

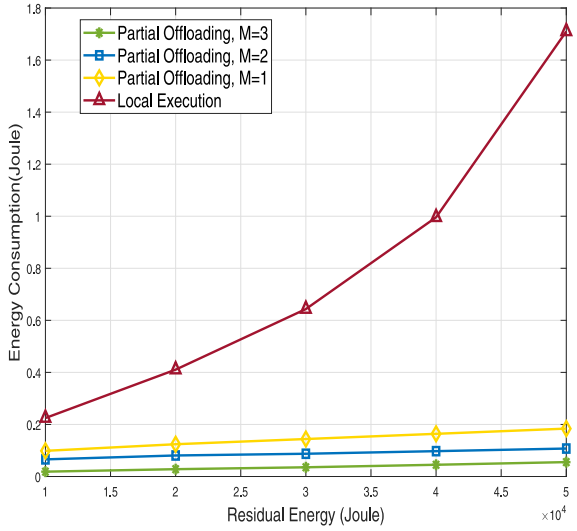


Fig. 3. Energy consumption versus residual energy.

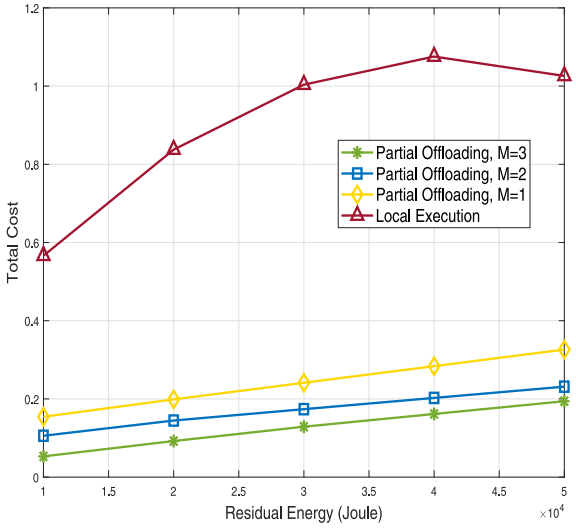


Fig. 5. Total cost versus residual energy.

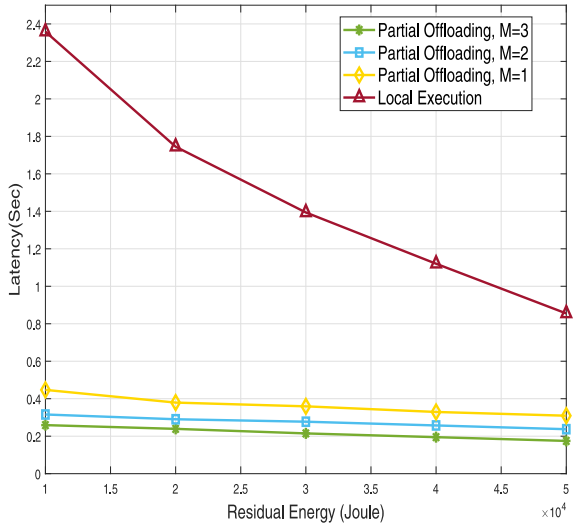


Fig. 4. Latency cost versus residual energy.

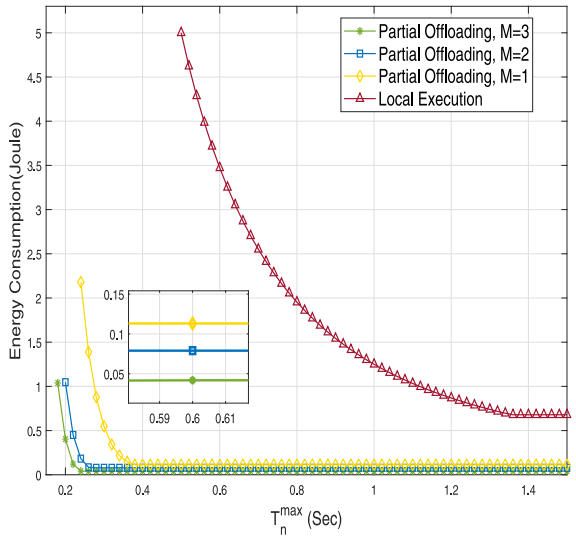


Fig. 6. Energy consumption versus latency requirement.

the residual energy in Figs. 3 and 4. Because that more residual energy will allow the UE to pursue a lower latency with higher consumption of energy. In theory, more residual energy leads to a larger weight for latency and a smaller weight for energy consumption, respectively, which prompts the system to minimize a smaller latency than energy consumption. The proposed partial algorithm is practical which provides a guideline for different network requirements.

Fig. 5 shows the total cost of the network in terms of the increase of residual energy. It shows that the total cost decreases with the increasing number of RATs. In addition, it also shows that the proposed partial offloading algorithm can reduce the total cost effectively compared with the local execution policy. Because the UE can make full use of the high-quality channels and then optimize the task splitting by taking the larger diversity gain of different RATs in wireless networks.

Fig. 6 shows the energy consumption performance in terms of the increasing latency requirement $T_n^{\max}$. It shows that energy consumption decreases in the first stage and then stays the same as $T_n^{\max}$ increases. We can see that the network consumes much less energy with the increasing number of RATs and the proposed algorithm performances much better compared with local execution. This is because when $T_n^{\max}$ is small, the system needs to consume more energy to meet the latency requirement. However, as $T_n^{\max}$ increases, the latency requirement constraint becomes ineffective, which leads to the unchanged curves.

Fig. 7 shows the latency cost in terms of the increasing latency requirement $T_n^{\max}$. It shows that the latency cost first increases and then remains unchanged as $T_n^{\max}$ increases. The latency cost of the proposed algorithm decreases with the increasing number of RATs. From Figs. 6 and 7, the trade-off performance between the energy consumption and latency cost under different latency requirements can be achieved. Furthermore, the lowest latency that can be supported for partial offloading is 0.18 s under $M = 3$, 0.20 s under $M = 2$, 0.24 s under $M = 1$, and that for local execution is 0.5 s,

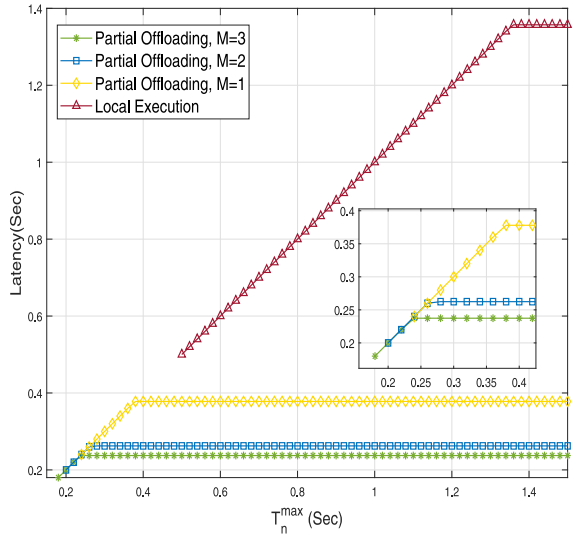


Fig. 7. Latency cost versus latency requirement.

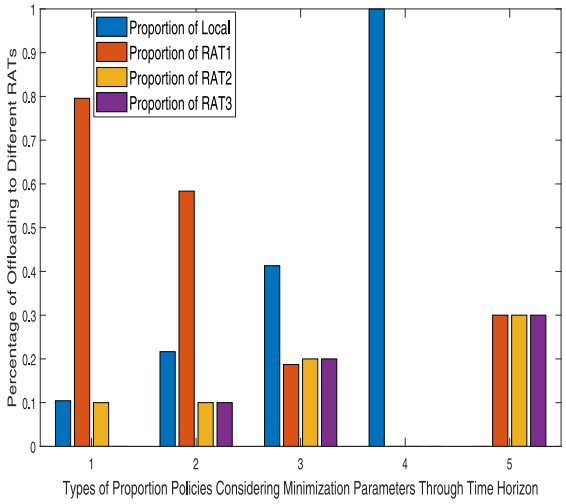


Fig. 9. Percentage of offloading to different RATs for L2SC task offloading with minimization parameters.

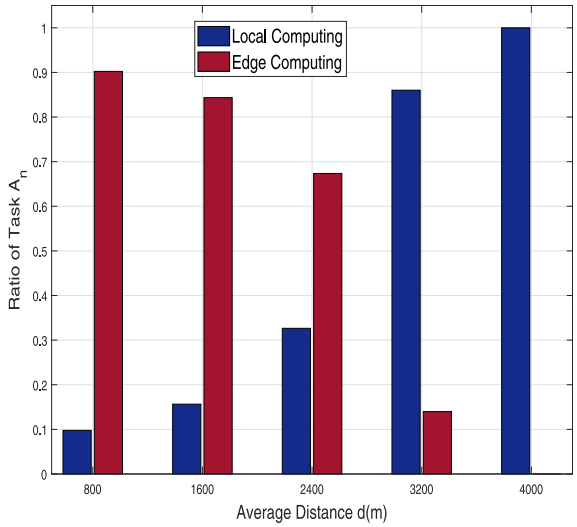


Fig. 8. Impact of average distance on task splitting, under $M = 3$.

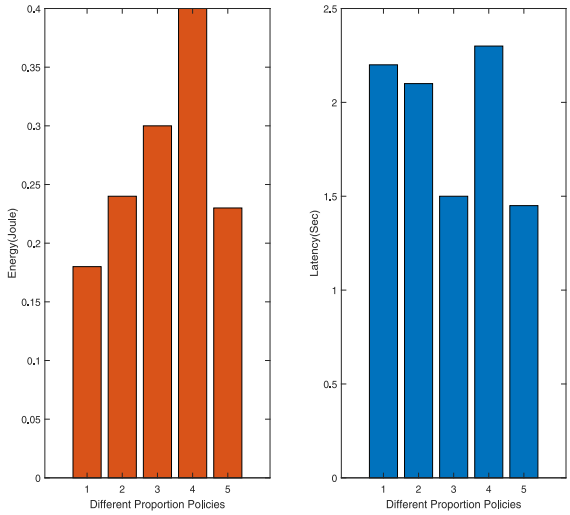


Fig. 10. Latency and energy versus the percentage of offloading to different RATs.

respectively. This demonstrates that the partial offloading with multi-RATs is more applicable to the latency-sensitive tasks.

Fig. 8 shows the task splitting performance in terms of the increasing average distance d between different RATs and the UE. It shows that when d is relatively small, a large proportion of the task is offloaded to the MEC server for task computing. Because as d is small, the UE can achieve good channel quality such that the uploading latency and the energy consumption can be reduced drastically. However, as d increases, the channel quality becomes worse so that most of the tasks prefer to compute locally considering the latency and energy limitations. The proposed algorithm can split the tasks properly and transmit through different RATs to satisfy the UEs' requirements.

Fig. 9 shows the percentage of offloading to different RATs for L2SC tasks offloading considering minimization parameters in our proposed algorithm. Fig. 10 shows the latency and energy performance in terms of different task splitting proportions of RATs. To reduce the latency and save more energy, the UE will partition one part of its L2SC tasks to offload to the MEC server by different RATs. Furthermore, benefitting from the multi-RAT capability, the UE can offload through multiple RATs in parallel. In particular, the percentage of offloading to different RATs considering the minimization parameters is given in Fig. 9, which can indicate the relativity between the energy and delay parameters on the offloading portion to each of the RATs through time horizon in our proposed algorithm process. In addition, as shown in Fig. 10, we can see that the latency cost of the proposed algorithm decreases with the increasing number of RATs. Because the proposed algorithm splits the tasks properly and offloads them to different RAT networks considering the different RAT network environments. However, with the increasing number of RATs, the cost both in energy and computation of UEs will increase gradually that will decrease the performance of the partial offloading, which exits a tradeoff between the increasing number of multi-RAT and the increasing performance of the partial offloading policy.

VII. Conclusion

In this article, we studied the energy-efficient IoT task offloading problem for L2SC services with huge task size and low-latency requirements, which is suitable with bit independence in MEC-enhanced multi-RAT wireless networks. We investigated the energy-latency tradeoff performance for partial task computation offloading. We formulated the problem as the minimization of the weighted sum of the latency cost and the energy consumption which was shown to be nonsmooth and nonconvex. In order to tackle this problem in an efficient way, we first converted it to a smooth biconvex problem by variable substitution, and then proposed an ACS-based algorithm by jointly optimizing the local computing frequency, task splitting, and transmit power. Numerical simulation results demonstrate that the proposed partial offloading algorithm can achieve good performance for the IoT networks, and it is quite practical which can provide an efficient guideline for different IoT devices' requirements in future 5G networks.

References

[1] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

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

[3] L. Lyu, C. Chen, S. Zhu, and X. Guan, "5G enabled codesign of energy-efficient transmission and estimation for industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2690–2704, Jun. 2018.

[4] H. Peng, Q. Ye, and X. S. Shen, "SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 156–162, Aug. 2019.

[5] L. Tang and S. He, "Multi-user computation offloading in mobile edge computing: A behavioral perspective," *IEEE Netw.*, vol. 32, no. 1, pp. 48–53, Jan./Feb. 2018.

[6] 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./Feb. 2018.

[7] S. Sukhmani, M. Sadeghi, M. Erol-Kantarci, and A. El Saddik, "Edge caching and computing in 5G for mobile AR/VR and tactile Internet," *IEEE MultiMedia*, vol. 26, no. 1, pp. 21–30, Jan.–Mar. 2019.

[8] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, and Y. Hao, "Ready player one: UAV-clustering-based multi-task offloading for vehicular VR/AR gaming," *IEEE Netw.*, vol. 33, no. 3, pp. 42–48, May/Jun. 2019.

[9] T. Dang and M. Peng, "Joint radio communication, caching, and computing design for mobile virtual reality delivery in fog radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1594–1607, Jul. 2019.

[10] Y. Gu, Q. Cui, Q. Ye, and W. Zhuang, "Game-theoretic optimization for machine-type communications under QoS guarantee," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 790–800, Feb. 2019.

[11] W. Wei, H. Gu, K. Wang, X. Yu, and X. Liu, "Improving cloud-based IoT services through virtual network embedding in elastic optical inter-DC networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 986–996, Feb. 2019.

[12] L. P. Qian, A. Feng, Y. Huang, Y. Wu, B. Ji, and Z. Shi, "Optimal SIC ordering and computation resource allocation in MEC-aware NOMA NB-IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2806–2816, Apr. 2019.

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

[14] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

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

[16] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[17] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[18] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.

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

[20] P. Gawlowicz, A. Zubow, and A. Wolisz, "Enabling cross-technology communication between LTE unlicensed and WiFi," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 144–152.

[21] L. Diez, A. Garcia-Saavedra, V. Valls, X. Li, X. Costa-Perez, and R. Aguero, "LaSR: A supple multi-connectivity scheduler for multi-RAT OFDMA systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 624–639, Mar. 2020.

[22] T. Shuminoski and T. Janevski, "Lyapunov optimization framework for 5G mobile nodes with multi-homing," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1026–1029, May 2016.

[23] W. Wu, Q. Yang, P. Gong, and K. S. Kwak, "Energy-efficient traffic splitting for time-varying multi-RAT wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6523–6535, Jul. 2017.

[24] S. Singh, M. Geraseminko, S. Yeh, N. Himayat, and S. Talwar, "Proportional fair traffic splitting and aggregation in heterogeneous wireless networks," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1010–1013, May 2016.

[25] J. Zhang *et al.*, "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.

[26] Q. Li, L. Zhao, J. Gao, H. Liang, L. Zhao, and X. Tang, "SMDP-based coordinated virtual machine allocations in cloud-fog computing systems," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1977–1988, Jun. 2018.

[27] Y. Wu, L. P. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 392–407, Jun. 2019.

[28] K. Ching-Ju Lin, H. Wang, Y. Lai, and Y. Lin, "Communication and computation offloading for multi-RAT mobile edge computing," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 180–186, Dec. 2019.

[29] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Trans. Cloud Comput.*, early access, Jun. 20, 2019, doi: 10.1109/TCC.2019.2923692.

[30] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. S. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," *IEEE Trans. Cloud Comput.*, early access, Mar. 05, 2019, doi: 10.1109/TCC.2019.2903240.

[31] H. Peng, Q. Ye, and X. Shen, "Spectrum management for multi-access edge computing in autonomous vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 3001–3012, Jul. 2020.

[32] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4642–4655, Oct. 2018.

[33] S. Ranadheera, S. Maghsudi, and E. Hossain, "Computation offloading and activation of mobile edge computing servers: A minority game," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 688–691, Oct. 2018.

[34] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[35] H. Sun, F. Zhou, and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3052–3056, Mar. 2019.

[36] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, Aug. 2019.

[37] U. Saleem, Y. Liu, S. Jangsher, and Y. Li, "Performance guaranteed partial offloading for mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.
[38] W. Wu, Q. Yang, P. Gong, and K. S. Kwak, "Energy-efficient resource optimization for OFDMA-based multi-homing heterogenous wireless networks," *IEEE Trans. Signal Process.*, vol. 64, no. 22, pp. 5901–5913, Nov. 2016.
[39] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.
[40] Y. Kao, B. Krishnamachari, M. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3056–3069, Nov. 2017.
[41] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.
[42] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: A survey and extensions," *Math. Methods Oper. Res.*, vol. 66, no. 3, pp. 373–407, 2007.
[43] V. Chvátal, *Linear Programming*. New York City, NY, USA: Macmillan, 1983.

Meng Qin (Member, IEEE) received the B.S. degree in communication engineering from Taiyuan University of Technology, Taiyuan, China, in 2012, and the M.S. and Ph.D. degrees in information and communication systems from Xidian University, Xi'an, China, in 2015 and 2018, respectively.

He is currently a joint Postdoctoral Fellow with Peng Cheng Laboratory, Shenzhen, China, and Peking University, Beijing, China. His research interests include green cloud storage AI-aided self-organized wireless networks, and edge intelligence in wireless networks.

Nan Cheng (Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2016.

He worked as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, from 2017 to 2019. He is currently a Professor with the State Key Laboratory of ISN and the School of Telecommunication Engineering, Xidian University, Xi'an, China. His current research focuses on B5G/6G, space-air-ground integrated network, big data in vehicular networks, and self-driving system. His research interests also include performance analysis, MAC, opportunistic communication, and application of AI for vehicular networks.

Zewei Jing received the B.S. degree from Inner Mongolia University, Hohhot, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Telecommunication Engineering, Xidian University, Xi'an, China.

His research interests include stochastic network optimization, and radio resource allocation and their applications in mobile edge computing networks.

Tingting Yang (Member, IEEE) received the B.Sc. and Ph.D. degrees from Dalian Maritime University, Dalian, China, in 2004 and 2010, respectively.

She is currently a Professor with the School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan, China. From September 2012 to August 2013, she was a Visiting Scholar with the Broadband Communications Research Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her research interests are in the areas of maritime wideband communication networks, DTN networks, and green wireless communications.

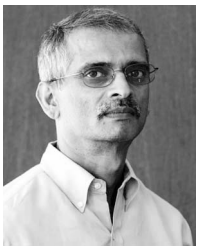
Prof. Yang serves as the Associate Editor-in-Chief of *IET Communications*, as well as the Advisory Editor for SpringerPlus. She also served as a TPC Member for IEEE ICC'14 and ICC'15 Conference.

Wenchao Xu (Member, IEEE) received the B.E. and M.E. degrees from Zhejiang University, Hangzhou, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2018.

He is currently a Research Associate with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. In 2011, he joined Alcatel Lucent Shanghai Bell Company Ltd., Shanghai, China, where he was a Software Engineer of telecom virtualization. His interests include wireless communications with emphasis on resource allocation, network modeling, and AI applications.

Qinghai Yang (Member, IEEE) received the B.S. degree in communication engineering from Shandong University of Technology, Zibo, China, in 1998, the M.S. degree in information and communication systems from Xidian University, Xi'an, China, in 2001, and the Ph.D. degree in communication engineering from Inha University, Incheon, South Korea, in 2007 with university-president award.

From 2007 to 2008, he was a Research Fellow with UWB-ITRC, Incheon. Since 2008, he has been with Xidian University. His current research interests include the fields of autonomic systems, embedded computing for real time machine learning, and networking for AI.



Ramesh R. Rao (Fellow, IEEE) received the bachelor's and M.S. degrees from the University of Madras (the National Institute of Technology), Tiruchirapalli, India, in 1980 and 1982, respectively, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 1984.

He has been a Faculty Member with the University of California at San Diego (UCSD), La Jolla, CA, USA, since 1984, and the Director of the Qualcomm Institute, UCSD Division of the California Institute for Telecommunications and Information Technology, La Jolla, since 2001. He served as the Director of UCSD's Center for Wireless Communications, La Jolla. He holds the Qualcomm Endowed Chair of telecommunications and information technologies with the Jacobs School of Engineering, UCSD, where he is a Member of the School Electrical and Computer Engineering Department.

Dr. Rao is a Senior Fellow of the California Council on Science and Technology.