Joint Task Offloading and Resource Allocation for IoT Edge Computing with Sequential Task Dependency

Xuming An, Rongfei Fan, *Member, IEEE*, Han Hu, *Member, IEEE*, Ning Zhang, *Senior Member, IEEE*, Saman Atapattu, *Senior Member, IEEE*, and Theodoros A. Tsiftsis *Senior Member, IEEE*

Abstract—Incorporating mobile edge computing (MEC) in Internet of Things (IoT) enables resource-limited IoT devices to offload their computation tasks to a nearby edge server. In this paper, we investigate an IoT system assisted by the MEC technique with its computation task subjected to sequential task dependency, which is critical for video stream processing and other intelligent applications. To minimize energy consumption per IoT device while limiting task processing delay, task offloading strategy, communication resource, and computation resource are optimized jointly under both slow and fast fading channels. In slow fading channels, an optimization problem is formulated, which is non-convex and involves one integer variable. To solve this challenging problem, we decompose it as a one-dimensional search of task offloading decision problem and a non-convex optimization problem with task offloading decision given. Through mathematical manipulations, the non-convex problem is transformed to be a convex one, which is shown to be solvable only with the simple Golden search method. In fast fading channels, optimal online policies depending on instant channel state are derived even though they are entangled. In addition, it is proved that the derived policy will converge to the offline policy when channel coherence time is low, which can help to save extra computation complexity. Numerical results verify the correctness of our analysis and the effectiveness of our proposed strategies over existing methods.

Index Terms—Internet of Things (IoT), mobile edge computing (MEC), sequential task dependency, task offloading, resource allocation.

I. Introduction

The last decade has witnessed the fast-growing of the Internet of Things (IoT). This growth is even more expedited since the IoT meets artificial intelligence, through which patterns that matter can be learned from the massive data collected by IoT devices, and interested objects or anomalies can be spotted from instant data monitored by an IoT device [1]–[3]. On the other hand, the task of performing intelligent applications to detect interested objects or anomalies is usually of heavy burden, which is hard to be fully undertaken by any single IoT device, considering its limited computational capability, and the restriction on energy consumption. Offloading the heavy computation task to the cloud center may be a solution for the IoT devices. However, this will lead to a long time delay, which is especially undesirable when the running application is time-critical. To overcome the above challenges, mobile edge computing (MEC) can offer a promising solution, which implements an edge server in the vicinity of the IoT devices and permits the IoT devices to offload their computation tasks to the edge server [4], [5].

Regarding offloading the computation task [1] from one IoT device to the edge server, binary and partial offloading are mainly investigated. For binary offloading, the computation task is not separable. Hence it is completed locally at the IoT device or offloaded to and then computed at the edge server. For partial offloading, it is assumed that the computation task can be separated into two sub-tasks by any fraction, with one sub-task completed at the IoT device and the other one offloaded to and then computed in parallel at the edge server [4]. Within the framework of either binary or partial offloading, the IoT device's energy consumption and task completion delay are the two most significant cost metrics in literature, which can be highly reduced by optimizing the task offloading decision, the allocation of communication and computation resources [6]–[12]. Specifically, task offloading decision determines how much data is left for local computations and offloading to the edge server, respectively [6]; the communication resource involves not only IoT device's transmit power for task offloading [8], [9], [13], but also the bandwidth [10] or time slot duration [11], [12] for every IoT device when multiple devices are accessing the edge server; the computation resource primarily indicates edge server or IoT device's CPU frequency [14] (which represents the computation capability) over time. The general research goal is to

---

[1] When the task offloading is mentioned in this paper, it actually means to offload the input data of a computation task (which would be explained in Section III-A). To avoid misleading, this course is uniformed to be *task offloading* throughout this paper.

minimize energy consumption per IoT device while promising the computation task completed within a tolerable delay, by jointly optimizing task offloading decision, the allocation of communication and computation resources.

On the other hand, many intelligent applications, such as the deep neural network (DNN) driven applications including digital forensics services [15], target recognition or anomaly detection [16], involve the type of computation task with multiple indivisible sub-tasks conforming to sequential dependency. To be specific, the whole computation task is comprised of multiple sub-tasks and every sub-task is executable only after completing its precedent sub-task, which is regarded as a *sequential task* in the following. Since a sequential task cannot be decomposed into two sub-tasks running in parallel, the strategies based on partial offloading in literature do not work anymore. Furthermore, although the strategies based on binary offloading in literature can help to generate a feasible solution for task offloading decision if we see the sequential task as a whole, energy consumption and time completion delay is not fully reduced since it can be further cut down when we choose to offload from some proper sub-task. Hence there exist model mismatch and performance degradation if the developed strategies based on partial offloading or binary offloading are utilized for our focused sequential task.

In terms of task offloading for sequential tasks, given that the edge server has more powerful computation capability over the IoT device, it would be sensible to offload its computation task to the edge server as early as possible, i.e., to offload the sub-task at an early stage. However, due to the randomness of channel gain and the inhomogeneous input data amount associated with every sub-task, a long delay would be lead to in case of large input data amount and poor channel condition, no matter the offloaded sub-task is at an early stage or not. Hence it is not always good to offload the sub-task at an early stage, and it is interesting to study the best occasion to offload, which corresponds to a task offloading decision problem.

Although there is some MEC literature working on the task offloading decision problem for the sequential task [17], [18], which would be surveyed in Section II, they do not explore other ways to reduce IoT device's energy consumption and task completion delay. It would be beneficial to further perform joint allocation of communication and computation resources since both of them can affect the above two types of cost metrics. However, due to the entanglement of multiple types of variables which will lead to the non-convexity of the investigated problem and unpredictable random fluctuation of channel state, it is challenging to solve the focused problem directly. Additionally, this challenging problem cannot be answered well by the developed strategies based on partial or binary offloading due to the aforementioned model mismatch and performance degradation.

In this work, we aim to solve the task offloading decision problem together with the allocation of communication and computation resources for the sequential tasks. Specifically, both slow and fast fading channels are considered. Under these two considered channel models, task offloading strategy, communication resource, and/or computation resource are jointly optimized so as to minimize the energy consumption

of the IoT device, while guaranteeing the task computation latency. The main contributions are summarized as follows.

- **System model:** We propose a framework of joint task offloading, communication and computation resource allocation for a special type of computation task in a MEC system, i.e., sequential task, which is particularly useful for describing intelligent applications such as DNN driven ones. Fast fading and slow fading channels are investigated and associated optimization problems are formulated, which turn out to be a non-convex optimization problem involving one integer variable and a stochastic optimization problem with multiple online policies entangled in one constraint, respectively. Both of these two formulated problems are challenging to solve.

- **Joint task offloading and resource allocation under slow fading channel:** Global optimal solution of the non-convex optimization problem involving one integer variable is achieved through our two innovative efforts: 1) Divide the associated problem into two levels, with the lower level working on the non-convex part for any given integer variable and with the upper level dealing with the unique integer variable through one-dimensional exhaustive search; 2) Transform the non-convex problem in the lower level to be a convex one equivalently. What is even further, simple Golden search is shown to be enough to find the optimal solution of the transformed convex problem by investigating its Karush-Kuhn-Tucker (KKT) condition.

- **Joint task offloading and resource allocation under fast fading channel:** The entangled multiple online policies are decomposed and the optimal online policies are found for the associated optimization problem through our two mathematical operations: 1) Divide the associated optimization problem into two levels, with the online offloading policy optimized freely from constraint in the upper level and the rest online policies optimized in the lower level for any given online offloading policy. 2) Explore some special properties of minimal expected energy consumption for task offloading to make the solving of the lower level problem possible. Moreover, the dependency of cost function on instant channel state is proved to fade away when the channel coherence time approaches zero, which makes the replacement of online policy with offline policy at the price of ignorable performance loss but with much less computation complexity happen.

- **Numerical results:** Numerical results show that: 1) Our proposed methods can reduce energy consumption of IoT devices over the existing methods under both slow fading and fast fading channels; 2) The time consumption of our proposed methods for slow fading and fast fading channels are fully acceptable in the real application; 3) The disclosed properties of the minimal expected energy consumption for task offloading and founded convergence between online policy and offline policy as the channel coherence time approaches to zero are correct.

The rest of the paper is organized as follows. In Section II,

related literature is surveyed. Section III presents the system model and formulates the research problems under slow fading channel and fast fading channel. In Section IV, the optimal solution for the formulated research problems under slow fading channel and fast fading channel are presented respectively, followed by the discussion when channel coherence time approaches zero. Numerical results are given in Section V and concluding remarks are presented in Section VI.

II. RELATED WORKS

In literature, plenty of works have studied binary offloading or partial offloading. Due to the limit of space, we only survey the most representative works here, which includes the reference papers [8], [11], [12], [19]–[25]. For binary offloading, [19] proposes to take the action with less energy consumption between the options of fully local computing and fully offloading for a single-user MEC system. [8] investigates partial offloading problem together with the communication and computation resource allocation for a single-user MEC system. [11] and [12] studies the joint resource allocation and partial offloading for a time division multiple access (TDMA) and orthogonal frequency division multiple access (OFDMA) MEC system, respectively. Since non-orthogonal multiple access (NOMA) can achieve higher spectrum efficiency over orthogonal multiple access (OMA), it is also taken into account for task offloading. With the setup of fully offloading, the benefit of NOMA for a MEC system is firstly analyzed and verified in [20] and the resource allocation problem in a NOMA-aided MEC system is investigated in [21]. For partial offloading, [22] studies task offloading and resource allocation jointly with decoding order fixed, while [23] further optimizes the decoding order. Besides partial offloading, binary offloading, together with resource allocation, is also investigated in a NOMA-MEC system [22]. In addition to the multiple access technique, the idea of energy-neutral is also brought into the MEC system, with every IoT device driven by the energy harvesting technique [24], [25]. For the energy harvesting MEC system, partial offloading is investigated with the IoT device supported by randomly arrived environmental energy [24], while both partial offloading and binary offloading are explored with the IoT device supported by wireless-power-transfer technique in [25].

Compared with the rich literature focusing on binary offloading or partial offloading, research works taking the sequential task model into account are very limited, [17], [18], [26]. In [17], task completion delay is minimized for a single IoT device. Not only the link rate but also the computation capability at the IoT device and edge server are supposed to be stable. Hence only task offloading strategy is needed to be investigated. The one-climb policy is disclosed to be optimal, in which the IoT device only offloads its computation task to the edge server once and only one segment of sub-tasks is computed at the edge server if ever. With such a framework, [17] figures out which segment of sub-tasks should be selected for the edge server to achieve the minimal task completion delay. In [18], a single IoT devices is considered and its expected energy consumption is minimized while limiting the expected

or outage probability of task completion delay. Three types of channel conditions are investigated, which are given as static channel, independently and identically distributed (i.i.d) block fading channel, and Markovian stochastic channel. Similar to [17], only task offloading decision is explored and one-climb policy is proven to be optimal for all the investigated cases. In [26], a MEC system with two IoT devices is considered under static channel conditions, both of which have a sequential task to compute. Differently, one intermediate sub-task of the second IoT device requires the last sub-task's output of the first IoT device. A weighted sum of these two IoT devices' energy consumption and task completion delay is aimed to be minimized by optimizing not only the task offloading decision but also the CPU frequency and transmit power for offloading of these two IoT devices. With the task offloading decision given, the optimal solution of all the other variables is found. In terms of the task offloading decision, the one-climb policy is also proven to be optimal.

This research also investigates the MEC system under the sequential task model adopted in [17] and [18] for both slow fading channels and fast fading channels. However, different from [17] and [18], both of which solely focus on task offloading policy, we investigate the joint optimization of task offloading decision, communication resource (including IoT device's transmit power in every fading block) allocation, and/or computation resource allocation (including the allocated CPU for completing every sub-task). Although [26] also investigates the joint design of task offloading decision and resource allocation, its working procedure involves inter-user task dependency, which is different from ours and the one in [17] and [18]. The research results published in [26] cannot address the questions raised in this paper.

III. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a MEC system with one edge server and multiple IoT devices. Every IoT device is allocated with one dedicated channel for task offloading and one dedicated CPU core at the edge server for computing. The bandwidth of the dedicated channel is W and the dedicated CPU core at the edge server has a frequency of f_e [2]. Without loss of generality and for the ease of presentation, we can study the MEC system by investigating one IoT device and the associated CPU core at the edge server, which is abbreviated to be one edge server for the ease of presentation, in the following. The IoT device has a computation task to complete. The task may be computationally intensive and has to be completed within time T_{th}. To complete the computation task in time and to save the IoT device's energy consumption, the IoT device can offload all or part of the computation task to the edge server on the allocated spectrum. In order to capture a good opportunity for

---

[2]Considering the incoming and outgoing of IoT devices in the coverage of the edge server, it would be computationally prohibitive if the edge server performs joint allocation of communication and computation resources among the connected IoT devices every once the set of connected IoT devices alters, which generally corresponds to a mixed-integer optimization problem. The dedicated allocation of spectrum and computation capability to every IoT device enable every IoT device to make its task offloading decision individually and separately, which can save the edge server from heavy and frequent optimization.
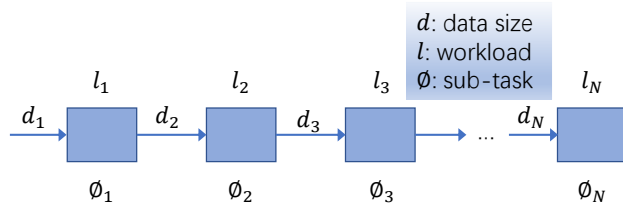
Fig. 1. Sequentially dependent task model. There are N consecutive sub-tasks in the mobile application and all sub-tasks have to be executed sequentially. For $i \in \{2, ..., N\}$, sub-task ϕ_i's input data, whose size is d_i, is also its precedent sub-task ϕ_{i-1}'s output data.

task offloading, instant channel gain between the IoT device and edge server will be measured before the IoT device makes the decision to offload. After the edge server completes the computation, it will return computational results to the IoT device, which is often of small data size and requires little time to be transmited from the edge server to the IoT device [11], [12], [23].

A. Computation Task Model

For the computation task, a sequentially dependent task model is considered. Specifically, the mobile application is assumed to be composed of N sub-tasks, denoted as ϕ_1, ϕ_2, ..., ϕ_N respectively, the indices of which constitute the set $\mathcal{N} \triangleq \{1, 2, ..., N\}$. These N sub-tasks have to be completed sequentially, i.e., one can only compute the sub-task of ϕ_i after completing the sub-task of ϕ_{i-1}, for $i = 2, 3, ..., N$. Every sub-task can be characterized by two parameters. For sub-task $i \in \mathcal{N}$, the associated parameter pair is (l_i, d_i), where l_i indicates the amount of computation (which is in the unit of CPU cycles) and d_i describes the amount of input data size (which is in the unit of nats for the ease of presentation). Note that with a sequentially dependent setup for the task model, the input data of sub-task ϕ_i, which has a size of d_i, is also the output data of sub-task ϕ_{i-1}. A diagram of the sequentially dependent task model is plotted in Fig. 1. With the selection of the sequentially dependent task model, the IoT device will first compute some sub-tasks at local and then offload the input data of the next sub-task to be computed to the edge server, at last the edge server will deal with the computing of the rest sub-tasks.

B. Computation Model

In terms of computation, it is composed of local computation both at the IoT device and edge server. Suppose the computation capability (which is also called CPU frequency) of the IoT device is $f_l \in [0, f_{\max}]$. Then the running power of the IoT device CPU is $k_0 f_l^3$ where k_0 is a physical parameter depending on the CPU's architecture according to [18], [27]. Hence to complete a computation task with the computation amount being l, the time consumed is l/f_l and the consumed energy is $k_0 f_l^2 l$. To save energy consumption of the IoT device considering that it is energy sensitive and its running power is largely determined by CPU frequency, f_l is assumed

to be adjustable over disjoint sub-tasks. For sub-task $i \in \mathcal{N}$, suppose the associated CPU frequency is f_i, then to complete a sub-task, say ϕ_i, the associated time consumption would be

$$t_m(i) = \frac{l_i}{f_i}, \forall i \in \mathcal{N}, \tag{1}$$

and the associated energy consumption would be

$$e_m(i) = k_0 \times l_i \times f_i^2, \forall i \in \mathcal{N}. \tag{2}$$

The computation capability of the edge server, i.e., the CPU frequency, is given as f_e where $f_e > f_{\max}$. According to the previous discussion, to complete a sub-task, say ϕ_j, the time consumption would be

$$t_e(j) = \frac{l_j}{f_e}, \forall j \in \mathcal{N}. \tag{3}$$

C. Communication Model

The communication activity in the system encompasses uploading the data of some sub-task from the IoT device to the edge server, and downloading the data of computational results from the edge server to the IoT device. In terms of data downloading, the data size of computational results is always small. In addition, the edge server has a strong power supply and can transmit information with a high data rate. Hence the time consumption for data downloading is omitted. This assumption is broadly adopted [12], [25].

In terms of task offloading, if the IoT device decides to stop local computing by sub-task ϕ_{n-1}, then it only needs to offload the input data for computing sub-task ϕ_n to the edge server, which has an amount of d_n nats. It is worth mentioning that the input data of any subsequent sub-task, including $\phi_{n+1}, \phi_{n+2}, ..., \phi_N$, is also the output data of its precedent sub-task and can be generated after completing its precedent sub-task at the edge server, considering the sequential dependency between adjacent sub-tasks. Hence the IoT device does not need to offload the input data of any sub-task among $\phi_{n+1}, ..., \phi_N$ to the edge server.

In the process of offloading these d_n nats, the corresponding *normalized channel* gain from the IoT device to the edge server is denoted as $\boldsymbol{h}_n$ for $n \in \mathcal{N}$ [3]. The normalized channel gain represents the received SNR at the edge server when the transmit power is 1. If the noise power at the edge server is 1, the normalized channel gain will coincide with the real channel gain. For ease of discussion in the following, we abbreviate every normalized channel gain as channel gain. When the timespan of task offloading is less than the channel coherence time, which is denoted as τ, the $\boldsymbol{h}_n$ would be a single random variable. Otherwise, the $\boldsymbol{h}_n$ should be split into multiple independently distributed fading blocks, which can be denoted as $\boldsymbol{h}_n = \left(h_{n,1}, h_{n,2}, h_{n,M(n)} \right)^T$. Detailed discussion on this point will be offered in the rest part of this subsection. In addition, the $\boldsymbol{h}_n$ no matter in case of scalar or vector is supposed to be i.i.d.

---

[3] The $\boldsymbol{h}_n$ may be a scalar or vector, as explained at the end of this paragraph. To be uniform, we write it in the form of bold font.

By comparing the time-scale of task offloading and sub-task computation at local with the channel coherence time τ, two categories of channel models can be expected.

1) Slow fading channel: In this type of channel model, channel coherence time would be longer than T_{th}. In this case, h_i does not change with i for $i \in \mathcal{N}$ in the whole process of completing the computation task, no matter the sub-task is being processed locally, offloaded, or processed at the edge server. Denote the unchanged h_i for $i \in \mathcal{N}$ as h, then the IoT device only needs to perform one measurement of h before it starts to execute the computation task.

With such an assumption, denote the transmit power of the IoT device as P_t when it is executing task offloading. Considering the fact that h is the normalized channel gain, the received SNR at the edge server can be written as $P_t h$ and the data transmission rate between the IoT device and edge server can be expressed as $r = W \ln(1 + P_t h)$. In order to offload d_n nats when the IoT device stops local computing by sub-task ϕ_{n-1}, the time consumption is

$$T_{\text{off}}^{\text{I}}(n) = \frac{d_n}{W \ln(1 + P_t h)}, \forall n \in \mathcal{N} \tag{4}$$

and the energy consumption can be written as

$$E_{\text{off}}^{\text{I}}(n) = \frac{d_n P_t}{W \ln(1 + P_t h)}, \forall n \in \mathcal{N}. \tag{5}$$

2) Fast fading channel: In this type of channel model, channel coherence time τ is much shorter than not only the time scale of computing any sub-task but also the time scale of offloading the input data of any sub-task. Therefore, for $n \in \mathcal{N}$, the task offloading time window, whose duration is supposed to be $T(n)$, may contain multiple (say $M(n)$) fading blocks, which are indexed as $(n,1)$, $(n,2)$, ..., $(n, M(n))$, respectively. In this case, $\boldsymbol{h}_n$ is a vector rather than a scalar, which is composed of independently distributed $h_{n,1}$, $h_{n,2}$, ..., $h_{n,M(n)}$, for $n \in \mathcal{N}$. In addition, we suppose the random variables $h_{n,1}$, $h_{n,2}$, ..., $h_{n,M(n)}$ are also subject to a common distribution function $p(\cdot)$ with a sample space $\mathcal{H}$.

Within $T(n)$, there are $M(n)$ fading blocks, which can be calculated as $M(n) = \lceil T(n)/\tau \rceil$. Define $t_{n,m}$ as the time duration of mth fading block for $n \in \mathcal{N}$, $m \in \{1, 2, ..., M(n)\}$, then there is

$$t_{n,m} = \begin{cases} \tau, & m < M(n), \\ \tilde{t}_n, & m = M(n), \end{cases}$$

where $\tilde{t}_n \in (0, \tau]$ characterizes the time duration of the ending fading block for task offloading, which may be less than τ since $T(n)$ may not be exactly integer multiple of channel coherence time τ. Hence $T(n)$ satisfies $T(n) = (M(n) - 1)\tau + \tilde{t}_n, \forall n \in \mathcal{N}$.

When the IoT device decides to offload the input data for computing sub-task ϕ_n, it will experience $M(n)$ fading blocks. In one fading block whose time duration is $t_{n,m}$ and channel gain is $h_{n,m}$, then to offload d nats, according to Shannon capacity formula, the transmit power can be written as $\left(e^{\frac{d}{W t_{n,m}}} - 1\right)/h_{n,m}$, and the associated energy consumption is $e(d, h_{n,m}, t_{n,m}) = \left(e^{\frac{d}{W t_{n,m}}} - 1\right) t_{n,m}/h_{n,m}$. Define

$Q_{n,m}(d, h; M(n), \tilde{t}_n)$ as the minimum amount of energy that the IoT device will consume for task offloading over the rest $(M(n) - m + 1)$ fading blocks under the following four conditions: 1) the channel gain at the beginning of fading block (n, m), i.e., $h_{n,m}$, is measured to be h; 2) there are d data nats to offload; 3) the total number of fading blocks for the task offloading is $M(n)$; 4) the last fading block, i.e, fading block $(n, M(n))$, has a time duration of $\tilde{t}_n$. Define $Q_{n,m}(d; M(n), \tilde{t}_n)$ as the minimum amount of energy that the IoT device will consume for the task offloading over the rest $(M(n) - m + 1)$ fading blocks when the IoT device has no knowledge of channel gain $h_{n,m}$ at the beginning of fading block (n, m) and the above mentioned conditions 2, 3, and 4 are satisfied.

With such a definition, when the IoT device finishes computing sub-task ϕ_{n-1}, it will first measure the channel gain $h_{n,1}$, then evaluate the function $Q_{n,1}(d_n, h_{n,1}; M(n), \tilde{t}_n)$, and finally make the decision on whether to offload the data for computing sub-task ϕ_n based on the evaluated $Q_{n,1}(d_n, h_{n,1}; M(n), \tilde{t}_n)$. How to make the decision according to $Q_{n,1}(d_n, h_{n,1}; M(n), \tilde{t}_n)$ will be discussed in Section IV-B. According to the discussion in [25], the functions of $Q_{n,m}(d, h; M(n), \tilde{t}_n)$ and $Q_{n,m}(d; M(n), \tilde{t}_n)$ for $m \in \{1, 2, ..., M(n)\}$ and $n \in \mathcal{N}$ can be calculated iteratively by following (6) and (7).

The $d_{n,m}$ in (7) represents the amount of data to be offloaded within fading block (n, m), and

$$\begin{aligned} &Q_{n,M(n)}\left(d, h_{n,M(n)}; M(n), \tilde{t}_n\right) \\ =&e\left(d, h_{n,M(n)}, t_{n,M(n)}\right) \\ =&\frac{\left(e^{\frac{d}{\tilde{t}_n W}} - 1\right)\tilde{t}_n}{h_{n,M(n)}}, \forall n \in \mathcal{N}. \end{aligned} \tag{8}$$

D. Problem Formulation

With the two channel models introduced in Section III-C, our target is to minimize the total energy consumption of the IoT device while guaranteeing the computation task is completed within time T_{th}. Specifically, the associated optimization problems are given as follows.

1) Slow fading channel: In this case, the total energy consumption of the IoT device is composed of two parts: 1) The energy consumption for local computing; 2) The energy consumption for task offloading. Hence combining the expressions in (2) and (5), the total energy consumption if the IoT device decides to offload data exactly after completing sub-task ϕ_{n-1} can be written as

$$E_1(n) = \frac{d_n P_t}{W \ln(1 + P_t h)} + \sum_{i=1}^{n-1} l_i f_i^2 k_0, \forall n \in \mathcal{N}, \tag{9}$$

where the second term of the right-hand side of (9) represents the energy consumption of the IoT device for computing sub-tasks ϕ_1, ϕ_2, ..., ϕ_{n-1} at local.

For the time consumption to complete the mobile application, if the IoT device decides to offload data exactly after

$$Q_{n,m}\left(d; M(n), \tilde{t}_n\right) = \int_0^\infty Q_{n,m}\left(d, h_{n,m}; M(n), \tilde{t}_n\right) p\left(h_{n,m}\right) dh_{n,m}, \forall m \in \{1, ..., M(n)\}, \forall n \in \mathcal{N}, \tag{6}$$

$$
\begin{aligned}
&Q_{n,m}(d, h_{n,m}; M(n), \tilde{t}_n) \\
&= \min_{0 \le d_{n,m} \le d}\left(e(d_{n,m}, h_{n,m}, t_{n,m}) + \int_0^\infty Q_{n,m+1}\left(d - d_{n,m}, h_{n,m+1}; M(n), \tilde{t}_n\right) p\left(h_{n,m+1}\right) dh_{n,m+1}\right), \\
&= \min_{0 \le d_{n,m} \le d}\left(e(d_{n,m}, h_{n,m}, t_{n,m}) + Q_{n,m+1}\left(d - d_{n,m}; M(n), \tilde{t}_n\right)\right), \forall m \in \{1, 2, ..., M(n) - 1\}, \forall n \in \mathcal{N}.
\end{aligned}
\tag{7}
$$

completing sub-task ϕ_{n-1}, by recalling (1), (3), and (4), the total time consumption, denoted as $T_1(n)$, can be written as

$$T_1(n) = \sum_{i=1}^{n-1} \frac{l_i}{f_i} + \frac{d_n}{W \ln(1 + P_t h)} + \sum_{i=n}^{N} \frac{l_i}{f_e}, \forall n \in \mathcal{N}. \tag{10}$$

Then we need to select the stopping time $n \in \mathcal{N}$ and optimize P_t and f_i for $i \in \{1, 2, ..., n-1\}$, so as to minimize the energy consumption in (9) while requiring the $T_1(n)$ to be no larger than T_{th}. Specifically, to minimize the total energy consumption of the IoT device, the following optimization problem needs to be solved [4]

Problem 1:

$$\min_{P_t, n, \{f_i\}} \quad \frac{d_n P_t}{W \ln(1 + P_t h)} + \sum_{i=1}^{n-1} l_i f_i^2 k_0$$

$$\text{s.t.} \quad \sum_{i=1}^{n-1} \frac{l_i}{f_i} + \frac{d_n}{W \ln(1 + P_t h)} + \sum_{i=n}^{N} \frac{l_i}{f_e} \le T_{th}, \tag{11a}$$

$$0 \le f_i \le f_{\max}, \forall i \in \{1, 2, ..., n-1\}, \tag{11b}$$

$$P_t \ge 0, \tag{11c}$$

$$n \in \mathcal{N}. \tag{11d}$$

2) Fast fading channel: In this case,
$h_{1,1}, ..., h_{1,M(1)}, h_{2,1}, ..., h_{2,M(2)}, ..., h_{N,1}, ..., h_{N,M(N)}$ are all i.i.d random variables. The IoT device cannot determine when to offload data in advance, due to the randomness of channel gains in forthcoming fading blocks. Therefore, an online policy is required to be designed.

When the IoT device finishes the computing of sub-task ϕ_{n-1} at local, it can measure the channel gain $h_{n,1}$ and then makes the decision on whether to offload the input data of sub-task ϕ_n right now. Hence before the IoT device determines to offload the computing data of sub-task ϕ_n, it has the knowledge of $h_1, h_2, ..., h_{n-1}, h_{n,1}$. In this case, the stopping rule n can be expressed as $n(h_1, ..., h_{n-1}, h_{n,1})$ or $n(h_1 \to h_{n,1})$ for brevity. Without confusion and for ease of presentation, we will interchange the use of n and $n(h_1 \to h_{n,1})$ in the

following discussion for this scenario. In terms of f_i for $i \in \{1, 2, ..., n-1\}$, from the perspective of the IoT device, it is unclear when it will switch to task offloading, not to mention how to dynamically adjust f_i over $i \in \{1, 2, ..., n-1\}$. To be equal with every sub-task, the IoT device will fix f_i at $f_l \in [0, f_{\max}]$ for $i \in \{1, 2, ..., n-1\}$.

In this scenario, the time delay for completing the whole mobile application can be written as

$$T_3(n, M(n), \tilde{t}_n) = \sum_{i=1}^{n-1} \frac{l_i}{f_l} + (M(n) - 1)\tau + \tilde{t}_n + \sum_{i=n}^{N} \frac{l_i}{f_e}, \forall n \in \mathcal{N}. \tag{12}$$

Then to meet the deadline requirement, there should be $T_3(n, M(n), \tilde{t}_n) \le T_{th}$.

The incurred energy consumption when the IoT device decides to stop local computing by sub-task ϕ_{n-1} can be written as

$$
\begin{aligned}
&E_3(n, M(n), \tilde{t}_n) \\
&= \quad Q_{n,1}(d_n, h_{n,1}; M(n), \tilde{t}_n) + \sum_{i=1}^{n-1} l_i f_l^2 k_0, \forall n \in \mathcal{N},
\end{aligned}
\tag{13}
$$

then to minimize the expected total energy consumption of the IoT device, the associated optimization problem can be formulated as

Problem 2:

$$\min_{n(h_1 \to h_{n,1}), M(n), \tilde{t}_n} \quad \mathbb{E}_{h_1, ..., h_N}\left\{E_3(n, M(n), \tilde{t}_n)\right\}$$

$$\text{s.t.} \quad n(h_1 \to h_{n,1}) \in \mathcal{N}, \tag{14a}$$

$$T_3(n, M(n), \tilde{t}_n) \le T_{th}, \tag{14b}$$

$$M(n) \in \mathcal{I}, \tag{14c}$$

$$0 < \tilde{t}_n \le \tau, \tag{14d}$$

where $\mathcal{I}$ is the integer set.

In Problem 2, it should be noticed that not only $n(h_1 \to h_{n,M(n)})$ but also $M(n)$ and $\tilde{t}_n$ are all online policies depending on $h_1, h_2, ..., h_{n-1}, h_{n,1}$, and the objective function represents the expected energy consumption $E_3(n, M(n), \tilde{t}_n)$ over the possible realizations of $h_1, h_2, ..., h_N$. Note that although $n(h_1 \to h_{n,1})$, $M(n)$, and $\tilde{t}_n$ are all random, they are not necessarily independent. Hence a restriction like (14b) can be imposed on these three random variables, which makes them to be correlated.

---

[4]Since the running time for working out the optimal solution of Problem 1 by the IoT device is ignorable compared with the time scale of T_{th}, which will be demonstrated in Section V-A, it is not necessary to take into account this part of time consumption in (11a). Even if it has to be accounted for, the T_{th} in constraint (11a) can be replaced with $(T_{th} - \Delta_c)$, where Δ_c represents the running time for solving the optimization problem. A similar discussion also follows for constraint (14b) in Problem 2.

IV. OPTIMAL SOLUTION

A. Optimal Solution of the Task Offloading and Resource Allocation Problem in Slow Fading Channels

In this subsection, the problem of task offloading and resource allocation in slow fading channels, which corresponds to Problem 1, would be solved. Problem 1 involves the optimizing of variables $n \in \mathcal{N}$, P_t, and f_i for $i \in \{1, 2, ..., n-1\}$, which contains both integer and continuous variables, and thus cannot be solved directly. To find the optimal solution of Problem 1, we decompose it into two levels. In the lower level, n is fixed while the rest variables P_t and f_i for $i \in \{1, 2, ..., n-1\}$ are optimized. The associated optimization problem can be written as

Problem 3:

$$E_I(n) \triangleq \min_{P_t, \{f_i\}} \frac{d_n P_t}{W \ln(1 + P_t h)} + \sum_{i=1}^{n-1} k_0 l_i f_i^2$$

$$\text{s.t.} \sum_{i=1}^{n-1} \frac{l_i}{f_i} + \frac{d_n}{W \ln(1 + P_t h)} + \sum_{i=n}^{N} \frac{l_i}{f_e} \leq T_{\text{th}}, \tag{15a}$$

$$0 \leq f_i \leq f_{\max}, \forall i \in \{1, 2, ..., n-1\}, \tag{15b}$$

$$P_t \geq 0. \tag{15c}$$

In the upper level, optimal n should be found so as to solve the following optimization problem

Problem 4:

$$\min_{n \in \mathcal{N}} E_I(n) \tag{16}$$

It can be checked that the upper level problem, Problem 4, is equivalent with the original optimization problem, Problem 1. In the following, the lower level problem, Problem 3 will be solved first. Inspecting the function $\frac{d_n P_t}{W \log(1 + P_t h)}$ in the objective function of Problem 3, it can be found that its second-order derivative with P_t is always non-positive, which implies the concavity of the function $\frac{d_n P_t}{W \log(1 + P_t h)}$. Thus Problem 3 is a non-convex optimization problem, considering that Problem 3 includes minimizing a concave function. Problem 3 is a non-convex optimization problem, since the function $\frac{d_n P_t}{W \log(1 + P_t h)}$ in the objective function of Problem 3 is non-convex with P_t. To simplify the solving of Problem 3, we define an assistant variable $\tau_t = \frac{d_t}{W \log(1 + P_t h)}$, which represents the time consumption for offloading the input data of sub-task ϕ_n. By expressing P_t with τ_t, i.e., $P_t = \left(e^{\frac{d_t}{W \tau_t}} - 1\right) \frac{1}{h}$, Problem 3 turns to be the following optimization problem

Problem 5:

$$E_I(n) \triangleq \min_{\tau_t, \{f_i\}} \tau_t \left(e^{\frac{d_n}{W \tau_t}} - 1\right) \frac{1}{h} + \sum_{i=1}^{n-1} k_0 l_i f_i^2$$

$$\text{s.t.} \sum_{i=1}^{n-1} \frac{l_i}{f_i} + \tau_t + \sum_{i=n}^{N} \frac{l_i}{f_e} \leq T_{\text{th}}, \tag{17a}$$

$$0 \leq f_i \leq f_{\max}, \forall i \in \{1, 2, ..., n-1\}, \tag{17b}$$

$$\tau_t \geq 0. \tag{17c}$$

For Problem 5, it can be checked that the function $\tau_t \left(e^{\frac{d_n}{W \tau_t}} - 1\right) \frac{1}{h}$ is convex with τ_t since its second-order derivative with τ_t is larger than 0. In addition, both the function $k_0 l_i f_i^2$ and the function $\frac{l_i}{f_i}$ are convex with f_i for $i \in \{1, 2, ..., n-1\}$. Hence Problem 5 is a convex optimization problem, which can be solved by traditional methods, such as the elliptical method or interior method [28].

To further simplify the solving of Problem 5, we target to design a simple yet optimal solution for Problem 5. To explore the special property of the optimal solution of Problem 5, with a given τ_t, we investigate the following optimization problem

Problem 6:

$$E_I^{\text{loc}}(\tau_t, n) \triangleq \min_{\{f_i\}} \sum_{i=1}^{n-1} k_0 f_i^2 l_i$$

$$\text{s.t.} \quad 0 \leq f_i \leq f_{\max}, \forall i \in \{1, 2, ..., n-1\}, \tag{18a}$$

$$\sum_{i=1}^{n-1} \frac{l_i}{f_i} \leq T_{\text{th}} - \tau_t - \sum_{i=n}^{N} \frac{l_i}{f_e}. \tag{18b}$$

In Problem 5, since $f_i \leq f_{\max}$ for $i \in \{1, 2, ..., n-1\}$ according to (17b), τ_t has to be no larger than $\left(T_{\text{th}} - \sum_{i=1}^{n-1} \frac{l_i}{f_{\max}} - \sum_{i=n}^{N} \frac{l_i}{f_e}\right)$ according to (17a). With this implicit constraint on τ_t and the definition of $E_I^{\text{loc}}(\tau_t, n)$ in Problem 6, Problem 5 is equivalent with the following optimization problem

Problem 7:

$$E_I(n) = \min_{\tau_t} \tau_t \left(e^{\frac{d_n}{W \tau_t}} - 1\right) \frac{1}{h} + E_I^{\text{loc}}(\tau_t, n)$$

$$\text{s.t.} \quad 0 \leq \tau_t \leq T_{\text{th}} - \sum_{i=1}^{n-1} \frac{l_i}{f_{\max}} - \sum_{i=n}^{N} \frac{l_i}{f_e}. \tag{19a}$$

Problem 6 is convex and satisfies the Slater's condition, then the KKT condition can serve as a sufficient and necessary condition for the optimal solution [28]. Specifically, the Lagrange function of Problem 6 can be written as

$$
\begin{aligned}
& L(\{f_i\}, \lambda, \{\mu_i\}, \{\nu_i\}) \\
= \ & \sum_{i=1}^{n-1} k_0 l_i f_i^2 - \sum_{i=1}^{n-1} \mu_i f_i \\
& + \lambda \left(\sum_{i=1}^{n-1} \frac{l_i}{f_i} + \tau_t + \sum_{i=n}^{N} \frac{l_i}{f_e} - T_{\text{th}}\right) + \sum_{i=1}^{n-1} \nu_i (f_i - f_{\max})
\end{aligned}
\tag{20}
$$

where $\lambda \geq 0$ is the Lagrange multiplier associated with constraint (18b), $\mu_i \geq 0$ and $\nu_i \geq 0$ are the Lagrange

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3150976, IEEE Internet of Things Journal

8

multipliers associated with constraints $f_i \geq 0$ and $f_i \leq f_{\max}$ in (18a) for $i \in \{1, 2, ..., n-1\}$, respectively. Then the KKT condition can be listed as in (21)

$$2k_0 f_i l_i - \lambda \frac{l_i}{f_i^2} - \mu_i + \nu_i = 0, \forall i \in \{1, 2, ..., n-1\} \quad (21a)$$

$$\sum_{i=1}^{n-1} \frac{l_i}{f_i} + \tau_t + \sum_{i=n}^{N} \frac{l_i}{f_e} \leq T_{\text{th}} \quad (21b)$$

$$0 \leq f_i \leq f_{\max}, \forall i \in \{1, 2, ..., n-1\} \quad (21c)$$

$$\lambda \left(\sum_{i=1}^{n-1} \frac{l_i}{f_i} + \tau_t + \sum_{i=n}^{N} \frac{l_i}{f_e} - T_{\text{th}} \right) = 0 \quad (21d)$$

$$\mu_i f_i, \forall i \in \{1, 2, ..., n-1\} \quad (21e)$$

$$\nu_i (f_i - f_{\max}), \forall i \in \{1, 2, ..., n\} \quad (21f)$$

$$\lambda \geq 0 \quad (21g)$$

$$\nu_i \geq 0, \mu_i \geq 0, \forall i \in \{1, 2, ..., n-1\} \quad (21h)$$

With KKT condition in (21), the optimal solution of f_i for $i \in \{1, 2, ..., n-1\}$ can be derived, which is given in the following theorem.

Theorem 1: The optimal solution of Problem 6 is $f_i = \frac{\sum_{i=1}^{n-1} l_i}{\left(T_{\text{th}} - \sum_{i=n}^{N} \frac{l_i}{f_e} - \tau_t \right)}$ for $i \in \{1, 2, ..., n-1\}$ when $\tau_t \in \left(0, T_{\text{th}} - \sum_{i=1}^{n-1} \frac{l_i}{f_{\max}} - \sum_{i=n}^{N} \frac{l_i}{f_e} \right]$.

Proof: Please refer to Appendix A. ∎

To this end, the optimal solution of Problem 6 has been found, and its minimal achievable cost function can be written as $E_I^{\text{loc}}(\tau_t, n) = k_0 \left(\sum_{i=1}^{n-1} l_i \right)^3 / \left(T_{\text{th}} - \sum_{i=n}^{N} l_i/f_e - \tau_t \right)^2$ which is actually a convex function with τ_t. Next we go back to solve Problem 7. Given the expression of $E_I^{\text{loc}}(\tau_t, n)$, it can be easily seen that for $\tau \in [0, T_{\text{th}} - \sum_{i=1}^{n-1} \frac{l_i}{f_{\max}} - \sum_{i=n}^{N} \frac{l_i}{f_e}]$, the objective function of Problem 7 is a convex function. Hence Problem 7 is a one-dimensional convex optimization problem, whose optimal solution can be found by the Golden search method [29]. To this end, Problem 7, which is also equivalent to Problem 3, has been solved optimally. Notice that our presented method only requires one dimension numerical search for the optimal τ_t while the optimal solutions of all the other variables, i.e., f_i for $i \in \{1, 2, ..., n-1\}$, are returned in closed-form. This procedure only involves a computation complexity of $O \left(\log \left(\frac{T_{\text{th}} - \sum_{i=1}^{n-1} \frac{l_i}{f_{\max}} - \sum_{i=n}^{N} \frac{l_i}{f_e}}{\delta} \right) \right)$, where δ is the tolerance. Compared with traditional numerical methods, which have a computation complexity $O(n^3)$ [28], our presented method can largely reduce the computation complexity.

In terms of solving Problem 4, by evaluating $E_I(n)$ for $n \in \mathcal{N}$, the n that associates the minimal $E_I(n)$ can be recognized as the optimal solution of Problem 4, and the optimal solution of Problem 1 as well. At the end of this subsection, the whole procedure for solving Problem 1 is summarized in the following algorithm.

Algorithm 1 The procedure for solving Problem 1.

1: **for** $n = 1 : N$ **do**
2:     With the given n, use the Golden section method to find the optimal τ_t for Problem 3, which is denoted as $\tau_t^*(n)$, and record the associated minimal achievable cost function $E_I(n)$.
3: Find $n^* = \underset{n \in \mathcal{N}}{\arg\min} \, E_I(n)$.
4: Output n^* as the optimal solution of n for Problem 1, $\left(e^{\frac{d_{n^*}}{W \tau_t^*(n^*)}} - 1 \right) \frac{1}{h}$ as the optimal P_t of Problem 1, $\frac{\sum_{i=1}^{n-1} l_i}{\left(T_{\text{th}} - \sum_{i=n}^{N} \frac{l_i}{f_e} - \tau_t^*(n^*) \right)}$ as the optimal f_i of Problem 1 for $i = 1, 2, ..., n^* - 1$.

Remark 1: In Algorithm 1, the computation complexity to evaluate $E_I(n)$ can be written as $O \left(\log \left(\left(T_{\text{th}} - \sum_{i=1}^{n-1} l_i/f_{\max} - \sum_{i=n}^{N} l_i/f_e \right) /\delta \right) \right)$. Hence the whole computation complexity to find the optimal solution of Problem 1 is $O \left(\sum_{n=1}^{N} \log \left(\left(T_{\text{th}} - \sum_{i=1}^{n-1} l_i/f_{\max} - \sum_{i=n}^{N} l_i/f_e \right) /\delta \right) \right)$.

B. Optimal Solution of the Task Offloading and Resource Allocation Problem in Fast Fading Channels

In this subsection, the problem of task offloading and resource allocation in fast fading channels, which corresponds to Problem 2, would be solved. In Problem 2, $n(\boldsymbol{h}_1 \to h_{n,1})$, $M(n)$, and $\tilde{t}_n$ are required to be solved but entangled in (14b). To free the entanglement of these online policies and find their optimal solutions, we decompose Problem 2 into two levels. In the lower level, under any realization of $\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N$, with $n(\boldsymbol{h}_1 \to h_{n,1})$ given, the rest variables, including $M(n)$ and $\tilde{t}_n$, are optimized. In the upper level, $n(\boldsymbol{h}_1 \to h_{n,1})$ is optimized over any realization of $\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N$. For the ease of presentation and without loss of correctness, n and $n(\boldsymbol{h}_1 \to h_{n,1})$ are used interchangeably in the following.

In the lower level, under any realization of $\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N$, with $n(\boldsymbol{h}_1 \to h_{n,1})$ given, Problem 2 dwells into Problem 8 since only $Q_{n,1}(d_n, h_{n,1}; M(n), \tilde{t}_n)$ involves $M(n)$ and $\tilde{t}_n$ in the objective function of Problem 2. The Problem 8 is given as follows

Problem 8:

$$Q(n, h_{n,1}) \triangleq \min_{M(n), \tilde{t}_n} \quad Q_{n,1}(d_n, h_{n,1}; M(n), \tilde{t}_n)$$
$$\text{s.t.} \quad T_3(n, M(n), \tilde{t}_n) \leq T_{\text{th}}, \quad (22a)$$
$$M(n) \in \mathcal{I}, \quad (22b)$$
$$0 < \tilde{t}_n \leq \tau, \quad (22c)$$

where the newly defined $Q(n, h_{n,1})$ represents the energy consumption for offloading sub-task ϕ_n's computing data with the measurement of $h_{n,1}$. [5]

In the upper level, $n(\boldsymbol{h}_1 \to h_{n,1})$ is required to be optimized to minimize the objective function of Problem 2 over any realization of $\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N$, which can be also written as $\left(Q(n, h_{n,1}) + \sum_{i=1}^{n-1} l_i f_l^2 k_0 \right)$ since $Q(n, h_{n,1})$ represents the

[5]

minimized $Q_{n,1}(d_n, h_{n,1}; M(n), \tilde{t}_n)$ over $M(n)$ and $\tilde{t}_n$. In such a case, it is equivalent to solve Problem 9, which is given as follows

Problem 9:

$$\min_n \quad \mathbb{E}_{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N} \left\{ Q(n, h_{n,1}) + \sum_{i=1}^{n-1} l_i f_l^2 k_0 \right\}$$

$$\text{s.t.} \quad n(\boldsymbol{h}_1 \to h_{n,1}) \in \mathcal{N}. \tag{23a}$$

We start by solving Problem 8. For Problem 8, to find its optimal solution, the following lemmas can be expected.

Lemma 1: With n, m, h, and d given, where $n \in \mathcal{N}$, $m \in \{1, 2, ..., M(n)\}$, and $d \geq 0$, there is $Q_{n,m}(d, h; M(n), t_1) \geq Q_{n,m}(d, h; M(n), t_2)$ for $0 \leq t_1 \leq t_2 \leq \tau$.

Proof: Please refer to Appendix B. ∎

Lemma 2: With n and d given, where $n \in \mathcal{N}$ and $d \geq 0$, for $m_1 \leq m_2$, there is $Q_{n,1}(d, h; m_1, \tau) \geq Q_{n,1}(d, h; m_2, \tau)$.

Proof: Please refer to Appendix C. ∎

Lemma 1 implies that the function $Q_{n,m}(d, h; M(n), t)$ is a non-increasing function with t for $t \in [0, \tau]$. Note that the original domain of definition for t is $(0, \tau]$, while the case $t = 0$ is considered in Lemma 1. It should be noticed that $Q_{n,m}(d, h; M(n), t) = Q_{n,m}(d, h; M(n) - 1, \tau)$ when $t = 0$. Hence the case that $t = 0$ is not weird. What is even further, the case that $t = 0$ can facilitate the proof for Theorem 2. Lemma 2 shows that the function $Q_{n,1}(d, h; m', \tau)$ is a non-increasing function with m'. With the help of Lemma 1 and Lemma 2, Theorem 2 can be expected, which can help to generate the optimal solution of Problem 8.

Theorem 2: To solve Problem 8, it is optimal to select $M(n) \in \mathcal{I}$ and $\tilde{t}_n \in (0, \tau]$ such that $T_3(n, M(n), \tilde{t}_n) = T_{\text{th}}$.

Proof: Please refer to Appendix D. ∎

With Theorem 2, the optimal $M(n)$ and t of Problem 8 have been found for a given n, which is defined as $M^*(n)$ and $\tilde{t}_n^*$ respectively. In the next step, we need to solve Problem 9. For Problem 9, the method of dynamic programming can be utilized to find the optimal solution [30].

Specifically, define

$$Z_N(h_{N,1}) = \min\left(Q_{N,1}(d_N, h_{N,1}; M^*(N), \tilde{t}_N^*), k_0 l_N f_l^2 \right), \tag{24}$$

$$Z_n(h_{n,1}) = \min\left(Q_{n,1}(d_n, h_{n,1}; M^*(n), \tilde{t}_n^*), k_0 l_n f_l^2 + Z_{n+1} \right), \quad \forall n \in \mathcal{N} \setminus \{N\}. \tag{25}$$

and

$$Z_n = \int_0^\infty Z_n(h_{n,1}) p(h_{n,1}) dh_{n,1}, \forall n \in \mathcal{N}. \tag{26}$$

According to the definition of $Q(n, h_{n,1})$ in Problem 8, it seems that $M(n)$ and $\tilde{t}_n$ only depend on $h_{n,1}$ when $n(\boldsymbol{h}_1 \to h_{n,1})$ is given. It should be noticed that this does not mean $M(n)$ and $\tilde{t}_n$ is irrelevant to $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, ..., $\boldsymbol{h}_{n-1}$. In essence, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, ..., $\boldsymbol{h}_{n-1}$ indirectly determine $M(n)$ and $\tilde{t}_n$, which is given in such a way: They first affect $n(\boldsymbol{h}_1 \to h_{n,1})$ and then the $n(\boldsymbol{h}_1 \to h_{n,1})$ affects $M(n)$ and $\tilde{t}_n$. Hence the definition of $Q(n, h_{n,1})$ given in Problem 8 is not contradict with the claim after Problem 2 and $M(n)$ and $\tilde{t}_n$ are still the policies depending on $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, ..., $h_{n,1}$.

It can be speculated that $Z_n(h_{n,1})$ is the minimal expected forthcoming energy consumption of the IoT device when the IoT device has just completed the local computing of sub-task ϕ_{n-1} and has measured the channel gain of $h_{n,1}$ for $n \in \mathcal{N}$. In addition, Z_n is the minimal expected forthcoming energy consumption of the IoT device when the IoT device has completed the local computing of sub-task ϕ_{n-1} but has no knowledge of $h_{n,1}$ for $n \in \mathcal{N}$. For the ease of presentation, we also define $Z_{N+1} = 0$. Then (25) will also be effective for $n = N$.

To achieve minimal energy consumption, the IoT device should decide whether to offload the input data of the next sub-task after it completes local computing of every sub-task, until it has made the decision to stop local computing. Suppose the IoT device has just completed the local computing of sub-task ϕ_{n-1} and has measured the channel gain $h_{n,1}$, then it will stop local computing and starts task offloading of sub-task ϕ_n if $Q_{n,1}(d_n, h_{n,1}; M^*(n), \tilde{t}_n^*) < \left(k_0 l_n f_l^2 + Z_{n+1} \right)$ and continue local computing otherwise. In summary, the solving procedure of Problem 2 can be summarized in Algorithm 2 as follows.

Algorithm 2 The procedure for solving Problem 2.

1: Offline Part:
2: Utilizing (6), (7), (8) and with the aid of Theorem 2 to calculate $Q_{n,m}(d_n, h_{n,1}; M(n)^*, \tilde{t}_n^*)$ for $n \in \mathcal{N}$ and $m \in \{1, 2, ..., M^*(n)\}$. Mark $Q_{n,1}(d_n, h_{n,1}; M(n)^*, \tilde{t}_n^*)$ as $Q_{n,1}(d_n, h_{n,1})$ for $n \in \mathcal{N}$. Utilizing (24), (25) and (26) to calculate $Z_n(h_{n,1})$ and Z_n for $n \in \mathcal{N}$. Store the calculated Z_n, and $Z_n(h_{n,1})$, $Q_{n,m}(d_n)$, $Q_{n,m}(d_n, h_{n,1})$ and the calculated optimal $d_{n,m}$ for solving the optimization problem in (7) at the IoT device before it starts computing the whole task. Store the calculated Z_n, and $Z_n(h_{n,1})$ for $n \in \mathcal{N}$ and $m \in \{1, 2, ..., M^*(n)\}$ at the IoT device.
3: Online Part:
4: **while** $n \leq N$ and the IoT device has not stopped local computing **do**
5:    **if** $Q_{n,1}(d_n, h_{n,1}; M^*(n), \tilde{t}_n^*) < \left(k_0 l_n f_l^2 + Z_{n+1} \right)$ **then**
6:       Stop local computing and upload sub-task ϕ_n to the edge server. In the procedure of task offloading, adopt the stored optimal solution of $d_{n,m}$ for solving (7) to realize the expected energy consumption $Q_{n,1}(d_n, h_{n,1}; M^*(n), \tilde{t}_n^*)$.
7:    **else**
8:       Complete the computing of sub-task ϕ_n at local.
9:    $n = n + 1$

Remark 2: In terms of computation complexity of Algorithm 2, exponential complexity would be involved in the procedure of iterative computing for $Q_{n,m}(d, h; m, t)$. However, this part of computation belongs to offline computing, which can be completed by the edge server and does not impose a computational burden on the IoT device. With a table of $Q_{n,m}(d, h; m, t)$ stored at the IoT device, it only needs to make a comparison between two values in each step, whose computation complexity is ignorable.

C. Further Discussion for Fast Fading Channels

In fast fading channels, the τ may be very small compared with the time scale for offloading the input data of one sub-task. To get some insight into this case, we will analyze the trend when τ approaches zero in the following. Suppose the IoT device selects to offload the data for computing sub-task ϕ_n. As τ converges to zero, $M(n)$ will converge to infinity and the $\tilde{t}_n$ can be taken as τ without loss of generality. In this case, looking into the expression of $Q_{n,1}(d, h_{n,1}; M(n), \tau)$, there is

$$Q_{n,1}(d, h_{n,1}; M(n), \tau) = \min_{0 \le d_{n,1} \le d} \left(e(d_{n,1}, h_{n,1}, \tau) + Q_{n,2}(d - d_{n,1}; M(n), \tau) \right). \quad (27)$$

As τ approaches zero, the optimal $d_{n,1}$ for achieving $Q_{n,1}(d, h_{n,1}; M(n), \tilde{t}_n)$, denoted as $d_{n,1}^*$, would also converge to zero. Otherwise an infinite amount of data would be offloaded within time $T(n)$. Hence there is $Q_{n,2}(d - d_{n,1}^*; M(n), \tilde{t}_n) = Q_{n,2}(d; M(n), \tilde{t}_n)$. Combing with the fact that $Q_{n,1}(d, h_{n,1}; M(n), \tau) \le Q_{n,2}(d; M(n), \tau)$ according to (40) and $e(d_{n,1}^*, h_{n,1}, t_{n,1}) \ge 0$. It can be easily derived that $e(d_{n,1}^*, h_{n,1}, t_{n,1}) = 0$. Therefore, there is

$$\begin{aligned} Q_{n,1}(d, h_{n,1}; M(n), \tilde{t}_n) &= Q_{n,2}(d; M(n), \tilde{t}_n) \\ &= Q_{n,1}(d; M(n), \tilde{t}_n). \end{aligned} \quad (28)$$

The equation (28) indicates that there is no difference in the expected energy consumption for task offloading no matter how much the $h_{n,1}$ is. In this case, we only need to evaluate the minimal expected energy consumption, i.e., $Q_{n,1}(d; M(n), \tilde{t}_n)$, rather than measure the $h_{n,1}$, before deciding to offload the sub-task ϕ_n. However, since $M(n)$ is approaching infinity, it is not practical to calculate $Q_{n,1}(d; M(n), \tilde{t}_n)$ by the iterative method presented in (7). To overcome this problem, recalling the fact that there is an infinite number of independently faded blocks within time $T(n)$, it is reasonable to assume that the process of $h_{n,1}, h_{n,2}, ..., h_{n,M(n)}$ are subject to an ergodic process. Therefore, within time $T(n)$, the amount of data for task offloading can be written as $\mathbb{E}\{W \log(1 + p_n(h))\}T(n)$, where $p_n(h)$ represents IoT device transmit power when instant channel state is h, and the expected energy consumption can be written as $\mathbb{E}\{p_n(h)\}T(n)$. Then the energy consumption minimization problem for offloading d_n nats can be reformulated as follows

Problem 10:

$$\min_{\{p_n(h)|h \in \mathcal{H}\}} \mathbb{E}\{p_n(h)\} T(n)$$
$$\text{s.t.} \quad \mathbb{E}\{W \log(1 + p_n(h))\} T(n) \ge d_n, \quad (29a)$$
$$p_n(h) \ge 0, \forall h \in \mathcal{H}, \quad (29b)$$

which is equivalent with the following optimization problem

Problem 11:

$$\min_{\{p_n(h)|h \in \mathcal{H}\}} \mathbb{E}\{p_n(h)\}$$
$$\text{s.t.} \quad \mathbb{E}\{\log(1 + p_n(h))\} \ge \frac{d_n}{T(n)W}, \quad (30a)$$
$$p_n(h) \ge 0, \forall h \in \mathcal{H}. \quad (30b)$$

Problem 11 is a convex optimization problem and falls into the category of the stochastic optimization problem. The Lagrangian of Problem 11 can be written as $\mathcal{L}(\{p_n(h)\}, \zeta) = \mathbb{E}\{p_n(h)\} - \zeta\left(\mathbb{E}\{\log(1 + p_n(h)h)\} - \frac{d_n}{T(n)W}\right)$, where ζ is the non-negative Lagrange multiplier associated with the constraint (30a). Then the Lagrange dual function can be written as

$$D(\zeta) = \min_{\{p_n(h) \ge 0|h \in \mathcal{H}\}} \mathcal{L}(\{p_n(h)\}, \zeta). \quad (31)$$

Given the fact that Problem 11 is a convex optimization problem and the Slater condition is satisfied [28], the maximum of the function $D(\zeta)$ over $\zeta \ge 0$ is also the minimum of Problem 11. To find the maximum of $D(\zeta)$ over $\zeta \ge 0$, a sub-gradient method can be utilized [31], which updates the ζ in the following way until the ζ converges

$$\zeta(i+1) = \left(\zeta(i) - a(i)\left(\frac{d_n}{T(n)W} - \mathbb{E}\{\log(1 + p_n(h, i)h)\}\right)\right)^+. \quad (32)$$

In (32), $\zeta(i)$ is the Lagrange multiplier ζ at ith iteration, $p_n(h, i)$ is the optimal $p_n(h)$ to achieve $D(\zeta(i))$ at ith iteration for $h \in \mathcal{H}$, $a(t)$ is the positive step size at ith iteration, and $(x)^+ = \max(x, 0)$. It should be noticed that $a(t)$ should satisfy $\sum_{i=0}^{\infty} a(i) = \infty$ and $\sum_{i=0}^{\infty} a(t)^2 < \infty$. One candidate $a(i)$ is $a(i) = 1/i$.

For a given ζ, to achieve $D(\zeta)$, the optimal $p_n(h)$ of the following optimization problem for every realization of h should be derived

Problem 12:

$$\min_{p_n(h)} p_n(h) - \zeta \log(1 + p_n(h)) + \zeta \frac{d_n}{T(n)W}$$
$$\text{s.t.} \quad p_n(h) \ge 0. \quad (33a)$$

For Problem 12, the optimal $p_n(h)$ can be easily obtained by setting the derivative of its objective function over $p_n(h)$ to be zero when $p_n(h) \ge 0$, which can be presented as $p_n(h) = \left(\zeta - \frac{1}{h}\right)^+$.

To this end, Problem 10 has been solved and how much energy would be consumed for offloading d_n nats when the length of one fading block τ goes to zero has been answered. Note that in this case, the minimal energy consumption for offloading d_n nats is a fixed number, which can be denoted as $Q(n)$, rather than a random number depending on $h_{n,1}$ anymore. Hence the optimal n can be easily found among $\{1, 2, ..., N\}$ by selecting the n with the minimal associated total energy consumption, which can be written as $\sum_{i=1}^{n} l_i f_l^2 k_0 + Q(n)$.

Remark 3: According to the discussion in this subsection, it can be found that as τ decreases to zero, the online policy derived in Section IV-B, which depends on the instant $h_{n,1}$ in the nth step, trends to converge to an offline policy that can predetermine the optimal n before the IoT device starts to run the whole computation task. Compared with online policy, offline policy can save the calculation of $Q_{n,m}(d, h; M(n), \tilde{t}_n)$ for every $m \in \{1, ..., M(n)\}$ and $h \in \mathcal{H}$, which will involve a lot of computation. Hence when τ is relatively small compared with the time scale for task offloading, offline policy discussed in this subsection would be preferred.

V. NUMERICAL RESULTS

In this section, numerical results are presented. Default system parameters are set as follows. By referring to the setup in [18], a computation task with 10 sub-tasks is adopted. The input data size d_n of these 10 sub-tasks are given as $\{36, 22, 30, 6, 47, 30, 5, 47, 14, 49\}$ k bits, and the workload l_n for $n \in \mathcal{N}$ are selected to be $\{7, 30, 25, 16, 32, 15, 37, 44, 24, 40\}$ M CPU cycles. Similar to [32], k_0 is set as 10^{-28}. The channel coherence time, i.e., the length of one fading block, τ is set as 20ms. The deadline for completing the whole computation task T_{th} is selected to be 350ms. [6]. The CPU frequency of the edge server $f_e = 3$ GHz. The maximal computation capability of the IoT device $f_{\max} = 500$ MHz. The computation capability of the IoT device in the case of fast fading channels f_l is also set as 500 MHz [18]. The bandwidth $W = 1$ MHz. In the case of fast fading channels, the channel is Rayleigh distributed, which means that the channel gain $h_{n,m}$ is subjected to an exponential distribution. The mean of $h_{n,m}$ is set as 50 [7]. All the simulations are run on a desktop with the CPU of Intel i7-7700K working at a frequency of 4.2 GHz.

A. Slow Fading Channels

Fig. 2(a) compares the IoT device's energy consumption between our proposed method in Section IV-A and the methods in [18] and [19], which represent existing strategies dealing with sequential task and binary offloading respectively, under various edge server's CPU frequency f_e [8]. For the method in [18], the transmit power, CPU frequency at the IoT device and the edge server are not optimized, so we adopt its default setting [18]. For the method in [19], we utilize its energy consumption model to generate an offloading decision in the framework of binary offloading. It can be observed that our proposed method always outperforms the methods in [18] and [19]. This result can be also found in Figs. 2(b), 3(a), and 3(b). These results prove the effectiveness of our proposed method. Additionally, it can be seen that the energy consumption under the method in [18] goes down like a "staircase". This can be

explained as follows: The transmit power and CPU frequency are fixed in [18]. Hence the energy consumption can be only reduced when the IoT device finds a better occasion to offload (by deciding to offload since another sub-task) for the method in [18]. Similar trends can be also found in Fig. 2(b) due to the same reason, which will not be discussed anymore in the part of Fig. 2(b). It can be also seen that as the CPU frequency of the edge server f_e grows, the associated energy consumption trends to decrease for the proposed method in Section IV-A. This is because the increase of f_e can contribute to the enlargement of Problem 1's feasible region. Hence the energy consumption under our proposed method, which is also the minimum of Problem 1, would decrease.

Fig. 2(b) plots IoT device's energy consumption with our proposed method in Section IV-A and the methods in [18] and [19] versus the deadline for completing the whole computation task T_{th}, when the channel gain h is set as 40 and 60, respectively. From Fig. 2(b), we can observe that for a fixed channel gain, the IoT device's minimal energy consumption decreases as T_{th} grows. This is due to the fact that the increase of T_{th} can help to relax the feasible region of Problem 1. Hence the IoT device's minimal energy consumption, which is also the minimum of Problem 1, would decrease.

Fig. 2(c) plots the running time when the number of sub-tasks N varies. When $N > 10$, we rerun the default computation task. It can be seen that the average running time to work out Problem 1 is around 3.0 ms, which is ignorable compared with the scale of T_{th}. This is attributed to the fact that one Golden search, which is simple to realize, is enough to find out the optimal solution of Problem 1, as shown in this paper. Thanks to the fast-growing manufacturing process of CPU in recent years, the computation capability of one IoT device nowadays does not lose to our desktop running the simulation. Hence we can claim the computation burden for working out the optimal solution of Problem 1 at the IoT device is ignorable when evaluating the time consumption for completing the computation task.
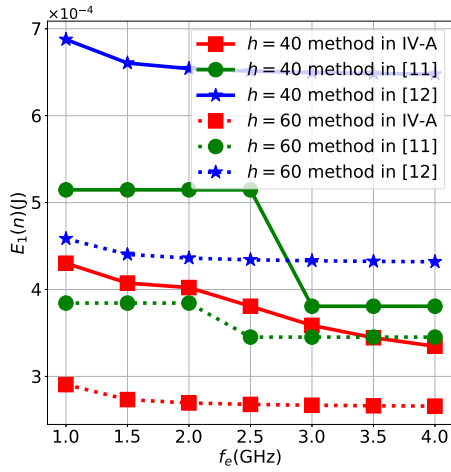
B. Fast Fading Channels

Fig. 3(a) illustrates IoT device's expected energy consumption for our proposed method in Section IV-B and the methods in [18] and [19] when edge server's CPU frequency f_e varies. In the legend of Fig. 3(a), "$k = 1.0$" and "$k = 0.96$" indicate the scaling factor imposed on the input data size of every sub-task. In terms of the trends of the methods in [18] and [19], the energy consumption for the method in [18] still goes down like a "staircase", which can be explained as the same way for Fig. 2(a), while the energy consumption for the method in [19] gradually decrease with f_e. The reason behind the behavior of the method in [19] can be explained as follows. In [19], a table is also built between expected energy consumption and data amount for offloading, which represents both continuity and convexity for the expected energy consumption with respect to the data amount for offloading as we have in our proposed method. Therefore a similar trend with our proposed method can be seen for the method in [19]. On the other hand, the table building procedure does not consider the optimization

[6]This time delay is necessary by evaluating the computation capability of the edge server and the workload of the investigated computation task.

[7]Considering the fact that $h_{n,m}$ actually represents the received SNR when the transmit power of the IoT device is 1, the selection of this number level is reasonable.

[8]The other literature dealing with the sequential task, including [17] and [26], are not compared due to similarity with the method in [18] or model mismatch.

(a) Energy consumption versus f_e.
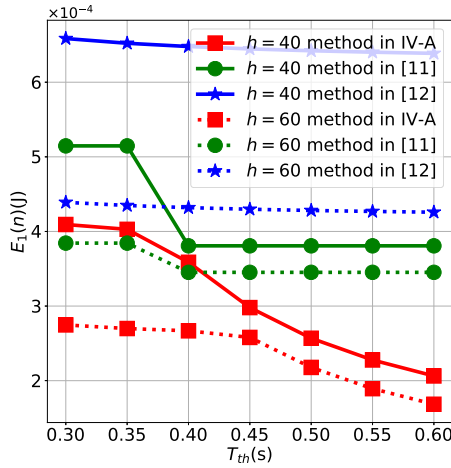
(b) Energy consumption versus T_{th}.
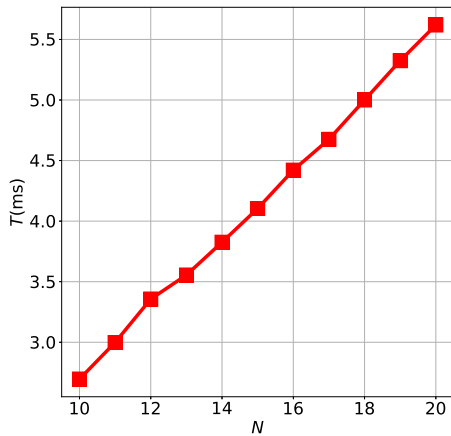
(c) Running time vs. Numbers of sub-tasks N.

Fig. 2. Performance analysis under slow fading channel.

(a) Energy consumption versus f_e.

(b) Energy consumption versus T_{th}.

(c) Running time vs. Numbers of sub-tasks N.

Fig. 3. Performance analysis under fast fading channel.

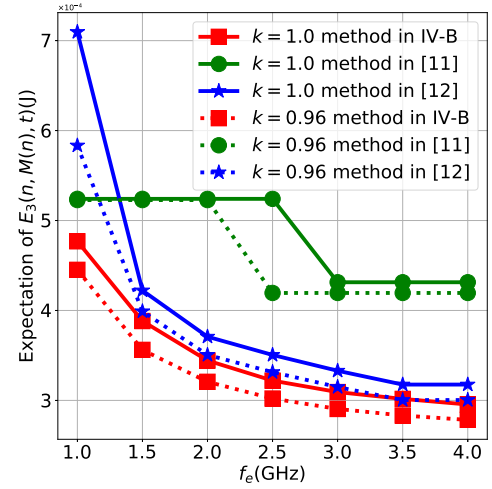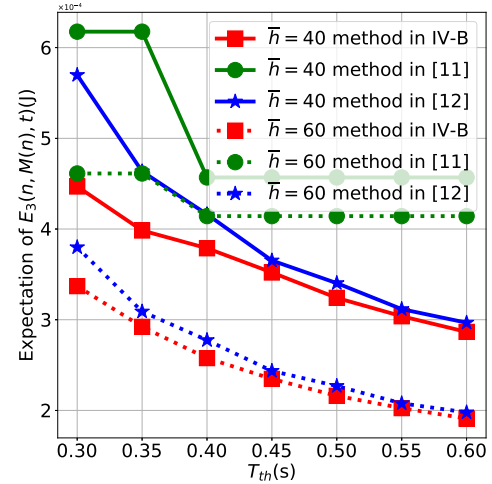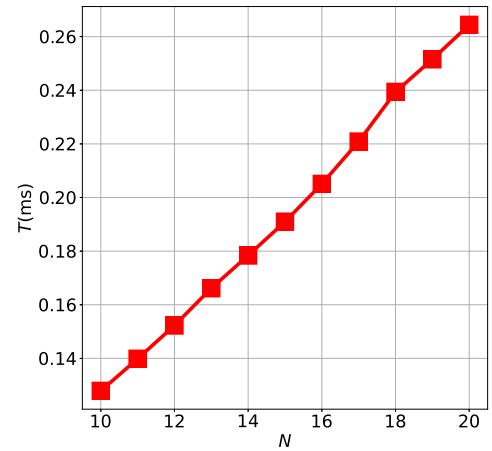This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3150976, IEEE Internet of Things Journal

13

of transmit power and will surely lead to higher energy consumption compared with our proposed method. Hence the method in [19] will always have weaker performance than our proposed method. Similar trends can be also found in Fig. 3(b) based on the same reason, which is not further discussed in the part of Fig. 3(b). It can be also observed that IoT device's expected energy consumption decreases as f_e grows, which can be explained in the same way as the case in the slow fading channel.

Fig. 3(b) plots IoT device's expected energy consumption for our proposed method in Section IV-B and the methods in [18] and [19] when T_{th} varies. It can be observed that IoT device's expected energy consumption for our proposed method will decrease with T_{th} when the mean of channel gain is set as 40 and 60, respectively. This can be explained as follows. Enlarged T_{th} can help to relax the feasible region of Problem 2. Hence the expected energy consumption, which is also the minimum of Problem 2, would decrease.

Fig. 3(c) illustrates the average running time of online computing for solving Problem 2 as the number of sub-tasks N varies. When $N > 10$, we rerun the default computation task. It can be observed that the running time is at the scale of 0.14 ms, while the offline computing time is at the scale of 1793.47 ms. As discussed in Remark 2, offline computing can be completed in advance by the edge server. Hence the computation burden for the IoT device, which merely comes from online computing, is ignorable (considering the rough equivalence of the computation capability between the IoT device and our simulating desktop as discussed in Section V-A) when evaluating the time consumption for solving Problem 2.

Fig. 4(a) plots the function $Q_{n,1}(d, h; m, t)$ with $d = 1.2 \times 10^6$ bits when the duration of the last fading block t varies from 5 ms to 20 ms. The channel gain of the first fading block h is set as 60. It can be observed that the $Q_{n,1}(d, h; m, t)$ degrades as the duration of the last block t increases when the number of fading blocks is fixed, which verifies the claim in Lemma 1. It can be also checked that the $Q_{n,1}(d, h; m, t)$ will decrease with the number of fading blocks when the duration of the last fading block t is fixed, which verifies the claim in Lemma 2.

In Fig. 4(b), the minimal expected energy consumption for offloading $d_n = 1.2 \times 10^6$ bits, i.e.,
$Q_{n,1}(d_n, h_{n,1}; M(n), \tau)$, under the various selection of $h_{n,1}$ is plotted versus channel coherence time τ when $n = 1$. The $T(n)$ is set as 840ms. As a comparison, the minimal $\mathbb{E}\{p_n(h)T(n)\}$ by solving Problem 10 is also plotted. It can be observed that as τ decreases, $Q_{n,1}(d, h_{n,1}; M(n), \tau)$, no matter what $h_{n,1}$ is, trends to converge to the value of minimal $\mathbb{E}\{p_n(h)T(n)\}$. This verifies the discussion in Section IV-C.

VI. CONCLUSION

In this paper, we considered a MEC system with sequentially dependent computation tasks under both slow and fast fading channels with the purpose of supporting intelligent applications for an IoT network. To minimize IoT device energy consumption, the optimization of task offloading strategy, communication resource allocation, and computation resource
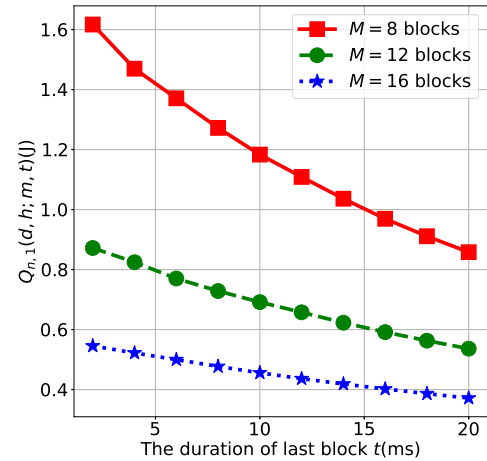
(a) Verification of Lemma 1 and Lemma 2.
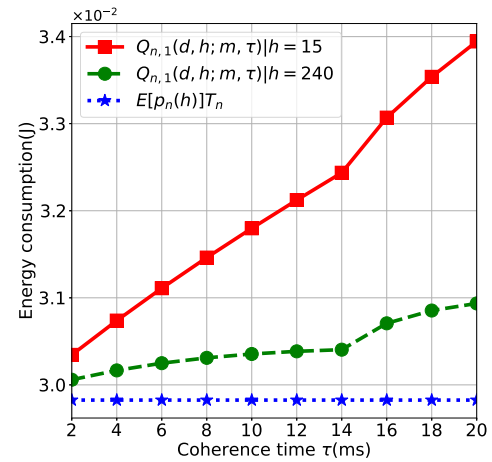
(b) Verification of the discussion in IV-C.

Fig. 4. Verification of lemmas and discussion.

allocation are performed. For slow fading channels, with a task offloading decision given, a simple and optimal solution of CPU frequency for local computing and transmit power for task offloading were found by dividing the associated optimization problem into two levels. The Golden search method is shown to be able to find the optimal solution in the upper level and a closed-form solution is derived in the lower level. Then the optimal task offloading decision can be easily found by one-dimensional search. For fast fading channels, an optimal online task offloading policy in response to the instant channel state was derived. It was also disclosed that the derived online policy trends to be an offline policy as the channel coherence time approaches zero, which can help to save the computation complexity when the channel coherence time is small enough. Our research results could provide helpful insights for a MEC-supported IoT network running intelligent applications with sequential tasks.

APPENDIX A
PROOF OF THEOREM 1

Since $f_i \in [0, f_{\max}]$ for $i \in \{1, 2, ..., n - 1\}$, the discussion will be unfolded from three possible cases: 1) $f_i = 0$; 2)

$0 < f_i < f_{\max}$; 3) $f_i = f_{\max}$. For the first case, it is evident that the optimal f_i for Problem 6 cannot be zero for $i \in \{1, 2, ..., n-1\}$. Otherwise, the objective function of Problem 6 would be infinite. Hence the case that $f_i = 0$ for $i \in \{1, 2, ..., n-1\}$ can be precluded. Define $\mathcal{A}_1(n-1) = \{i | 0 < f_i < f_{\max}, i \in \{1, 2, ..., n-1\}\}$ and $\mathcal{A}_2(n-1) = \{i | f_i = f_{\max}, i \in \{1, 2, ..., n-1\}\}$, then there is $\mathcal{A}_1(n-1) \cup \mathcal{A}_2(n-1) = \{1, 2, ..., n-1\}$.

With regard to the second case, we can first derive that $\mu_i = \nu_i = 0$ for $i \in \mathcal{A}_1(n-1)$ according to (21e) and (21f). Therefore, for $i \in \mathcal{A}_1(n-1)$, Eqn. (21a) dwells into $2k_0 f_i l_i = \lambda \frac{l_i}{f_i^2}, \forall i \in \mathcal{A}_1(n-1)$, which further indicates that

$$f_i = \left(\frac{\lambda}{2k_0} \right)^{\frac{1}{3}}, \forall i \in \mathcal{A}_1(n-1). \tag{34}$$

In other words, for $i \in \mathcal{A}_1(n-1)$, all the f_is are equal.

With regard to the third case, it can be also derived that $\mu_i = 0$ for $i \in \mathcal{A}_2(n-1)$ according to (21e). Then Eqn. (21a) turns to $2k_0 f_i l_i = \lambda \frac{l_i}{f_i^2} - \nu_i, \forall i \in \mathcal{A}_2(n-1)$, which can be further written as

$$\lambda = 2k_0 f_{\max}^3 + \frac{f_{\max}^2 \nu_i}{l_i}, \forall i \in \mathcal{A}_2(n-1) \tag{35}$$

since $f_i = f_{\max}$ for $i \in \mathcal{A}_2(n-1)$. On the other hand, Eqn. (34) indicates that

$$\lambda = 2k_0 f_i^3, \forall i \in \mathcal{A}_1(n-1). \tag{36}$$

Combing (35), (36), and the facts that $f_i < f_{\max}$ for $i \in \mathcal{A}_1(n-1)$ and $\nu_j \geq 0$ for $j \in \mathcal{A}_2(n-1)$, there is $\lambda = 2k_0 f_i^3 < 2k_0 f_{\max}^3 \leq 2k_0 f_{\max}^3 + \frac{f_{\max}^2 \nu_j}{l_j} = \lambda$, which contradicts the fact the $\lambda = \lambda$. Hence the case that $\mathcal{A}_1(n-1) \neq \emptyset$ and $\mathcal{A}_2(n-1) \neq \emptyset$ cannot coexist. In other words, only two possible cases for the optimal solution of Problem 6 could happen: 1) $\mathcal{A}_1(n-1) = \{1, 2, ..., n-1\}$ while $\mathcal{A}_2(n-1) = \emptyset$; 2) $\mathcal{A}_2(n-1) = \{1, 2, ..., n-1\}$ while $\mathcal{A}_1(n-1) = \emptyset$. It should be noticed that all the f_is are equal in every possible case.

On the other hand, we notice that the objective function of Problem 6 is monotonically increasing function with f_i while the left-hand side of constraint (18b) is decreasing function with f_i for $i \in \{1, 2, ..., n-1\}$. Hence to minimize the objective function of Problem 6, it would be optimal to decrease f_i as much as possible, which will lead the activeness of constraint (18b), i.e., the holding the following equality

$$\sum_{i=1}^{n-1} \frac{l_i}{f_i} = T_{\text{th}} - \tau_t - \sum_{i=n}^{N} \frac{l_i}{f_e}. \tag{37}$$

As a summary, when $\tau_t = T_{\text{th}} - \sum_{i=1}^{n-1} \frac{l_i}{f_{\max}} - \sum_{i=n}^{N} \frac{l_i}{f_e}$, (37) and the fact that f_i are all equal can lead to the solution $f_i = \frac{\sum_{i=1}^{n-1} l_i}{\left(T_{\text{th}} - \sum_{i=n}^{N} \frac{l_i}{f_e} - \tau_t \right)} = f_{\max}$ for $i \in \{1, 2, ..., n-1\}$, which corresponds to the case that $\mathcal{A}_1(n-1) = \{1, 2, ..., n-1\}$ and $\mathcal{A}_2(n-1) = \emptyset$; when $0 < \tau_t < T_{\text{th}} - \sum_{i=1}^{n-1} \frac{l_i}{f_{\max}} - \sum_{i=n}^{N} \frac{l_i}{f_e}$, (37) and the fact that f_i are all equal can lead to the solution $f_i = \frac{\sum_{i=1}^{n-1} l_i}{\left(T_{\text{th}} - \sum_{i=n}^{N} \frac{l_i}{f_e} - \tau_t \right)}$ for $i \in \{1, 2, ..., n-1\}$,

which corresponds to the case that $\mathcal{A}_2(n-1) = \{1, 2, ..., n-1\}$ and $\mathcal{A}_1(n-1) = \emptyset$.

APPENDIX B
PROOF OF LEMMA 1

We complete this proof via induction method. In the first step, the case that $m = M(n)$ is investigated. When $m = M(n)$, there is one fading block left for task offloading. In this case, there is $Q_{n,M(n)}(d, h; M(n), t) = e(d, h, t)$. For the function $e(d, h, t)$, the first-order derivative of $e(d, h, t)$ with t can be given as

$$\frac{de(d, h, t)}{dt} = \frac{1}{h} \left(e^{\frac{d}{t}} - 1 - \frac{d}{t} \times e^{\frac{d}{t}} \right) \tag{38}$$

Define $x \triangleq d/t \geq 0$, the term $\left(e^{\frac{d}{t}} - 1 - \frac{d}{t} \times e^{\frac{d}{t}} \right)$ in the right-hand side of (38) can be written as $g(x) = e^x - 1 - xe^x$, whose first-order derivative $g'(x) = -xe^x \leq 0$. Then there is $g(x) \leq g(0) = 0$ for $x \geq 0$, which proves that $\frac{de(d,h,t)}{dt} \leq 0$. Hence for $0 \leq t_1 \leq t_2 \leq \tau$, there would be $e(d, h, t_1) \geq e(d, h, t_2)$.

In the second step, we need to prove the holding of $Q_{n,m}(d, h; M(n), t_1) \geq Q_{n,m}(d, h; M(n), t_2)$ given the condition that $Q_{n,m+1}(d, h; M(n), t_1) \geq Q_{n,m+1}(d, h; M(n), t_2)$ for $0 \leq t_1 \leq t_2 \leq \tau$ and any d and h value. Denote d_1^* and d_2^* as the optimal $d_{n,m}$ to achieve $Q_{n,m}(d, h; M(n), t_1)$ and $Q_{n,m}(d, h; M(n), t_2)$ respectively, where the definition of $d_{n,m}$ can be found in (7), then there is

$$\begin{aligned}
& Q_{n,m}(d, h; M(n), t_1) \\
= & e(d_1^*, h, \tau) + Q_{n,m+1}(d - d_1^*; M(n), t_1) \\
\geq & e(d_1^*, h, \tau) + Q_{n,m+1}(d - d_1^*; M(n), t_2) \\
\geq & e(d_2^*, h, \tau) + Q_{n,m+1}(d - d_2^*; M(n), t_2) \\
= & Q_{n,m}(d, h; M(n), t_2).
\end{aligned} \tag{39}$$

The first inequality of (39) comes from the fact that $Q_{n,m+1}(d - d_1^*; M(n), t_1) \geq Q_{n,m+1}(d - d_1^*; M(n), t_2)$ is a given condition. The second inequality holds since d_2^*, rather than d_1^*, is the optimal solution of $d_{n,m}$ to achieve $Q_{n,m}(d, h; M(n), t_2)$.

APPENDIX C
PROOF OF LEMMA 2

We first prove that $Q_{n,1}(d; m-1, \tau) \geq Q_{n,1}(d; m, \tau)$. Suppose d_m^* is the optimal solution of $d_{n,1}$ to achieve $Q_{n,1}(d, h; m, \tau)$, where the definition of $d_{n,1}$ is given in (7). Then there would be

$$\begin{aligned}
& Q_{n,1}(d, h; m, \tau) \\
= & e(d_m^*, h, \tau) + Q_{n,2}(d - d_m^*; m, \tau) \\
\overset{(a)}{\leq} & e(0, h, \tau) + Q_{n,2}(d; m, \tau) \\
= & Q_{n,2}(d; m, \tau) \\
= & Q_{n,1}(d; m-1, \tau)
\end{aligned} \tag{40}$$

where (a) comes from the fact that d_m^* is the optimal solution of d_m for minimizing $e(d_m, h, \tau) + Q_{n,2}(d - d_m; m, \tau)$ and it

would achieve less cost for $d_m = d_m^*$ compared with the case with $d_m = 0$.

Hence according to (40), there is [9]

$$
\begin{aligned}
& Q_{n,1}(d; m, \tau) \\
= & \int_0^\infty Q_{n,1}(d, h_{n,1}; m, \tau) p(h_{n,1}) dh_{n,1} \\
\overset{(b)}{\leq} & \int_0^\infty Q_{n,1}(d; m-1, \tau) p(h_{n,1}) dh_{n,1} \\
= & Q_{n,1}(d; m-1, \tau)
\end{aligned}
\tag{41}
$$

where (b) holds because of (40).

With the conclusion that $Q_{n,1}(d; m, \tau) \leq Q_{n,1}(d; m-1, \tau)$ derived in (41) and it can be further found that

$$
\begin{aligned}
& Q_{n,1}(d, h; m, \tau) \\
= & e(d_m^*, h, \tau) + Q_{n,1}(d - d_m^*; m-1, \tau) \\
\overset{(c)}{\leq} & e(d_{m-1}^*, h, \tau) + Q_{n,1}(d - d_{m-1}^*; m-1, \tau) \\
\overset{(d)}{\leq} & e(d_{m-1}^*, h, \tau) + Q_{n,1}(d - d_{m-1}^*; m-2, \tau) \\
= & Q_{n,1}(d, h; m-1, \tau)
\end{aligned}
\tag{42}
$$

where the holding of (c) is due to a similar reason for (a) in (40), and the holding of (d) comes from (41).

Then with (42), it can be derived that for $m_1 \leq m_2$, there is

$$
\begin{aligned}
Q_{n,1}(d, h; m_1, \tau) & \geq Q_{n,1}(d, h; m_1 + 1, \tau) \\
& \geq \cdots \\
& \geq Q_{n,1}(d, h; m_2, \tau).
\end{aligned}
\tag{43}
$$

APPENDIX D
PROOF OF THEOREM 2

Define $M^*(n)$ and $\tilde{t}_n^*$ as the $M(n)$ and $\tilde{t}_n$ such that the constraint (22a) is active, i.e., $T_3(n, M(n), \tilde{t}_n) = T_{\text{th}}$, respectively. To prove this lemma, we need to prove that

$$
Q_{n,1}(d_n, h; M^*(n), \tilde{t}_n^*) \leq Q_{n,1}(d_n, h; M'(n), \tilde{t}_n')
\tag{44}
$$

for any $M'(n) \in \mathcal{I}$ and $\tilde{t}_n' \in (0, \tau]$ such that

$$
T_3(n, M'(n), \tilde{t}_n') \leq T_{\text{th}}.
\tag{45}
$$

Since $T_3(n, M^*(n), \tilde{t}_n^*) = T_{\text{th}}$ and $T_3(n, M'(n), \tilde{t}_n') \leq T_{\text{th}}$, there is

$$
(M^*(n) - 1)\tau + \tilde{t}_n^* \geq (M'(n) - 1)\tau + \tilde{t}_n'.
\tag{46}
$$

Combing the fact that both $\tilde{t}_n^*$ and $\tilde{t}_n'$ falls into the interval $(0, \tau]$, then it can be concluded that $M^*(n) \geq M'(n)$. Then the discussion can be unfolded from the following two cases: 1). $M'(n) = M^*(n)$. In this case, there is $\tilde{t}_n^* > \tilde{t}_n'$ according to (46). Then according to Lemma 1, there is $Q_{n,1}(d_n, h; M^*(n), \tilde{t}_n^*) \leq Q_{n,1}(d_n, h; M'(n), \tilde{t}_n')$. 2).

$M'(n) < M^*(n)$. In this case, it is hard to say $\tilde{t}_n^* > \tilde{t}_n'$ or $\tilde{t}_n^* \leq \tilde{t}_n'$, but there is $M'(n) \leq (M^*(n) - 1)$. Then there is

$$
\begin{aligned}
& Q_{n,1}(d_n, h; M'(n), \tilde{t}_n') \\
\overset{(a)}{\geq} & Q_{n,1}(d_n, h; M'(n), \tau) \\
\overset{(b)}{\geq} & Q_{n,1}(d_n, h; M^*(n) - 1, \tau) \\
= & Q_{n,1}(d_n, h; M^*(n), 0) \\
\overset{(c)}{\geq} & Q_{n,1}(d_n, h; M^*(n), \tilde{t}_n^*)
\end{aligned}
\tag{47}
$$

where the holding of (a) and (c) can be explained by Lemma 1, and (b) holds according to Lemma 2.

REFERENCES

[1] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, pp. 2961–2991, June 2018.

[2] Z. Zhou, B. Wang, M. Dong, and K. Ota, "Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, pp. 43–57, May 2020.

[3] T. Guo, K. Yu, M. Aloqaily, and S. Wan, "Constructing a prior-dependent graph for data clustering and dimension reduction in the edge of AIoT," *Future Generation Computer Systems*, vol. 128, pp. 381–394, Oct. 2021.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, pp. 2322–2358, Fourthquarter 2017.

[5] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Trans. Veh. Technol.*, vol. 69, pp. 13 784–13 795, Nov. 2020.

[6] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surv. Tutor.*, vol. 19, pp. 1628–1656, Thirdquarter 2017.

[7] J. Xu, K. Ota, M. Dong, and H. Zhou, "MCTS-enhanced hybrid offloading for aerial multi-access edge computing," *IEEE Wireless Commun.*, vol. 28, pp. 82–87, Oct. 2021.

[8] 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, pp. 4268–4282, Oct. 2016.

[9] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, pp. 4188–4200, June 2019.

[10] 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*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[11] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, pp. 5506–5519, Aug. 2018.

[12] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, pp. 1397–1411, Mar. 2017.

[13] S. Mao, N. Zhang, L. Liu, J. Wu, M. Dong, K. Ota, T. Liu, and D. Wu, "Computation rate maximization for intelligent reflecting surface enhanced wireless powered mobile edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 70, pp. 10 820–10 831, Oct. 2021.

[14] W. Huang, K. Ota, M. Dong, T. Wang, S. Zhang, and J. Zhang, "Result return aware offloading scheme in vehicular edge networks for IoT," *Computer Communications*, vol. 164, pp. 201–214, Oct. 2020.

[15] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered digital forensics for edge consumer electronics in 5G HetNets," *IEEE Trans. Consum. Electron.*, early access, Dec., 2020, doi:10.1109/MCE.2020.3047606.

[16] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, pp. 197–387, Jun. 2014.

[17] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, May 2014, pp. 352–357.

---

[9] The inequality in (41) can be also envisioned intuitively: The function $Q_{n,1}(d; m, \tau)$ actually represents the minimum expected energy consumption for offloading d nats over $(m-1)$ fading blocks, which would be surely no larger than $Q_{n,1}(d; m-1, \tau)$, which represents the minimum expected energy consumption for offloading the same amount of data over a shorter time interval.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3150976, IEEE Internet of Things Journal

16

[18] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, pp. 81–93, Jan. 2015.

[19] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, pp. 4569–4581, Sept. 2013.

[20] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Trans. Commun.*, vol. 67, pp. 375–390, Jan. 2019.

[21] B. Liu, C. Liu, and M. Peng, "Resource allocation for energy-efficient MEC in NOMA-enabled massive iot networks," *IEEE J. Sel. Areas Commun.*, vol. 39, pp. 1015–1027, Apr. 2021.

[22] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted Multi-Access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, pp. 12 244–12 258, Dec. 2018.

[23] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, pp. 2450–2463, Mar. 2019.

[24] 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, pp. 4642–4655, Oct. 2018.

[25] Q. Gu, Y. Jian, G. Wang, R. Fan, H. Jiang, and Z. Zhong, "Mobile edge computing via wireless power transfer over multiple fading blocks: An optimal stopping approach," *IEEE Trans. Veh. Technol.*, vol. 69, pp. 10 348–10 361, Sept. 2020.

[26] J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Trans. Wireless Commun.*, vol. 19, pp. 235–250, Jan. 2020.

[27] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, pp. 203–221, Aug. 1996.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, England, UK: Cambridge University Press, 2004.

[29] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge, England, UK: Cambridge University Press, 2007.

[30] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Nashua, NH, USA: Athena Scientific, 1995.

[31] D. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and Optimization*. Nashua, NH, USA: Athena Scientific, 2003.

[32] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, pp. 1757–1771, May 2016.

Han Hu is currently a Professor with the School of Information and Electronics, Beijing Institute of Technology, China. He received the B.E. and Ph.D. degrees from the University of Science and Technology of China, China, in 2007 and 2012 respectively. His research interests include multimedia networking, edge intelligence and space-air-ground integrated network. He received several academic awards, including Best Paper Award of IEEE TCSVT 2019, Best Paper Award of IEEE Multimedia Magazine 2015, Best Paper Award of IEEE Globecom 2013, etc. He served as an Associate Editor of IEEE TMM and Ad Hoc Networks, and a TPC Member of Infocom, ACM MM, AAAI, IJCAI, etc.

Dr. Ning Zhang is an Associate Professor in the Department of Electrical and Computer Engineering at University of Windsor, Canada. He received the Ph.D degree in Electrical and Computer Engineering from University of Waterloo, Canada, in 2015. After that, he was a postdoc research fellow at University of Waterloo and University of Toronto, Canada, respectively. His research interests include connected vehicles, mobile edge computing, wireless networking, and machine learning. He is a Highly Cited Researcher. He serves as an Associate Editor of IEEE Internet of Things Journal, IEEE Transactions on Cognitive Communications and Networking, and IEEE Systems Journal; and a Guest Editor of several international journals, such as IEEE Wireless Communications, IEEE Transactions on Industrial Informatics, IEEE Transactions on Intelligent Transportation Systems, and IEEE Transactions on Cognitive Communications and Networking. He also serves/served as a general chair for IEEE SAGC 2021, TPC chair for IEEE VTC 2021 and IEEE SAGC 2020, a track chair for several international conferences including IEEE ICC 2022, CollaborateCom 2021, IEEE VTC 2020, AICON 2020 and CollaborateCom 2020, and a co-chair for numerous international workshops. He received the IEEE TCSVC Rising Star Award for outstanding contributions to research and practice of MEC and IoT Service, an NSERC PDF award, and 6 Best Paper Awards from IEEE Globecom, IEEE WCSP, IEEE ICC, IEEE ICCC, IEEE Technical Committee on Transmission Access and Optical Systems, and Journal of Communications and Information Networks, respectively.

Xuming An received the B.E. degree in information countermeasure technology and the M.S. degree in information and communication engineering from the Beijing Institute of Technology, Beijing, China, in 2015 and 2019, respectively, where he is currently pursuing a Ph.D. degree in information and communication engineering. His current research interests include machine learning and resource management for mobile edge computing systems.

Rongfei Fan received the B.E. degree in communication engineering from Harbin Institute of Technology, Harbin, China, in 2007, and the Ph. D degree in electrical engineering from the University of Alberta, Edmonton, Alberta, Canada, in 2012. Since 2013, he has been a faculty member at the Beijing Institute of Technology, Beijing, China, where he is currently an Associate Professor in the School of Cyberspace Science and Technology. His research interests include edge computing, federal learning, resource allocation in wireless networks, and statistical signal processing.

Saman Atapattu (Senior Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from the University of Peradeniya, Sri Lanka, the M.Eng. degree in telecommunications from the Asian Institute of Technology (AIT), Thailand, and the Ph.D. degree in electrical engineering from the University of Alberta, Canada. He is currently working with the Department of Electrical and Electronic Engineering, University of Melbourne, Australia. His research interests include wireless communications and signal processing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3150976, IEEE Internet of Things Journal

17

Theodoros A. Tsiftsis (Senior Member, IEE) received the B.Sc. degree in physics from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1993, the M.Sc. degree in digital systems engineering from the Heriot-Watt University, Edinburgh, U.K., in 1995, the M.Sc. degree in decision sciences from the Athens University of Economics and Business, Athens, Greece, in 2000, and the Ph.D. degree in electrical engineering from the University of Patras, Patras, Greece, in 2006. He is currently a Professor with Jinan University, Guangzhou, China, affiliated with the University of Thessaly, Volos, Greece, and also an Honorary Professor with Shandong Jiaotong University, Jinan, China. His research interests include the broad areas of communication theory and wireless communications with an emphasis on energy- efficient communications, smart surfaces, ultra-reliable and low-latency communication, and physical layer security.

Dr. Tsiftsis was the Editorial Boards of IEEE Transactions on Communications, IEEE Transactions on Vehicular Technology, and IEEE Communications Letters. He is currently the Editor of IEEE Transactions on Wireless Communications, and an Associate Editor for IEEE Transactions on Mobile Computing. Prof. Tsiftsis has been appointed as an IEEE Vehicular Technology Society Distinguished Lecturer (IEEE VTS DL), Classes 2018 & 2020.