Optimal Task Scheduling and Resource Allocation for Self-Powered Sensors in Internet of Things: An Energy Efficient Approach

Jiajie Xu, Kaixin Li, Ying Chen, *Senior Member, IEEE*, and Jiwei Huang, *Senior Member, IEEE*

***Abstract*—The prosperous development of the Internet of Things (IoT) and wireless communication technologies has led to explosive growth in the number of IoT sensor devices. However, some sensor devices are inevitably deployed in remote and inaccessible areas. How to continuously and reliably power sensor devices is a critical problem that needs to be addressed. Deploying self-powered modules on sensor devices by adopting self-powered technology is an effective solution to the energy shortage of sensor devices. Besides, Mobile Edge Computing (MEC) as a promising paradigm has provoked widespread popularity. With the help of MEC, devices can offload computing tasks to edge servers for processing, which greatly alleviates the limitations on energy, storage, and computation capability of devices. In this paper, we jointly study task scheduling and resource allocation in the MEC scenario where the sensor devices are with self-powered modules. Our goal is to minimize the long-term average energy consumption of self-powered sensor devices while ensuring system performance. We adopt stochastic optimization techniques to transform the modeled stochastic problem into a deterministic problem. Then, the deterministic problem is decomposed into four sub-problems, and we propose a task scheduling and resource allocation (TSRA) algorithm to solve these problems. Finally, we carry out a series of parameter analysis and comparison experiments to verify the TSRA algorithm. The experimental results show that our TSRA algorithm can make a dynamic tradeoff between energy consumption and system performance. It also demonstrates the effectiveness of our TSRA algorithm compared with other baseline algorithms.**

***Index Terms*—Internet of Things (IoT), self-powered sensors, mobile edge computing (MEC), task scheduling, resource allocation.**

I. Introduction

THE MATURITY of the Internet of Things (IoT) and wireless communication technologies have made an intensive impact on our social development and daily life [1]. IoT devices can be interconnected with one another through wireless communication technology [2]. As an essential part of IoT devices, IoT sensor devices have recently been widely used in smart home, intelligent agriculture, smart healthcare, etc. [3], [4]. However, some sensor devices are inevitably deployed in remote and inaccessible areas, such as marine areas, and remote forests. The traditional power supply methods are complex or unable to reach these places due to realistic conditions [5]. In addition, the battery capacity of these sensor devices is also greatly limited because of their manufacturing cost and physical size [6], [7]. How to deal with the limited energy storage of sensor devices, and thus the barrier to sensor sustainability has attracted the widespread attention of researchers and industrial designers [8], [9]. As an effective way to utilize renewable resources, self-powered technology allows sensor devices to extend their lifespan by harvesting ambient energy (e.g., solar power and wind power) through self-powered modules [10]. In this way, the energy limitation of sensor devices is alleviated [11].

Mobile edge computing (MEC) is a promising paradigm that allows devices to transmit their computing tasks to the base station (BS) for execution [12], [13]. The devices can offload the generated tasks completely or partially to the edge server (ES) which is deployed on the BS typically, and then the ES allocates the computing resources to the devices for calculating the offloaded tasks [14]. Compared with local computing, the ES can allocate the transmission and computing resources to reduce the computing and storage pressure of the devices and greatly decrease the local energy consumption while ensuring the system's performance [15].

In this paper, we jointly study the problem of task scheduling and resource allocation in self-powered sensor systems. For each sensor device, a self-powered module is deployed to capture the green energy from the surrounding environment to extend its life cycle. We model the problem as a stochastic problem. Our optimization goal is to minimize the total long-term average energy consumption of self-powered sensor devices with the constraints of ES computation resources and system performance. The decisions contain the task scheduling and the resource allocation policies on both the local side and the ES side. Then, we adopt a stochastic optimization approach to transform the task scheduling and resource allocation problem into a deterministic problem and decompose it into four sub-problems. Further, we design and propose a task scheduling and resource allocation (TSRA) algorithm to

Manuscript received 29 May 2023; revised 26 February 2024 and 11 June 2024; accepted 16 June 2024. Date of publication 27 June 2024; date of current version 21 August 2024. This work was supported in part by the Beijing Natural Science Foundation under Grant L232050, in part by the Project of Cultivation for Young Top-motch Talents of Beijing Municipal Institutions under Grant BPHR202203225, in part by the Young Elite Scientists Sponsorship Program by BAST under Grant BYESS2023031. The associate editor coordinating the review of this article and approving it for publication was A. Santos. *(Corresponding author: Ying Chen.)*

Jiajie Xu, Kaixin Li, and Ying Chen are with the Computer School, Beijing Information Science and Technology University, Beijing 100101, China (e-mail: chenying@bistu.edu.cn).

Jiwei Huang is with the Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing 102249, China.

Digital Object Identifier 10.1109/TNSM.2024.3420254

solve these sub-problems. The main contributions are given as follows.

- We investigate task scheduling and resource allocation problems in self-powered sensor systems. The system consists of multiple BSs which are installed with ESs to serve the set of associated sensor devices. For each sensor device, a self-powered module is equipped to capture the green energy for alleviating the shackling issue of electricity. The tasks generated by self-powered sensor devices can be transmitted to the corresponding BS for further processing by the ES. Our goal is to minimize the long-term average consumption of self-powered sensor devices while ensuring system performance and satisfying the computation resource constraint. The decision variables contain: 1) tasks generated by the self-powered sensor device are to be placed into the local computation queue or local offloading queue, 2) the local CPU-cycle allocation for processing the tasks in the local computation queue, 3) the transmission power allocation for offloading the tasks in local offloading queue, 4) the ES computation resource allocation for self-powered sensor device.
- We transform the long-term average energy consumption optimization problem to a series time-slotted problem and propose the TSRA algorithm to solve it. Specifically, we convert the initial long-term average energy consumption optimization problem into a deterministic problem with the help of stochastic optimization technology. In this way, the constraint of long-term system performance can be addressed. The problem is then decomposed into four sub-problems, including task scheduling and resource allocation on the self-powered sensor device side and ES side. Furthermore, we design and propose the TSRA algorithm to solve these problems with different approaches. Our TSRA algorithm can adjust the strategy to optimize energy consumption. Finally, we give rigorous mathematical derivation to prove the validity of our proposed TSRA algorithm.
- We conduct extensive parameter experiments and comparison experiments to evaluate the performance of the TSRA algorithm. The parameter experimental results verify that our proposed algorithm can dynamically adjust the task scheduling and resource allocation decisions and make a balance between energy consumption and system performance. The comparison experimental results also demonstrate the effectiveness and superiority of the TSRA algorithm compared to other baseline algorithms.

The structure of this paper is organized as follows. The related work is presented in Section II followed by the system model and problem formulation in Section III. In Section IV, we design the task scheduling and resource allocation algorithm and analyze its performance. We carry out extensive experiments in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Recently, the research on IoT sensor devices has attracted extensive attention from academia and industry [16], [17]. IoT sensor devices have been used in many scenarios such as intelligent factories, security cameras, smart homes, and environmental monitoring. Coulby et al. [18] investigated the IoT-enabled multimodal device for monitoring indoor environmental quality. Then, the variability and validity between the inter-device and standard sensors were established. Al-Hawawreh et al. [19] adopted deep reinforcement learning to make the real-time decisions of data transmission for the sensors network to obtain the minimum energy consumption in the smart city. The experience results verified that the proposed approach can achieve the optimization goal effectively while the data transmission in a lower level. Deng et al. [20] considered solving the problem of real-time demand and energy constraints in the IoT system. The optimization goal was to minimize response time and packet losses of tasks while ensuring energy queue stability. The dynamic parallel computing offloading and energy management algorithm was proposed to address the problem.

The MEC paradigm as a promising method has been developed rapidly [21], [22]. IoT devices can offload the generated tasks to the ES which is typically installed in BS for executing the tasks. Hazra et al. [23] studied the cost-efficient task offloading in industrial sensor networks. The authors designed and proposed a named EaDO framework for minimizing delay and energy consumption. Jiang et al. [24] adopted centralized and distributed algorithms to solve the task offloading problem in the MEC system to satisfy the quality of experience of users. In addition, the authors adopted virtual queues to resolve the long-term energy constraints of MEC servers. Tong et al. [25] investigated the energy-efficient computation offloading in the MEC system. Then, they proposed a called LOECOA algorithm to balance the queue backlog and energy consumption. Chen et al. [26] presented an online and polynomial-time-complexity algorithm to solve the offloading problem in MEC. The proposed algorithm could achieve an arbitrary tradeoff between queue backlog and transmission energy consumption. Hua et al. [27] aimed to maximize the long-term system utility which was composed of throughput and fairness. Then, they proposed an asymptotically optimal offloading algorithm, and the experimental results demonstrated its effectiveness.

Furthermore, the self-powered module can be used to obtain green energy from the surrounding environment, thus extending the life of the sensor devices. Guo et al. [28] aimed to minimize the discard rate and long-term average energy consumption with delay as the constraint. For each mobile device, an energy collection module was deployed to capture energy to maintain computing and offloading requirements. Pan [29] aimed to optimize the energy efficiency and reliability of self-powered IoT devices with several software approaches. Mei et al. [30] focused on the task scheduling and resource allocation problem for a MEC system with multiple energy harvesting devices. Based on the stochastic optimization techniques, the studied problem was transformed and decomposed into two subproblems. The authors proposed a named CTMOA algorithm to solve these subproblems. Munir et al. [31] studied the energy dispatch problem in a self-powered network where the BS acts as a self-powered unit. Based on multi-agent

TABLE I
MAIN COMPARISON OF OUR WORK WITH OTHER FORMER WORKS IN THE FIELD OF COMPUTATION OFFLOADING

Works	Investigated scenario	Optimization goals	Optimization Targets	Considered constraints	Employed techniques
Zhao et al. [25]	Traditional MEC system	• Energy consumption of device and ES • Workload queue's stability of sensor devices	• CPU frequency of devices • Size of offloading tasks	• ES computing resource constraint	• Convex optimization • Minimum weight optimization
Chen et al. [26]	Traditional MEC system	• Energy consumption of devices • Workload queue's stability of devices	• Offloading time of devices	• Transmission resource constraint	• Minimum weight optimization
Hua et al. [27]	Traditional MEC system	• Transmission rate of devices • Workload queue's stability of sensor devices	• Offloading decision of devices • CPU frequency of devices • Transmission power of devices	• Transmission resource constraint	• Integer Programming • Convex optimization
Guo et al. [28]	Self-powered MEC system	• Energy consumption of sensor device • Workload queue's stability of sensor devices and ESs	• Harvested energy • Offloading decision of devices • CPU frequency of devices • Transmission power of devices	• Delay constraint	• Linear optimization • Convex optimization • Newton method
Mei et al. [30]	Self-powered MEC system	• Energy consumption of devices • Workload queue's stability of sensor devices	• Size of processed tasks by devices • Offloading time of devices	• Transmission resource constraint	• Minimum weight optimization • Convex optimization
This work	Self-powered MEC system	• Energy consumption of sensor device • Workload queue's stability of sensor devices and ESs	• Offloading decision of sensor devices • CPU frequency of sensor devices • Transmission power of sensor devices • ES computation resource allocation	• Transmission resource constraint • ES computing resource constraint	• Integer programming • Convex optimization • Minimum weight optimization

meta-reinforcement learning, a semi-distributed data-driven solution was presented to solve the problem.

Different from this investigation, we focus on designing and proposing a task scheduling and resource allocation scheme for self-powered MEC systems in this paper. We combine task scheduling, frequency scaling, transmission power allocation, and the computation resources allocation of edge servers. We aim to minimize the long-term energy consumption of the sensor side while ensuring the system's performance. We consider the self-powered sensor devices for harvesting green energy to decrease energy consumption and extend the device's lifetime. Besides, with the increase in the number of devices, more tasks will be offloaded to the edge server. However, the resources of the server are limited. We consider the condition of limited server resources in this paper. Through reasonable task scheduling and resource allocation strategies, the stability of the system can be guaranteed. The main comparison of our work with other former works in the field of computation offloading are listed in Table I.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. Network Model of Self-Powered Sensors in Internet of Things

We consider a self-powered sensor system that supports the Internet of Things. There are multiple BSs denoted by $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$. For each BS, an ES is deployed to provide computing services to the set of self-powered sensor devices within its coverage area. For simplicity, we adopt the index of BS to the corresponding ES and consider it interchangeable. Let $\mathcal{U}^k = \{1, \ldots, u, \ldots, U^k\}$ represents the set of self-powered sensor devices associated with BS k. Each sensor device has a self-powered model for capturing the energy from the environment and a single antenna is used for communicating with the BS. We study a time slot model with the time set defined as $\mathcal{T} = \{1, \ldots, t, \ldots, T\}$. The main symbols used in this paper are listed in Table II.

B. Task and Queue Model

For each self-powered sensor device, there is a task generated at the beginning of each time slot. Let $J_u(t)$ (in bits) denote the size of the arrived task for self-powered sensor device u in time slot t which is subject to $0 \leq J_u(t) \leq J_u^{\max}$. We establish three queues (the queue length is measured in bits) for each self-powered sensor device. Let $H_u^l(t)$ represent the queue for caching the task that will be processed locally, $H_u^o(t)$ represent the queue for caching the task which will be offloaded to BS, and $H_u^k(t)$ represent the queue of self-powered sensor device u in ES k. We consider that tasks in the three queues adopt the strategy of First-In-First-Out (FIFO). Besides, a binary offloading strategy is adopted for the generated tasks. Let $\varkappa_u(t) \in \{0, 1\}$ be the indicator of offloading policy, where $\varkappa_u(t) = 0$ denotes the task generated by self-powered sensor device u in time slot t is placed into $H_u^l(t)$, and $\varkappa_u(t) = 1$ represents the task is placed into $H_u^o(t)$.

1) Local Computation: For the tasks in $H_u^l(t)$, it is processed by the CPU of the self-powered sensor device. Let $f_u(t)$ represent the CPU-cycle frequency (in Hz) of self-powered sensor device u in time slot t. Note that $0 \leq f_u(t) \leq f_u^{\max}$. Therefore, the number of tasks (in bits) processed locally can be expressed by

$$C_u^l(t) = \frac{f_u(t)\tau}{\delta_u}, \tag{1}$$

where τ denotes the duration (in seconds) of each time slot, and δ_u represents the required CPU cycles for processing one

TABLE II
NOTATIONS

Symbols	Descriptions
$\mathcal{K}$	The set of BSs
$\mathcal{U}^k$	The set of IoT devices associated with BS k
$\mathcal{T}$	The set of time slots
$J_u(t)$	The size (in bits) of the generated task for sensor device u
$H_u^l(t)$	The local computation queue length (in bits) for sensor device u in time slot t
$H_u^o(t)$	The local offloading queue length (in bits) for sensor device u in time slot t
$H_u^k(t)$	The queue length (in bits) in ES k for sensor device u in time slot t
$\varkappa_u(t)$	The offloading decision for sensor device u in time slot t
$f_u(t)$	The CPU-cycle frequency (in Hz) of sensor device u in time slot t
τ	The duration time (in seconds) of each time slot
δ_u	The required CPU cycles for processing one bit of task of sensor device u
$p_u(t)$	The transmission power (in watt) of sensor device u in time slot t
g_u	The channel gain of sensor device u
σ^2	The Gaussian channel noise
$r_u(t)$	Transmission rate (in bits/second) of sensor device u in time slot t
b_u	The bandwidth (in Hz) of BS allocates to sensor device u
$\xi_u(t)$	The proportion of computing resources allocated by ES to sensor device u
f^k	The computing capacity of ES k
ϑ_u	The effective switched capacitor coefficient of sensor device u
$E_u^h(t)$	The arrived green energy of sensor device u in time slot t
$B_u(t)$	The battery capacity of sensor device u

bit of task. Thus, the evolution of $H_u^l(t)$ is formulated by

$$H_u^l(t+1) = \max\Big\{H_u^l(t) - C_u^l(t), 0\Big\} + (1-\varkappa_u(t))J_u(t), \tag{2}$$

where $C_u^l(t)$ is subject to $C_u^l(t) \le H_u^l(t)$.

2) Offloading Transmission: As to $H_u^o(t)$, it maintains the tasks that will be transmitted to BS for further processing by the ES. For each BS k, the channel bandwidth is W_k. We consider that the BSs serve self-powered sensor devices via Frequency Division Multiple Access (FDMA) schemes [32] and there is no interference between different self-powered sensor devices. Hence, the bandwidth of each self-powered sensor device is $b_u = \frac{W_k}{N_k}$, where N_k denotes the number of self-powered sensor devices that are connected to BS k. According to Shannon Theory [33], the transmission rate $r_u(t)$ of self-powered sensor device u in time slot t is

$$r_u(t) = b_u \log_2\left(1 + \frac{p_u(t)g_u}{\sigma^2}\right), \tag{3}$$

where $p_u(t)$ denotes the transmission power (in watt) of self-powered sensor device u in time slot t with the upper bound of $p_u^{\max}$, g_u represents the channel gain between self-powered sensor device u and BS k, and σ^2 denotes the white Gaussian channel noise. The offloading size $C_u^o(t)$ (in bits) of self-powered sensor device u in time slot t is given by

$$C_u^o(t) = r_u(t)\tau. \tag{4}$$

After that, the evolution of $H_u^o(t)$ is expressed by

$$H_u^o(t+1) = \max\{H_u^o(t) - C_u^o(t), 0\} + \varkappa_u(t)J_u(t). \tag{5}$$

Note that the offloading size should not exceed the queue backlog of $H_u^o(t)$, i.e., $C_u^o(t) \le H_u^o(t)$.

3) ES Computation: After the tasks are transmitted to BS, they will be placed into the corresponding queue in the ES. And ES needs to allocate computing resources for $H_u^k(t)$. Let $\xi_u(t)$ represent the proportion of computing resources allocated to $H_u^k(t)$ by ES k. Therefore, the number (in bits) of processing tasks is defined as

$$C_u^k(t) = \frac{\xi_u(t) f^k \tau}{\delta_u}, \tag{6}$$

where f^k represents the CPU-cycle frequency (in Hz) of ES k. For each ES and the corresponding collection of self-powered sensor devices, the constraints of $\sum_{u=1}^{U^k} \xi_u(t) \le 1$ and $\frac{\xi_u(t)\tau f^k}{\delta_u} \le H_u^k(t)$ are needed to meet. Further, we can obtain the evolution of $H_u^k(t)$, which is expressed by

$$H_u^k(t+1) = \max\Big\{H_u^k(t) - C_u^k(t), 0\Big\} + C_u^o(t), \tag{7}$$

where $C_u^k(t)$ is subject to $C_u^k(t) \le H_u^k(t)$.

C. Self-Powered and Consumption Model

In this work, we consider that the energy of the sensor device comes from two parts, including the energy of the traditional battery and the harvesting energy. For the task offloading, we consider that the sensor device uses the energy in the traditional battery directly, and the harvesting energy is only used for local computation requirements. When the harvesting energy can not satisfy local computing, the traditional energy can be used as a supplement. Besides, due to the ES having a sufficient energy supply, we only concentrate on the energy consumption of self-powered sensor devices. Therefore, the energy consumption consists of two parts, including the computation and offloading of the self-powered sensor devices locally. Let $E_u^l(t)$ represent the energy consumption (in J) of self-powered sensor device u in time slot t. According to [34], $E_u^l(t)$ can be expressed by

$$E_u^l(t) = \vartheta_u f_u(t)^3 \tau, \tag{8}$$

where ϑ_u represents the effective switched capacitor coefficient, which depends on the chip structure of the CPU.

For each self-powered sensor device, a self-powered module is installed to obtain green energy from the environment, such as solar energy and wind energy to extend life. We consider that the discharging and charging process of each self-powered sensor device can be executed simultaneously. Let $E_u^h(t)$ represent the arrived green energy of self-powered sensor device u in time slot t, which meets the relationship of $E_u^h(t) \le E_u^{h,\max}$. Thus, the evolution of battery capacity $B_u(t)$ can be expressed by

$$B_u(t+1) = \max\Big\{B_u(t) - E_u^l(t), 0\Big\} + E_u^h(t). \tag{9}$$

Note that $B_u(t)$ should satisfy $B_u^{\min} \le B_u(t) \le \infty$, where $B_u^{\min}$ represents the minimum capacity of energy battery.

For each self-powered sensor device, the captured green energy is used to process the tasks first. Thus, the final energy consumption $E'_u(t)$ of self-powered sensor device u in time slot t for processing task is defined as

$$E'_u(t) = \max\Big\{E^l_u(t) - B_u(t), 0\Big\}. \tag{10}$$

Regarding the energy consumption of task offloading for self-powered sensor device u in time slot t, it can be formulated by

$$E^o_u(t) = p_u(t)\tau. \tag{11}$$

According to the energy consumption of local computation and offloading transmission, we can obtain the total energy consumption of self-powered sensor device u in time slot t. The total energy consumption is represented as

$$E_u(t) = E'_u(t) + E^o_u(t). \tag{12}$$

Then, the total energy consumption is expressed by

$$E(t) = \sum_{k=1}^{K}\sum_{u=1}^{U^k} E_u(t). \tag{13}$$

D. Problem Formulation

While the task arriving process and obtained green energy in each time slot are stochastic and hard to predict, it is significant to formulate a suitable strategy for task scheduling and resource allocation. In this paper, our target is to minimize the long-term average energy consumption of self-powered sensor devices which is subject to the system performance. For each self-powered sensor device u in time slot t, the decisions contain whether the arrived task $J_u(t)$ to be placed into $H^l_u(t)$ or $H^o_u(t)$, the local CPU-cycle frequency $f^l_u(t)$ and transmission power $p_u(t)$ allocation. For each ES k in time slot t, its computation resources should also be allocated to the set of sensor devices that connect to it. For self-powered sensor device u, the proportion of computing resources allocated by ES k to self-powered sensor device u is $\xi_u(t)$. Let $\Upsilon(t) = \{\varkappa(t), f(t), p(t), \xi(t)\}$ be the decision parameter vector in time slot t. The optimization problem can be expressed by

$$\mathcal{P}_1: \quad \min_{\Upsilon(t)} \bar{E} = \lim_{T\to\infty} \frac{\sum_{t=0}^{T-1}\mathbb{E}\{E(t)\}}{T}, \tag{14}$$

$$\begin{aligned}
s.t. \quad & C1: \varkappa_u(t) \in \{0,1\}, \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C2: 0 \le f_u(t) \le f^{\max}_u, \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C3: 0 \le p_u(t) \le p^{\max}_u, \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C4: \sum_{u=1}^{U^k} \xi_u(t) \le 1, \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C5: \frac{\tau f_u(t)}{\delta_u} \le H^l_u(t), \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C6: C^o_u(t) \le H^o_u(t), \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C7: \frac{\xi_u(t)\tau f_m}{\delta_u} \le H^k_u(t), \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C8: \lim_{t\to\infty} \frac{\mathbb{E}\Big\{H^l_u(t)\Big\}}{t} = 0, \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C9: \lim_{t\to\infty} \frac{\mathbb{E}\{H^o_u(t)\}}{t} = 0, \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K},\\
& C10: \lim_{t\to\infty} \frac{\mathbb{E}\Big\{H^k_u(t)\Big\}}{t} = 0, \forall u \in \mathcal{U}^k, \forall k \in \mathcal{K}.
\end{aligned}$$

$C1$ is the constraint of task scheduling which represents the arrived task that can be placed into $H^l_u(t)$ or $H^o_u(t)$. $C2$ and $C3$ are the constraints of local CPU-cycle frequency and transmission power where $f^{\max}_u$ and $p^{\max}_u$ are the upper bounds of the two variables, respectively. $C3$ denotes the total time ratio of the allocated computation resources should be less 1. $C5{\sim}C7$ represent the size of processing or offloading tasks that can not exceed the queue backlog. $C8{\sim}C10$ are the constraints of long time queue length, which guarantee the queue backlog should not be longer all the time.

Note that $\mathcal{P}_1$ is a mixed-integer optimization problem typically that jointly optimizes the task scheduling and resource allocation. Besides, it is also a stochastic problem because of the generation of tasks and the green energy arrival process are random. By this, the Lyapunov optimization technique is considered a suitable and feasible approach for solving this problem. Thus, we adopt this optimization framework to transform and address this stochastic problem in the following section.

IV. Task Scheduling and Resource Allocation Algorithm Design for Self-Powered Sensors in Internet of Things

A. Energy Efficient Task Scheduling and Resource Allocation Problem Reformulation

We define $\Omega(t) = \{H^l_u(t), H^o_u(t), H^k_u(t)\}$ as vector the queue backlog in time slot t. The Lyapunov function $\Gamma(\Omega(t))$ is defined as

$$\Gamma(\Omega(t)) = \frac{1}{2}\sum_{k=1}^{K}\sum_{u=1}^{U^k}\Big[H^l_u(t)^2 + H^o_u(t)^2 + H^k_u(t)^2\Big]. \tag{15}$$

By subtracting the Lyapunov function $\Gamma(\Omega(t+1))$ in time slot $t+1$ from the Lyapunov function $\Gamma(\Omega(t))$ in time slot t, we can obtain the Lyapunov drift function $\triangle\Gamma(\Omega(t))$, which is defined as

$$\triangle\Gamma(\Omega(t)) = \mathbb{E}\{\Gamma(\Omega(t+1)) - \Gamma(\Omega(t))|\Omega(t)\}. \tag{16}$$

Then, the drift-plus-penalty function $\triangle_V(\Omega(t))$ can be expressed by

$$\triangle_V(\Omega(t)) = \triangle\Gamma(\Omega(t)) + V\mathbb{E}\{E(t)|\Omega(t)\}. \tag{17}$$

Note that $V \ge 0$, and V is a parameter for the tradeoff of the queue backlog and energy consumption. A larger V means that we focus more on energy consumption.

For each time slot t, minimizing $\triangle_V(\Omega(t))$ is our target. However, it is hard to optimize $\triangle_V(\Omega(t))$ directly. Therefore, we seek to optimize the upper bound of (17), which is given in Theorem 1.

Theorem 1: The upper bound of $\Delta\Gamma(\Omega(t)) + V\mathbb{E}\{E(t)|\Omega(t)\}$ is

$$\triangle_V(\Omega(t)) = \triangle\Gamma(\Omega(t)) + V\mathbb{E}\{E(t)|\Omega(t)\} \le$$

$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^l(t)\Big[(1-\varkappa_u(t))J_u(t)-C_u^l(t)|\Omega(t)\Big]\right\}$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^o(t)[\varkappa_u(t)J_u(t)-C_u^o(t)|\Omega(t)]\right\}$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_k(t)\Big[C_u^o(t)-C_u^k(t)|\Omega(t)\Big]\right\}$$
$$+V\mathbb{E}\{E(t)|\Omega(t)\}+M, \tag{18}$$

where $M=\max\{\sum_{k=1}^{K}\sum_{u=1}^{U^k}\{\frac{1}{2}\mathbb{E}\{[(1-\varkappa_u(t))J_u(t)]^2+H_u^l(t)^2\}+\frac{1}{2}\mathbb{E}\{[\varkappa_u(t)J_u(t)]^2+H_u^o(t)^2\}+\frac{1}{2}\mathbb{E}\{[H_u^o(t)]^2+H_u^k(t)^2\}\}\}$ and M is a constant.

Proof: According to $[\max\{x-y,0\}+z]^2\le x^2+y^2+z^2+2x(z-y)$, the following inequality equations can be obtained.

$$H_u^l(t+1)^2\le H_u^l(t)^2+[(1-\varkappa_u(t))J_u(t)]^2+H_u^l(t)^2+2H_u^l(t)\Big[(1-\varkappa_u(t))J_u(t)-C_u^l(t)\Big]. \tag{19}$$
$$H_u^o(t+1)^2\le H_u^o(t)^2+[\varkappa_u(t)J_u(t)]^2+H_u^o(t)^2+2H_u^o(t)[\varkappa_u(t)J_u(t)-C_u^o(t)]. \tag{20}$$
$$H_u^k(t+1)^2\le H_u^k(t)^2+C_u^o(t)^2+C_u^k(t)^2+2H_u^k(t)\Big[C_u^o(t)-C_u^k(t)\Big]. \tag{21}$$

After that, we add the above three inequality Equations into Equation (17) and we have

$$\triangle_V(\Omega(t))=\triangle\Gamma(\Omega(t))+V\mathbb{E}\{E(t)|\Omega(t)\}\le$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^l(t)\Big[(1-\varkappa_u(t))J_u(t)-C_u^l(t)|\Omega(t)\Big]\right\}$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^o(t)[\varkappa_u(t)J_u(t)-C_u^o(t)|\Omega(t)]\right\}$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_k(t)\Big[C_u^o(t)-C_u^k(t)|\Omega(t)\Big]\right\}$$
$$+V\mathbb{E}\{E(t)|\Omega(t)\}+M. \tag{22}$$

■

Through adopting the Lyapunov framework, we can transform the optimization problem from $\mathcal{P}_1$ to a series slot optimization problem. For each time slot, our goal is to optimize the right side of Equation (18). Further, because *M* is constant, the initial problem can be reformulated to $\mathcal{P}_2$ which can be expressed by

$$\mathcal{P}_2:\quad \min_{\Upsilon(t)} V\mathbb{E}\{E(t)|\Omega(t)\}$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^l(t)\Big[(1-\varkappa_u(t))J_u(t)-C_u^l(t)\Big]|\Omega(t)\right\}$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^o(t)\big[\varkappa_u(t)J_u(t)-C_u^o(t)\big]|\Omega(t)\right\}$$
$$+\mathbb{E}\left\{\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^k(t)\Big[C_u^o(t)-C_u^k(t)\Big]|\Omega(t)\right\}, \tag{23}$$
$$s.t.\quad C1:\varkappa_u(t)\in\{0,1\},\forall u\in\mathcal{U}^k,\forall k\in\mathcal{K},$$
$$C2:0\le f_u(t)\le f_u^{\max},\forall u\in\mathcal{U}^k,\forall k\in\mathcal{K},$$
$$C3:0\le p_u(t)\le p_u^{\max},\forall u\in\mathcal{U}^k,\forall k\in\mathcal{K},$$
$$C4:\sum_{u=1}^{U^k}\xi_u(t)\le 1,\forall u\in\mathcal{U}^k,\forall k\in\mathcal{K},$$
$$C5:\frac{\tau f_u(t)}{\delta_u}\le H_u^l(t),\forall u\in\mathcal{U}^k,\forall k\in\mathcal{K},$$
$$C6:C_u^o(t)\le H_u^o(t),\forall u\in\mathcal{U}^k,\forall k\in\mathcal{K}.$$

B. Task Scheduling and Resource Allocation Algorithm for Self-Powered Sensors

In this section, we design and propose a task scheduling and resource allocation algorithm to address the problem $\mathcal{P}_2$. In detail, the problem is divided into four sub-problems in each time slot to solve which include 1) task scheduling decision, 2) local CPU-cycle frequency allocation, 3) transmission power allocation, 4) ES computation resources allocation. We decoupled the decision variables $\varkappa(t)$, $f(t)$, $p(t)$ and $\xi(t)$ from the problem $\mathcal{P}_2$ for easier handling.

1) Task Scheduling Decision: By extracting all the items in $\mathcal{P}_2$ including $\varkappa_u(t)$, we can obtain the sub-problem $\mathcal{P}_{2-1}$ which is expressed by

$$\mathcal{P}_{2-1}:\quad \min_{\varkappa(t)}\sum_{k=1}^{K}\sum_{u=1}^{U^k}\Big\{H_u^l(t)(1-\varkappa_u(t))J_u(t)+H_u^o(t)\varkappa_u(t)J_u(t)\Big\},$$
$$s.t.\quad \varkappa_u(t)\in\{0,1\}. \tag{24}$$

By extracting all the terms in $\mathcal{P}_2$ including $\varkappa_u(t)$, we can obtain the sub-problem $\mathcal{P}_{2-1}$ which is expressed by

$$\varkappa_u^*(t)=\begin{cases}0, & H_u^o(t)\ge H_u^l(t),\\ 1, & else.\end{cases} \tag{25}$$

Through the above solution of the task scheduling sub-problem, if the local offloading queue $H_u^o(t)$ is longer than the local computation queue $H_u^l(t)$, the task will be placed in $H_u^l(t)$. Otherwise, the task will be placed in $H_u^o(t)$. Thus, we can conclude that the generated task will be placed into a shorter queue for each sensor device locally to maintain the queue stability.

2) Local CPU-Cycle Frequency Allocation: The optimal local CPU-cycle frequency allocation can be solved $\mathcal{P}_{2-2}$ to obtain. The allocation of frequency needs to satisfy the constraints of maximum CPU-cycle frequency and the total computation task size. Thus, the problem can be expressed by

$$\mathcal{P}_{2-2}:\quad \min_{f(t)}\sum_{k=1}^{K}\sum_{u=1}^{U^k}\Big\{-H_u^l(t)\frac{f_u(t)\tau}{\delta_u}+V\max\Big\{\vartheta_u f_u(t)^3\tau-B_u(t),0\Big\}\Big\},$$
$$s.t.\quad 0\le f_u\ (t)\le f_u^{\max},$$

$$\frac{f_u(t)\tau}{\delta_u} \leq H_u^l(t). \tag{26}$$

It is easy to see that each self-powered sensor device is mutually independent of the decision $f_u(t)$. For $\frac{f_u(t)\tau}{\delta_u} \leq H_u^l(t)$, we can obtain $f_u(t) \leq \frac{H_u^l(t)\delta_u}{\tau}$. Therefore, the definition domain of $J(f_u(t))$ is $f_u(t) \in [0, \min\{f_u^{\max}, \frac{H_u^l(t)\delta_u}{\tau}\}]$. Next, we solve the above function $J(f_u(t))$ according to the mathematical part $\max\{\vartheta_u f_u(t)^3\tau - B_u(t), 0\}$. Let $\vartheta_u f_u(t)^3\tau - B_u(t) = 0$, we have $f_u(t) = \sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}}$.

When $\sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}} \leq \min\{f_u^{\max}, \frac{H_u^l(t)\delta_u}{\tau}\}$, the function $J(f_u(t))$ is transformed to a segmentation function, which is given by

$$J(f_u(t)) = \begin{cases} -H_u^l(t)\frac{f_u(t)\tau}{\delta_u}, & f_u(t) \leq \sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}}, \\ -H_u^l(t)\frac{f_u(t)\tau}{\delta_u} + V\Big[\vartheta_u f_u(t)^3\tau - B_u(t)\Big], & else. \end{cases} \tag{27}$$

Our goal is to minimize the function of $J(f_u(t))$, which is a convex function. Thus, the optimal solution $f_u^*(t)$ of local CPU-cycle frequency allocation for self-powered sensor device u in time slot t is

$$f_u^*(t) = \begin{cases} \sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}}, \sqrt{\frac{H_u^l(t)}{3V\vartheta_u\delta_u}} \leq \sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}}, \\ \sqrt{\frac{H_u^l(t)}{3V\vartheta_u\delta_u}}, \sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}} \leq \sqrt{\frac{H_u^l(t)}{3V\vartheta_u\delta_u}} \leq \min\Big\{f_u^{\max}, \frac{f_u(t)\tau}{\delta_u}\Big\}, \\ \min\Big\{f_u^{\max}, \frac{f_u(t)\tau}{\delta_u}\Big\}, \sqrt{\frac{H_u^l(t)}{3V\vartheta_u\delta_u}} \geq \min\Big\{f_u^{\max}, \frac{f_u(t)\tau}{\delta_u}\Big\}. \end{cases} \tag{28}$$

When $\sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}} > \min\{f_u^{\max}, \frac{H_u^l(t)\delta_u}{\tau}\}$, we only need to consider one case, which is $E_u^l(t) \leq B_u(t)$. The function of $J(f_u(t))$ can be given by

$$J(f_u(t)) = -H_u^l(t)\frac{f_u(t)\tau}{\delta_u}. \tag{29}$$

Clearly, the optimal solution of local CPU-cycle frequency for self-powered sensor device u in time slot t is

$$f_u^*(t) = \begin{cases} \frac{H_u^l(t)\tau}{\delta_u}, \frac{H_u^l(t)\tau}{\delta_u} \leq f_u^{\max}, \\ f_u^{\max}, \quad else. \end{cases} \tag{30}$$

For the local CPU-cycle frequency allocation, it is calculated according to whether the harvesting energy in the battery can meet the requirement for the tasks computation in the local computation queue, and it is also subject to the maximum CPU-cycle frequency of the sensor device. When the harvesting energy stored in the battery is not enough to meet the computation requirement, that is, $\sqrt[3]{\frac{B_u(t)}{\vartheta_u\tau}} \leq \min\{f_u^{\max}, \frac{H_u^l(t)\delta_u}{\tau}\}$, the problem $\mathcal{P}_{2-2}$ is converted to a segmentation function $J(f_u(t))$ shown in Equation (27). By solving the convex segmentation function, we can get its optimal solution as shown in Equation (28). When the harvesting energy can meet the computation requirement of all tasks in the local computation queue, the problem $\mathcal{P}_{2-2}$ is transformed into Equation (29), which is a linear programming problem. By solving the linear programming problem, we obtain the optimal solution in this case, as shown in Equation (30).

3) Transmission Power Allocation: $\mathcal{P}_{2-3}$ represents the problem of transmission power allocation.

$$\begin{aligned} \mathcal{P}_{2-3}: \min_{p(t)} \quad & \Bigg[-\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^o(t) b_u \log_2\Big(1 + \frac{p_u(t)g_u}{\sigma^2}\Big)\tau + \sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^k(t) \\ & \times b_u \log_2\Big(1 + \frac{p_u(t)g_u}{\sigma^2}\Big)\tau + V\sum_{k=1}^{K}\sum_{u=1}^{U^k} p_u(t)\tau\Bigg], \\ s.t \quad & 0 \leq p_u(t) \leq p_u^{\max}, \\ & b_u \log_2\Big(1 + \frac{p_u(t)g_u}{\sigma^2}\Big)\tau \leq H_u^o(t). \end{aligned} \tag{31}$$

We can also find that decision of the transmission power allocation for each self-powered sensor device u in time slot t are decoupled. Let $G(p_u(t))$ be the optimization function and our goal is to minimize it. The function $G(p_u(t))$ is expressed by

$$\begin{aligned} G(p_u(t)) = & -H_u^o(t) b_u \log_2\Big(1 + \frac{p_u(t)g_u}{\sigma^2}\Big)\tau \\ & + H_u^k(t) b_u \log_2\Big(1 + \frac{p_u(t)g_u}{\sigma^2}\Big)\tau + V p_u(t)\tau. \end{aligned} \tag{32}$$

Then, we can obtain the first and second derivative of $G(p_u(t))$, Which is given by

$$\frac{dG(p_u(t))}{dp_u(t)} = V\tau + \frac{-\Big[-H_u^k(t) + H_u^o(t)\Big] b_u \frac{g_u}{\sigma^2}\tau}{\Big[1 + p_u(t)\frac{g_u}{\sigma^2}\Big]\ln 2}, \tag{33}$$

$$\frac{d^2G(p_t)}{d^2p_u(t)} = \frac{\Big[-H_u^k(t) + H_u^o(t)\Big] b_u \frac{g_u}{\sigma^2}^2\tau}{\Big[1 + p_u(t)\frac{g_u}{\sigma^2}\Big]^2 \ln 2}. \tag{34}$$

When $H_u^o(t) > H_u^k(t)$, we have $\frac{d^2G(p_u(t))}{d^2p_u(t)} \geq 0$. For $\frac{G(p_u(t))}{dp_u(t)} = 0$, we can get $p_u(t) = \frac{[-H_u^k(t) + H_u^o(t)]b_u}{V\ln 2} - \frac{\sigma^2}{g_u}$. Therefore, the optimal transmission power for self-powered sensor device u in time slot t is

$$p_u^*(t) = \begin{cases} 0, V \geq \frac{\Big[-H_u^k(t) + H_u^o(t)\Big] b_u \frac{g_u}{\sigma^2}}{\ln 2}, \\ \min\Bigg\{p_u^{\max}, \frac{\Big[-H_u^k(t) + H_u^o(t)\Big] b_u}{V\ln 2} - \frac{\sigma^2}{g_u}, \frac{\Big(2^{\frac{H_u^o(t)}{b_u\tau}} - 1\Big)\sigma^2}{g_u}\Bigg\}, \quad else, \end{cases} \tag{35}$$

where $\frac{(2^{\frac{H_u^o(t)}{b_u\tau}} - 1)\sigma^2}{g_u}$ is converted through the constrain of $b_u \log_2(1 + \frac{p_u(t)g_u}{\sigma^2})\tau \leq H_u^o(t)$.

When $H_u^o(t) \leq H_u^k(t)$, $\frac{d^2G(p_u(t))}{d^2p_u(t)} \leq 0$ and $\frac{dG(p_u(t))}{dp_u(t)} \geq 0$ can be obtained. So, we can conclude that function $G(p_u(t))$ is a monotonically increasing function. Therefore, when $p_u^*(t) = 0$, the minimum value of $G(p_u(t))$ can be obtained in this case.

As for the transmission power allocation, it is corresponding to the local offloading queue and the queue in the edge server. For self-powered sensor device u, when the local offloading queue is longer than the queue in the server, that is $H_u^o(t) > H_u^k(t)$, the sub-problem is a convex problem. Further, through Equation (35), the optimal transmission

power can be obtained. Otherwise, when $H_u^o(t) < H_u^k(t)$, the optimization function of this sub-problem is monotonically increasing in the definition domain. Thus for minimizing the optimization goal, we set the optimal transmission power as 0.

4) ES Computation Resources Allocation: By extracting the items which includes $\xi_u(t)$ from the $\mathcal{P}_2$, we can get the following sub-problem $\mathcal{P}_{2-4}$.

$$\begin{aligned}\mathcal{P}_{2-4}: \quad \min_{\xi(t)} & -\sum_{k=1}^{K}\sum_{u=1}^{U^k} H_u^k(t)\frac{f^k(t)\tau\xi_u(t)}{\delta_u}, \\ s.t. \quad & \sum_{u=1}^{U^k} \xi_u^k(t) \le 1, \\ & \frac{f^k(t)\tau\xi_u(t)}{\delta_u} \le H_u^k(t). \end{aligned} \tag{36}$$

For each ES k, the decisions of computation resource allocation are decoupled. Thus, the problem can be transformed into

$$\mathcal{P}'_{2-4}: \quad \min_{\xi_u(t)} -\sum_{u=1}^{U^k} H_u^k(t)\frac{f^k(t)\tau\xi_u(t)}{\delta_u}. \tag{37}$$

This problem is considered a knapsack problem. The greedy algorithm is regarded as an efficient method to solve this problem [25], [35]. Hence, we introduced $S_u(t) = -H_u^k(t)\frac{f^k(t)\tau}{\delta_u}$ as the weight of each item and the initial capacity Z^k of knapsack is 1. We need to take the minimum $S_u(t)$ in order from the set U^k into the knapsack until it is filled. The solution of the ES computation resource allocation is

$$\xi_u^*(t) = \begin{cases} \min\{\frac{H_u^k(t)\delta_u}{\tau f^k}, Z^k\}, & if \quad S_u(t) < 0, u = u^*, \\ 0, & if \quad S_u(t) \ge 0. \end{cases} \tag{38}$$

Note that $u^* = \arg\min_{u\in U^k} S_u(t)$ and Z^k is update with $Z^k = Z^k - \xi_u^*(t)$ dynamically. After self-powered sensor device u is allocated by ES k in time slot t, u^* will be selected again from the set of U^k which does not contain the self-powered sensor devices that have been allocated.

From Equation (37), we can conclude that for each edge server, the edge server will give priority to allocating computing resources to sensor devices with longer queue length. This policy can ensure the stability of the queue.

After solving the four sub-problems, we can obtain the value of task scheduling decision, local CPU-cycle frequency allocation, transmission power allocation, and ES computation resource allocation for each self-powered sensor device. Then, we conduct the corresponding actions and the state of the queues will be updated. Algorithm 1 shows the process of the TSRA algorithm.

Algorithm 1 Task Scheduling and Resource Allocation (TSRA) Algorithm

Input: $H_u^l(t)$, $H_u^o(t)$, $H_u^k(t)$, $J_u(t)$.
Output: $\Upsilon_u(t) = \{\varkappa_u(t), f_u(t), p_u(t), \xi_u(t)\}$.

for all $k \in \mathcal{K}$ **do**
  for all $u \in \mathcal{U}^k$ **do**
    Obtaining the optimal task scheduling strategy $\varkappa_u^*(t)$ through the Equation (25).
    Determining the optimal CPU-cycle frequency $f_u^*(t)$ adopting the solution of $\mathcal{P}_2$ in Equation (28) or (30).
    Calculating the optimal transmission power allocation through solving the problem of $\mathcal{P}_3$.
  end for
end for
for $k \in \mathcal{K}$ **do**
  Initialize the available computation resources $Z^k = 1$.
  Sort $S_u(t)$ of the set of self-powered sensor device $\mathcal{N}^k$ from low to high.
  for $u \in \mathcal{U}^k$ **do**
    Allocating the ES computation resources $\xi_u(t)^*$ to self-powered sensor device u according to (38) with the sorted order.
    Updating Z^k according to $Z^k = Z^k - \xi_u^*(t)$.
  end for
end for

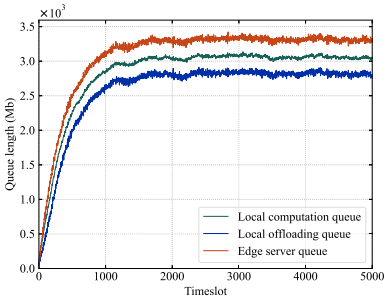


Fig. 1. Queue backlog stability experiment.

V. Performance Evaluation

A. Experiment Settings

The experiments are conducted on a computing platform equipped with an AMD R9 7945HX @ 2.5 GHz CPU and 16 GB of RAM. We implement our experiments using VScode 1.8 based on Python 3.9. In the experiment, we consider 100 sensor devices and 5 BSs for experimental simulation. The total channel bandwidth B^k is 10 MHz and the CPU-cycle frequency f^k is 15 GHz. For each self-powered sensor device, the maximum CPU-cycle frequency $f_u^{\max}$ is 1 GHz and the maximum transmission power $p_u^{\max}$ is 0.1 w [25]. The green energy for each self-powered sensor device is $E_u^h \sim U[0, E_u^{h,\max}]$. The task arrival process is obeying $J_u \sim [0, 2.4]$ Mb. Besides, the CPU cycles required to process a bit task is 1000 cycles/bit and the energy consumption coefficient is 10^{-27} [35]. The Gaussian white noise is 9×10^{-14} and the duration time of each time slot is 1s.

B. System Stability Evaluations

Fig. 1 shows the evolution of the queue backlog of the local computation queue, local offloading queue, and edge server queue. We can clearly find the queue backlog. It verifies their queue backlog updates are stable. The results

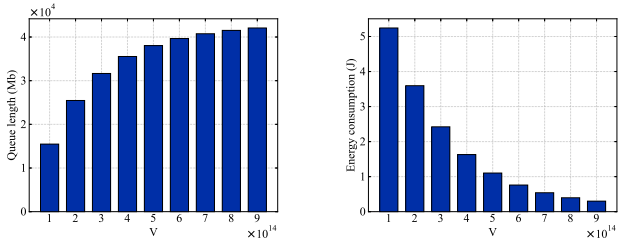


Fig. 2. Tradeoff parameter V vs. queue length and energy consumption.

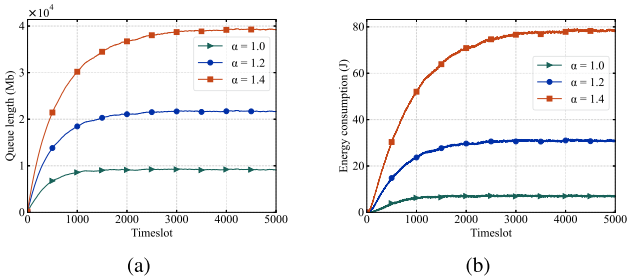


Fig. 3. Task arrival rate vs. queue length and energy consumption.

show that the backlog of self-powered sensor device queues increases rapidly and finally stabilizes over time. Compared with the curve of the local offloading queue, we can find that the curve of the local computing queue is higher. This is because offloading tasks with the same amount of tasks consumes much less energy than local computing. This meets Equation (25) because tasks will be placed on queues with less queue backlog, that is, more tasks will be offloaded. In addition, it can be seen from the curve of the ES server queue that its queue eventually tends to be stable, which confirms that our the TSRA algorithm has good performance in ensuring system performance.

C. Parameter Analysis

1) Impact of Parameter V: As shown in Fig. 2(a), it demonstrates the tradeoff between the total average queue backlog and the parameter V. The total queue length rises with the increase of tradeoff parameter V. The reason is that a larger V makes the system tend to emphasize the penalty function which focuses on energy consumption. Similarly to Fig. 2(a), Fig. 2(b) reveals the total average energy consumption changes with the increase of parameter V. We can conclude that a higher V will lead to lower energy consumption. Combining Fig. 2(a) and Fig. 2(b), we can obtain that the queue backlog and energy consumption can be arbitrarily balanced by changing V.

2) Impact of Task Arrival Rate: Fig. 3(a) and Fig. 3(b) draw the evolution of average queue backlog and energy consumption with different task arrive rates, where the task arrive rate is defined as $J_u \sim [0, \alpha \times J_u^{\max}]$. We pick the value of α as 1, 1.2 and 1.4 respectively. It is not difficult to find that the total queue backlog and energy consumption increase with the rise in task arrival rate. The reason for the phenomenon is that the total computing capacity and transmission capacity

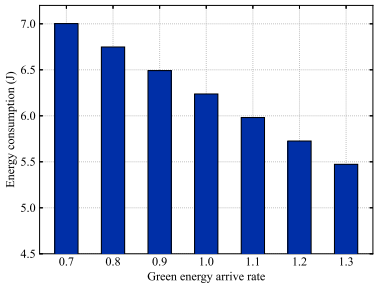


Fig. 4. Green energy harvesting rate vs. Energy consumption.

have not changed, and a larger task arrival rate will make more tasks enter the system, which will lead to a larger queue backlog. However, in order to stabilize the queue, more tasks will be calculated, which will lead to increased energy consumption. Furthermore, whether the value of α is 1, 1.2, or 1.4, the queue backlog and energy consumption can converge quickly and remain stable.

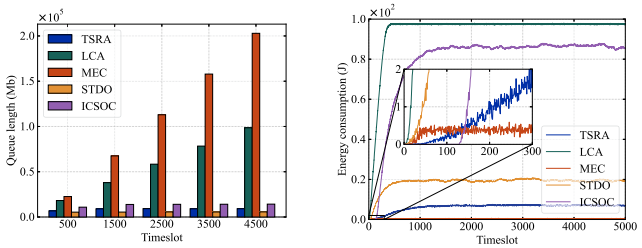
3) Impact of Green Energy Harvesting Rate: Fig. 4 shows the energy consumption with different green energy harvesting rates. The green energy harvesting rate is expressed by $E_u^h \sim [0, \beta \times E_u^{h\max}]$. We let the value of β be 0.7, 0.8, 0.9, 1.0, 1.1, 1.2 and 1.3 respectively. It is easy to observe that energy consumption decreases with the increase of β. This is because a higher green energy harvesting rate means more green energy will be used for processing the generated task. By this, the energy consumption from the traditional battery can be reduced.

D. Comparison Experiments

Our TSRA algorithm is compared to the other four algorithms for verifying the performance of both queue backlog and energy consumption.

- *LCA* algorithm: The tasks generated by sensor devices are processed locally, which means the arrived tasks are placed into the local computation queue.
- *MEC* algorithm: The tasks generated by sensor devices are offloaded to the corresponding BS. It represents the arrived tasks that are put into the local offloading queue.
- *SDTO* algorithm: The SDTO algorithm is inspired by the offloading strategy in [36] while the other decisions same as our proposed algorithm.
- *ICSOC* algorithm: This algorithm is extended from [37]. Each self-powered sensor device adopts the greedy policy when making offloading decisions.

As shown in Fig. 5(a), it depicts the state of queue backlog in the different time slots from 500 to 4500 with the step of 1000. We can see that the queue length of the LCA and MEC algorithms is growing with the time slot all the time. However, our proposed TSRA, STDO, and ICSOC algorithms can maintain the queue stability after a certain time slot. The reason why the LCA algorithm cannot keep the queue stable is that the sensor device cannot handle all the generated tasks. The MEC algorithm is because if all the tasks are transmitted to the BS through the wireless channel, the transmission resource constraints will cause the task



(a) (b)

Fig. 5. Different algorithms vs. energy consumption and queue length.

backlog in the offloading queues. Furthermore, the constraint of ES computation resources will be unable to process all the offloaded tasks. Similarly, it can be seen in Fig. 5(b) that if we calculate all the tasks locally, the energy consumption will be far greater than our proposed TSRA strategy. Although the TSRA algorithm energy consumption is higher than the MEC algorithm, it can be known from Fig. 5(a) that the MEC algorithm also cannot guarantee the stability of the queue. Compared with the ICSOC algorithm, the TSRA algorithm outperforms the ICSOC algorithm in both energy consumption and queue length. Compared with the SDTO algorithm, the queue length is lower than the TSRA algorithm. However, together with these two algorithms, we can conclude that our TSRA algorithm sacrifices a little queue length to achieve lower energy consumption. The experience results confirm that our TSRA algorithm can make efficient decisions of task scheduling and resource allocation to decrease the energy consumption while ensuring queue stability.

VI. Conclusion

In this paper, we investigate the problem of task scheduling and resource allocation in self-powered sensor systems. We adopt the stochastic optimization approach to decompose the initial problem into a deterministic problem. Then, we design a named TSRA algorithm to deal with the problem. In detail, we jointly optimize the task scheduling, local CPU-cycle frequency allocation, transmission power allocation, and ES computation resource allocation. We aim to minimize the total average energy consumption of the sensor side while guaranteeing the system's performance. Finally, we conduct extensive experiments to verify the performance of our proposed algorithm. The results demonstrate that the TSRA algorithm can make dynamic decisions to balance the system performance and energy consumption effectively and have better performance compared with the baseline algorithms. One important direction in our future work is to incorporate practical examples to verify the performance of the proposed algorithm in real-world experiments.

References

[1] G. W. De Oliveira, M. Nogueira, A. L. d. Santos, and D. M. Batista, "Intelligent VNF placement to mitigate DDoS attacks on Industrial IoT," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1319–1331, Jun. 2023.

[2] J. Huang, F. Liu, and Z. Jianbing, "Multi-dimensional QoS evaluation and optimization of mobile edge computing for IoT: A survey," *Chin. J. Electron.*, vol. 33, no. 5, pp. 1–16, 2024.

[3] P. Rosa, A. Souto, and J. Cecílio, "Light-SAE: A lightweight authentication protocol for large-scale IoT environments made with constrained devices," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 2428–2441, Sep. 2023.

[4] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive Internet of Things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 28–33, Mar. 2017.

[5] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.

[6] D. Wu, J. He, H. Wang, C. Wang, and R. Wang, "A hierarchical packet forwarding mechanism for energy harvesting wireless sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 92–98, Aug. 2015.

[7] E. Andrade, F. Matos, and A. Santos, "A virtual models-based CAVs platoon resilient to network and sensor attacks," *Ad Hoc Netw.*, vol. 149, 2023, Art. no. 103225.

[8] X. Su, Y. Ren, Z. Cai, Y. Liang, and L. Guo, "A Q-learning based routing approach for energy efficient information transmission in wireless sensor network," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1949–1961, Jun. 2023.

[9] J. Huang, M. Zhang, J. Wan, Y. Chen, and N. Zhang, "Joint data caching and computation offloading in UAV-assisted Internet of Vehicles via federated deep reinforcement learning," *IEEE Trans. Veh. Technol.*, to be published.

[10] D. K. Sah, A. Hazra, R. Kumar, and T. Amgoth, "Harvested energy prediction technique for solar-powered wireless sensor networks," *IEEE Sensors J.*, vol. 23, no. 8, pp. 8932–8940, Apr. 2023.

[11] F. Zhao, Y. Chen, Y. Zhang, Z. Liu, and X. Chen, "Dynamic offloading and resource scheduling for mobile-edge computing with energy harvesting devices," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 2154–2165, Jun. 2021.

[12] H. Xiao, J. Huang, Z. Hu, M. Zheng, and K. Li, "Collaborative cloud-edge-end task offloading in MEC-based small cell networks with distributed wireless backhaul," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 4, pp. 4542–4557, Dec. 2023.

[13] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–18, Apr. 2019.

[14] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1133–1146, Jun. 2020.

[15] X. Deng, J. Li, L. Shi, Z. Wei, X. Zhou, and J. Yuan, "Wireless powered mobile edge computing: Dynamic resource allocation and throughput maximization," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2271–2288, Jun. 2022.

[16] J. Huang et al., "Incentive mechanism design of federated learning for recommendation systems in MEC," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2596–2607, Feb. 2024.

[17] Y. Chen, J. Xu, Y. Wu, J. Gao, and L. Zhao, "Dynamic task offloading and resource allocation for NOMA-aided mobile edge computing: An energy efficient design," *IEEE Trans. Services Comput.*, early access, Mar. 12, 2024, doi: 10.1109/TSC.2024.3376240.

[18] G. Coulby, A. K. Clear, O. Jones, and A. Godfrey, "Low-cost, multimodal environmental monitoring based on the Internet of Things," *Build. Environ.*, vol. 203, Oct. 2021, Art. no. 108014.

[19] M. Al-Hawawreh, I. Elgendi, and K. Munasinghe, "An online model to minimize energy consumption of IoT sensors in smart cities," *IEEE Sensors J.*, vol. 22, no. 20, pp. 19524–19532, Oct. 2022.

[20] Y. Deng, Z. Chen, X. Yao, S. Hassan, and A. M. A. Ibrahim, "Parallel offloading in green and sustainable mobile edge computing for delay-constrained IoT system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12202–12214, Dec. 2019.

[21] Y. Li, J. Li, Z. Lv, H. Li, Y. Wang, and Z. Xu, "GASTO: A fast adaptive graph learning framework for edge computing empowered task offloading," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 932–944, Jun. 2023.

[22] J. Huang, B. Ma, Y. Wu, Y. Chen, and X. Shen, "A hierarchical incentive mechanism for federated learning," *IEEE Trans. Mobile Comput.*, submitted for publication.

[23] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "Fog computing for energy-efficient data offloading of IoT applications in industrial sensor networks," *IEEE Sensors J.*, vol. 22, no. 9, pp. 8663–8671, May 2022.

[24] H. Jiang, X. Dai, Z. Xiao, and A. Iyengar, "Joint task offloading and resource allocation for energy-constrained mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 4000–4015, Jul. 2023.
[25] Z. Tong, J. Cai, J. Mei, K. Li, and K. Li, "Dynamic energy-saving offloading strategy guided by Lyapunov optimization for IoT devices," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19903–19915, Oct. 2022.
[26] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. Shen, "Energy efficient dynamic offloading in mobile edge computing for Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 1050–1060, Jul.–Sep. 2021.
[27] M. Hua, H. Tian, and W. Ni, "Online scheduling and optimality analysis for NOMA-based MEC systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2021, pp. 1–6.
[28] M. Guo, W. Wang, X. Huang, Y. Chen, L. Zhang, and L. Chen, "Lyapunov-based partial computation offloading for multiple mobile devices enabled by harvested energy in MEC," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 9025–9035, Jun. 2022.
[29] C. Pan, "Optimizing embedded software of self-powered IoT edge devices for transient computing," Ph.D. dissertation, School Eng., Univ. Pittsburgh, Pittsburgh, PA, USA, 2020.
[30] J. Mei, L. Dai, Z. Tong, X. Deng, and K. Li, "Throughput-aware dynamic task offloading under resource constant for MEC with energy harvesting devices," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 3460–3473, Sep. 2023.
[31] M. S. Munir, N. H. Tran, W. Saad, and C. S. Hong, "Multi-agent meta-reinforcement learning for self-powered and sustainable edge computing systems," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3353–3374, Sep. 2021.
[32] T. Mahn, H. Al-Shatri, and A. Klein, "Distributed algorithm for energy efficient joint cloud and edge computing with splittable tasks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–7.
[33] R. Lin et al., "Distributed optimization for computation offloading in edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8179–8194, Dec. 2020.
[34] F. Pervez, A. Sultana, C. Yang, and L. Zhao, "Energy and latency efficient joint communication and computation optimization in a multi-UAV assisted MEC network," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 1728–1741, Mar. 2024.
[35] Y. Chen, K. Li, Y. Wu, J. Huang, and L. Zhao, "Energy efficient task offloading and resource allocation in air-ground integrated MEC systems: A distributed online approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8129–8142, Aug. 2024.
[36] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.
[37] P. Lai et al., "Edge user allocation with dynamic quality of service," in *Proc. Int. Conf. Service-Orient. Comput.*, 2019, pp. 86–101.

Jiajie Xu is currently pursuing the M.Eng. degree with the School of Computer Science, Beijing Information Science and Technology University, Beijing, China. His current research interests include edge computing and machine learning.

Kaixin Li is currently pursuing the M.Eng. degree with the School of Computer Science, Beijing Information Science and Technology University, Beijing, China. His current research interests include edge computing, stochastic optimization theory, and game theory.

Ying Chen (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2017. She was a joint Ph.D. student with the University of Waterloo, Waterloo, ON, Canada, from 2016 to 2017. She is a Professor with the Computer School, Beijing Information Science and Technology University, Beijing. Her current research interests include Internet of Things, mobile edge computing, wireless networks and communications, machine learning. She is the recipient of the Best Paper Award with IEEE SmartIoT 2019, the Google Ph.D. Fellowship Award in 2016, the Google Anita Borg Award in 2014, and the 2022 OUTSTANDING CONTRIBUTION Award in 18th EAI CollaborateCom. She serves/served as a Leading Guest Editor for *Journal of Cloud Computing* (Springer), a TPC Member of IEEE HPCC, and a PC Member of IEEE Cloud, CollaborateCom, IEEE CPSCom, and CSS. She is also a Reviewer of several journals, such as the *IEEE Wireless Communications Magazine*, IEEE TDSC, IEEE IOT, IEEE TCC, and IEEE TSC.

Jiwei Huang (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in computer science and technology from Tsinghua University in 2009 and 2014, respectively. He was a Visiting Scholar with the Georgia Institute of Technology. He is currently a Professor and the Vice-Dean of the College of Artificial Intelligence, China University of Petroleum, Beijing, China, and the Director of the Beijing Key Laboratory of Petroleum Data Mining. His research interests include services computing, Internet of Things, and edge computing. He has published one book and more than 70 articles in international journals and conference proceedings, including the lEEE TMC, IEEE TSC, IEEE TCC, IEEE TVT, ACM SIGMETRICS, IEEE ICWS, and EEE SCC. He is currently on the editorial board of *Chinese Journal of Electronics and Scientific Programming*.