Xiaoshuang *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:153
https://doi.org/10.1186/s13638-019-1421-5

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Energy harvesting-based data uploading for Internet of Things

Gaofei Sun[1,2], Xiaoshuang Xing[1*] and Xiangping Qin[1]

Abstract

Uploading data from tremendous devices is one of the most challenging tasks for Internet of Things (IoT) due to the heterogeneous data features and the energy constraints of IoT devices. In this paper, we study the data uploading problem from two tiers. In the first tier, we focus on the scheduling of data uploading from different IoT devices to a specific access point (AP) by considering the heterogeneous data features such as the freshness of the data, the data length, the data uploading period, and the energy state of the IoT device. In the second tier, we deal with the selection among different APs for one IoT device. An AP selection algorithm is proposed and we prove that the AP selection process would eventually reach a stable state. Simulation study shows that our proposed algorithm can improve the successful data uploading rate and the time slot utilization.

Keywords: Energy harvesting, Data uploading, Internet of Things

1 Introduction

With the expected 50 billion deployment of Internet of Things (IoT) devices in 2020, massive data is to be uploaded through wired or wireless networks [1, 2]. However, it is hard to satisfy such data uploading demands with existing network structures due to the following reasons: first, massive data is to be uploaded from the huge amount of IoT devices; second, data from different IoT devices are with heterogeneous features, such as data volume and uploading period; third, some kinds of IoT devices acquire energy from ambiance around [3, 4] which brings a high uncertainty on their energy states.

To deal with the data uploading and/or transmission requirements of IoT devices, several schemes and protocols have been proposed. For example, there already existed several power supply monitoring applications used for data acquiring by using 2G, 3G, or 4G cellular networks. Within WiFi coverage, most of IoT devices are connected by accessing into femtocells. The Bluetooth and LORA are also used by IoT devices. Along with the 5G network evolution, narrow band-Internet of Things (NB-IoT) and EMTC are proposed for IoT communications. However, there are still some concerns to be solved as follows: First, the coverage of any given scheme cannot be perfect anywhere. Therefore, it is unreliable for IoT devices to rely on one specific access scheme. Second, information fusion [5] has not been considered in existing scheme which brings large information redundancy. Practically, for monitoring the environment in a given area, the acquired information from several IoT devices can be fused locally and upload the derived key information only. Third, no existing scheme considers the energy states of IoT devices. We should take the energy states of the IoT devices into consideration when scheduling their data uploading since the data uploading process cannot be conducted without enough energy supply.

In this paper, we deal with these concerns by designing a three-layer network framework and scheduling the data uploading from two tiers, as shown in Fig. 1. In the three-layer framework, the IoT devices with short-range wireless transmission capability belong to the bottom layer; the access points (APs) with wide-range transmission capability belong to the middle layer; and the traditional backbone networks, such as the cellular base station or the cable-based computer networks, belong to the top layer. Within the framework, each IoT device accesses to an appropriate AP for data uploading. Each AP acts as a controller that schedules the data uploading of the accessing devices and ensures the AP's spectrum efficiency. The main contributions of our work are as follows:

*Correspondence: xing@cslg.edu.cn
[1] Changshu Institute of Technology, Hushan Road, Changshu, China
Full list of author information is available at the end of the article

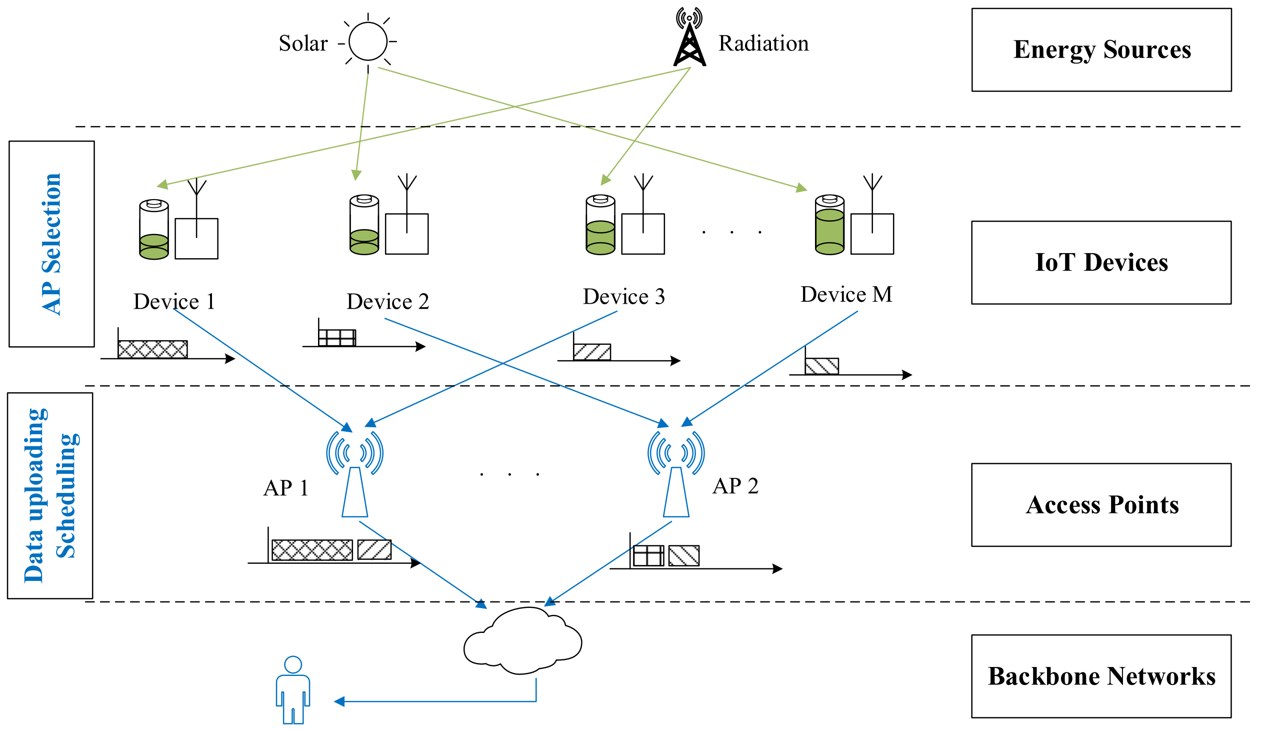

Fig. 1 System model. The figure describes the considered system model and the problems we are solving

- In the first tier, we propose a scheduling algorithm where an emergency function is defined for each data uploading requirement by considering the freshness of the data, the data length, the data uploading period, and the energy state of the IoT device. Then, the priority of the data uploading requirements is decided according to their emergency functions.
- In the second tier, we propose an AP selection algorithm where each IoT device accesses to the AP that can provide the highest expected data uploading probability. It is proved that the AP selection process would eventually reach a stable state.
- A central slot utilization and also a distributive AP selection algorithm for IoT devices are proposed, to achieve the joint optimization on data uploading and energy harvesting (Table 1).

The rest of this paper is organized as follows. In Section 2, we summarize the related works on energy harvesting and data uploading. In Section 3, we describe the system model and formulate the data uploading problem. In Section 4, we analyze the data uploading of multiple IoT devices accessing to one identical AP and propose a scheduling algorithm based on the urgency functions of the IoT devices. In Section 5, the AP selection problem is solved by designing a distributive AP selection algorithm. The simulation results are given in Section 6, and we concluded our work in Section 7.

2 Related work

Wireless energy harvesting (WEH) is critical for IoT devices due to their long-term deployment and self-sustainable operation features [6]. IoT devices with WEH capability can harvest energy from environmental sources such as vibration, solar, thermal, and wireless RF [7]. The harvested energy is converted to usable power by the RF-to-DC rectifier and the power management unit (PMU). Several design issues have been studied for WEH including circuit hardware design, multi-antenna-based WEH, energy converting, and so on [3]. Moreover, information disseminating algorithm [8–10], privacy-preserving scheme [11, 12], and dominating set discovering algorithm [13] have been proposed for better utilization of the harvested energy in energy harvesting networks.

For data uploading, the well-investigated scheme is the harvest-then-transmit protocol [8–10]. In [8], the joint optimization problem of down-link RF energy harvesting and up-link information transmission is studied based on time division multiple access (TDMA). A common-throughput metric is proposed to address the doubly near-far problem. Ju and Zhang [14] extend the protocol proposed in [8] by incorporating user cooperation. Sun et al. [10] propose a low-complexity iteration algorithm to achieve a similar throughput as in [8]. However, these work mainly focus on the fairness among multiple users while sacrificing the system throughput to a certain extent. In order to maximize the system throughput, [15, 16] schedule some users for data transmission on certain frequency blocks at given time while arranging other users for energy harvesting. Ng et al. and Sun et al. [17, 18] study the data uploading scheme that can maximize the energy efficiency. By exploring nonlinear fractional programming and Lagrange dual decomposition, an iterative algorithm is proposed to achieve better performance in average energy efficiency and system performance. However, all the abovementioned schemes are centralized that perform well in small-scale network. With the increasing number of users in the network, the computation complexity of these centralized scheme will become intractable.

To accommodate with the data uploading requirements of the large number of users/devices, the multi-tier network structure has been proposed in [19–21] without considering the energy harvesting capabilities of the users. Devices are divided into groups, and a device is selected as the gateway/AP within each group. The devices inside the same group may transmit their data to the gateway/AP while the gateway/AP can merge the information of the group and decide the accessing strategy to decrease the data flow in the backbone networks and improve the spectrum efficiency.

Our work is more challenging than the aforementioned ones since we take both the large-scale property of the IoT network and the energy harvesting capability of the IoT devices into consideration when designing the data uploading scheme.

3 System model and problem formulation

As shown in Fig. 1, we consider a network consisting of three layers: the IoT devices, access points, and the backbone infrastructures. The IoT devices are equipped with energy harvesting capabilities and can be deployed for different applications. Each device is equipped with short-range communication module for data uploading, and this module can support multiple technologies, such as WiFi, BLE4.0, and cognitive access. Each device can select a proper AP for data uploading according to its requirement and communication load. Each AP establishes one specific communication link to the backbone network. For example, AP 1 links to the Internet by wired connection and AP 2 links to cellular base station by NB-IoT. The main concern here lies in two aspects: how to schedule the data uploading of devices which select the identical AP, and how devices select the appropriate AP for data uploading among multiple APs. Without loss of generality, we make the following assumptions: First, the data uploading within an AP operates in a slotted manner (as TDMA). Multiple devices share the same spectrum band by using different time slots. Second, there exists a specific channel for command control between devices and APs. The information about spectrum usage and slot scheduling is exchanged on this channel. An example of the data structure of the devices and the scheduled uploading at AP is given in Fig. 2.

Explicitly, we consider that there are M IoT devices and N APs in a given area, which are denoted by $\mathcal{M} = \{1, \cdots, i, \cdots, M\}$ and $\mathcal{N} = \{1, \cdots, j, \cdots, N\}$, respectively. These IoT devices are deployed for different applications, thus have different data uploading demands. We characterize an IoT device $i, i \in \mathcal{M}$ with the following parameters: the data volume s_i, the data uploading period t_i, and

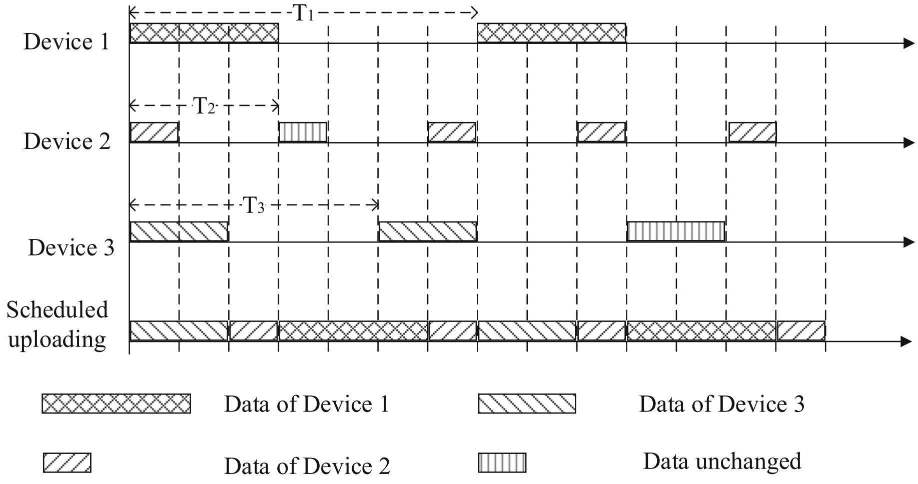

Fig. 2 Frame structure and scheduled uploading. The figure describes the data structure of devices 1, 2, and 3 and the scheduled uploading of these three devices at the AP

the energy state e_i. The data volume denotes how many slots are needed to upload its data through AP. The data uploading period t_i denotes the intervals for data acquirement, such as the environment monitoring, or system state detections. The energy state e_i denotes the current energy of device i, determining whether the data uploading can be performed. Further, we divide each device's energy state into $K+1$ states, from empty energy to full energy, and introduce a matrix for describing the energy harvesting process as

$$\Xi_i = \begin{bmatrix} \xi^i_{0,0} & \xi^i_{0,1} & \cdots & \xi^i_{0,K} \\ \xi^i_{1,0} & \xi^i_{1,1} & \cdots & \xi^i_{1,K} \\ \vdots & \vdots & \ddots & \vdots \\ \xi^i_{K,0} & \xi^i_{K,1} & \cdots & \xi^i_{K,K} \end{bmatrix} \tag{1}$$

Here, $\xi^i_{k_1,k_2}, k_1, k_2 \in [0, K]$ means the probability that the energy state of device i changes from state k_1 to state k_2. For example, $\xi^i_{0,0}$means the probability that device i does not acquire any energy and remains in empty energy and $\xi^i_{0,k}$means the probability that device i has acquired k unit energy from state 0. Note that, although these devices may be equipped with identical energy harvesting module, they also have different Ξ_i due to different data sensing periods. Moreover, Ξ_i only represents the energy harvesting ability of device i, but not the energy consumption of data sensing and transmission. Hence, the element $\xi^i_{k_1,k_2}, k_1 > k_2$ has zero probability.

We further denote the energy required for finishing data upload as e^r_i. Device $i, i \in \mathcal{M}$ with $e_i \geq e^r_i$ can upload its data to an AP, and its energy becomes $e_i - e^r_i$. We define matrix H_i as an indicator whether data uploading is performed,

$$H_i \left[\begin{matrix} & 1 & 0 & \cdots & & \cdots & 0 \\ & 0 & 1 & \cdots & & \cdots & 0 \\ = & 0 & \mathcal{F} & 1-\mathcal{F} & & \cdots & 0 \\ & \vdots & \vdots & \vdots & & \ddots & \vdots \\ & 0 & 0 & 0 & & \mathcal{F} & 1-\mathcal{F} \end{matrix}\right] \tag{2}$$

Note that the energy state between $\mathcal{F}$ and $1-\mathcal{F}$ is e^r_i for device i. Hence, we derive the energy transfer probability as $\Xi_i H_i$ to represent the sequential process of energy harvesting and data uploading.

Whether device i uploads its sensing data depends on its acquired data and the uploading strategy. For saving energy, device i would not upload its data if the current data that is unchanged refers to the last period. Similarly, device i can postpone its current uploading to the next period when it infers that the data acquired in the next period is unchanged or energy is below the threshold e^r_i. Therefore, we model the data transition as a Markov chain and define the data transition matrix for device i as

$$\Gamma_i = \begin{bmatrix} \gamma^i_{1,1} & \gamma^i_{1,2} & \cdots & \gamma^i_{1,L_i} \\ \gamma^i_{2,1} & \gamma^i_{2,2} & \cdots & \gamma^i_{2,L_i} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma^i_{L_i,1} & \gamma^i_{L_i,2} & \cdots & \gamma^i_{L_i,L_i} \end{bmatrix} \tag{3}$$

Here, $\gamma^i_{m,n}$ represents the probability of transferring from state $m, m \in [1, L_i]$ to state $n, n \in [1, L_i]$. Note that the dimension of Γ_i is different referring to a specified sensing object.

The main issue is that how to satisfy these IoT devices' data uploading requirements. We classify these devices into groups as $\left\{\mathcal{M}_1, \cdots, \mathcal{M}_j, \cdots, \mathcal{M}_N\right\}, \mathcal{M}_j \cap \mathcal{M}_k = \phi, \forall j, k \in [1, N]$. Here, $\mathcal{M}_j$ denotes the set of devices uploading data through AP j and AP j needs to properly schedule the uploading requirements for achieving the spectrum efficiency. Taking the WiFi for an example, although there are 14 bands available in 2.4 GHz, only 1 band can be selected for a device at a given time. Thus, each device should work on an identical band. When scheduling, it is necessary to avoid collision when two devices are uploading their data simultaneously on the same slot through the same AP. When several devices have the identical data uploading period, we need to delay some of these devices and avoid possible collisions. In our model, the main problem is how to derive the optimal delay for devices, which is related to their energy state, data state, and so on. Hence, we incorporate depth κ as a metric for measuring the available remaining slots for data uploading. Therefore, $\kappa = 0$ means device i decides whether to upload its data based on its current data and energy state only. Similarly, $\kappa = 1$ means device i decides whether to upload its data based on its current data and energy and also the possible data and energy in the next period. By incorporating depth κ, a portion of devices may have a data uploading slots more than one period, even larger than two periods, thus avoiding the potential collision in advance. Note that our proposed mechanism is different from the traditional CSMA mechanism, as users delay a random number of slots before the next transmission. From a given time stamp $t = 0$ to relatively large time stamp $t = T$, we denote device i's data in each period as $\mathbf{d}_i = \left\{d^1_{i,j}, d^2_{i,j}, \cdots, d^{n_i}_{i,j}\right\}, n_i \cdot t_i \leq T \leq (n_i + 1) \cdot t_i$ once it selected AP j for uploading. Here, $d^k_{i,j}, k \in [1, n_i]$ denotes a serial of slots beginning with $t = t_i \cdot (k-1) + 1$ and ending with $t = t_i \cdot (k - 1 + s_i)$, that is $|d_{i,j}| = s_i$. By considering the possible uploading in the next κ periods, the data uploading slots can be changed, such as $d^2_{i,j}$ changed from $[t_i + 1, 2 \cdot t_1]$ to $[t_i + 1 + \triangle t, 2 \cdot t_1 + \triangle t]$. Hence, the derived data sequence of device i is denoted as $\tilde{\mathbf{d}}_i$. Note that the data sequence's transformation is affected by other devices which selected identical AP; thus, the new data sequences $\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \cdots, \tilde{\mathbf{d}}_{M_j}$ have already no collision between each other's uploading from AP *j*. Therefore, the optimization problem can be denoted as

$$\max_{\kappa,\tilde{\mathbf{d}}_1,\tilde{\mathbf{d}}_2,\cdots,\tilde{\mathbf{d}}_M} \frac{\sum_j \sum_{i,i\in\mathcal{M}_j} |\tilde{\mathbf{d}}_i|}{T} \tag{4}$$

the constraints are

$$\tilde{\mathbf{d}}_m \cap \tilde{\mathbf{d}}_n = \phi, \forall m, n \in [1, \mathcal{M}_j], j \in [1, N] \tag{5}$$

Therefore, this optimization problem needs an iterative process of two parts: devices select the proper AP for data uploading and the scheduling of data for devices which access the same AP. Thus, we first analyze the data uploading scheduling among devices under a given AP, then begin with the AP selection for *M* devices.

4 Data uploading scheduling within one identical AP

In this section, we consider the case when $\mathcal{M}_j$ devices decide to upload their data through AP *j*. We skip the trivial case that only one device needs to upload its data for a given slot, then focus on the case when there are multiple devices, more than 2, that need to upload their data at the same time. Hence, AP *j* needs to figure out which device should be uploading first, and other devices wait for the next available slot and avoid possible collisions.

As for comparison, we first derive the collision probability of $\mathcal{M}_j$ devices without scheduling. Taking an example of two devices, each with a data uploading period as three slots and the data volume is one slot, their data uploading probabilities are 0.3 and 0.8. It is apparent that their collision probability is 0.3*0.8/3. Hence, we begin with deriving the uploading probability for $|\mathcal{M}_j|$ devices $P_i^S, i \in \mathcal{M}_j$ by their energy transition matrix Ξ and data transition matrix Γ. As we mentioned before, the device would upload its data under conditions: enough energy, that is, $e_i \geq e_i^r$, and data changed compared with the previous period, that is $d_i \neq d_i^{-1}$. Therefore, we derive the data uploading probability of device $i, i \in \mathcal{M}_j$ as the following lemma.

Lemma 1 *The data uploading probability of device i is denoted as*

$$P_i^S = \left(\sum_{i=1}^{K+1} \pi_i^e \sum_{e_k \geq e_i^r, k \in [1,K+1]} \xi_{i,k} \right) \cdot \left(\sum_{i=1}^{L_i} \pi_i^d \sum_{j \neq i} \gamma_{ij}^i \right). \tag{6}$$

Proof Firstly, we notice the independence of energy acquirement process and data transition process, and we have

$$P_i^S = P\left(e_i \geq e_i^r\right) \cdot P\left(d_i \neq d_i^{-1}\right) \tag{7}$$

Next, we derive the expression of $P(e_i \geq e_i^r)$. Device *i* acquires energy from solar in each period, then decreases e_i^r by performing the data uploading; thus, we denote this process by $\Xi_i \cdot H_i$. Moreover, we derive the stationary probability over energy states as a solution of $\pi_i^e \Xi_i H_i = \pi_i^e$. Hence, the probability of device *i* with enough energy for data uploading is

$$P\left(e_i \geq e_i^r\right) = \sum_{i=1}^{K+1} \pi_i^e \sum_{e_k \geq e_i^r, k \in [1,K+1]} \xi_{i,k} \tag{8}$$

Similar to $P\left(e_i \geq e_i^r\right)$, we derive the stationary probability of L_i data states by solving $\pi_i^d \Gamma = \pi_i^d$, then we have

$$P(d_i \neq d_i^{-1}) = \sum_{i=1}^{L_i} \pi_i^d \sum_{j \neq i} \gamma_{ij}^i \tag{9}$$

Finally, by incorporating these two parts, we derive the conclusion. □

As we derived device *i*'s data uploading probability P_i^S, and the period s_i is given, we then derive the collision probability of $|\mathcal{M}_j|$ devices without scheduling for comparison.

Proposition 1 *The collision probability of* $|\mathcal{M}_j|$ *devices without scheduling is*

$$\begin{aligned} P_c \approx & \sum_{(m,n),m,n \in \mathcal{M}_j} \frac{P_m^S \cdot P_n^S}{[s_m, s_n]} - \\ & 2 \sum_{(m,n,k),m,n,k \in \mathcal{M}_j} \frac{P_m^S \cdot P_n^S \cdot P_k^S}{[s_m, s_n, s_k]} \end{aligned} \tag{10}$$

Proof We begin with collision among any two devices. For device $m, n, \in m, n \in \mathcal{M}_j$, and we denote the beginning slot as $t = 1$, then their uploading data may be colluded when $t = [s_m, s_n] \cdot k, k = 1, 2, \cdots$; here, $[\cdot]$ means the least common multiple (LCM) of two variables. Therefore, the overall collisions over the whole set is

$$P_c\,(2\text{devices}) = \sum_{(m,n),m,n \in \mathcal{M}_j} \frac{P_m^S \cdot P_n^S}{[s_m, s_n]} \tag{11}$$

However, when we calculate $P_c(2\text{devices})$, we count some of the collisions repeatedly, that is, when more than two devices' data collude at the same time. Hence, we need to reduce these collision probabilities. Similarly, the overall collisions between every three devices can be denoted as

$$P_c(3\text{devices}) = \sum_{(m,n,k),m,n,k \in \mathcal{M}_j} \frac{P_m^S \cdot P_n^S \cdot P_k^S}{[s_m, s_n, , s_k]} \tag{12}$$

As the probability $P_c(\text{3devices})$ is calculated three times when calculating $P_c(\text{2devices})$, we derive the modified collision probability as

$$P_c = P_c(\text{2devices}) - 2 \cdot P_c(\text{3devices}) \tag{13}$$

Also the collision between four and five devices may be counted repeatedly, and we omit these parts due to complexity and comparatively small than collision between two and three devices. □

$$U_i(n_i + T_i) = \begin{cases} \gamma_{d_i,d_i} \cdot \sum\limits_{j \neq d_i} \gamma^i_{d_i,j}, \bar{e}_i \geq 2e_i^{req}, \\ \gamma_{d_i,d_i} \cdot \sum\limits_{j \neq d_i} \gamma^i_{d_i,j} \cdot \sum\limits_{j=e_i^{req}}^{K} \xi^i_{\bar{e}_i - e_i^{req},j}, e_i^{req} \leq \bar{e}_i < 2e_i^{req}, \\ \sum\limits_{j \neq d_i} \gamma^i_{d_i,j} \cdot \sum\limits_{j < e_i^{req}} \sum\limits_{e_i^{req} \leq k \leq K} \xi^i_{\bar{e}_i,j} \xi^i_{j,k} + \\ \gamma_{d_i,d_i} \cdot \sum\limits_{j \neq d_i} \gamma^i_{d_i,j} \cdot \sum\limits_{j \geq e_i^{req}} \sum\limits_{e_i^{req} \leq k \leq K} \\ \quad \xi^i_{\bar{e}_i,j} \xi^i_{j-e_i^{req},k}, \bar{e}_i < e_i^{req}. \end{cases} \tag{14}$$

By data uploading scheduling, devices may postpone their data to following periods depending on the data transition matrix and energy states. Hence, the main concern is how to sort the data uploading of these devices? Which device can upload in current slot and which devices must postpone their data uploading to following periods? For facilitating the scheduling process, we propose a single metric for each device, which is related to the following factors:

- The data sequence $\left\{d_i^{-1}, d_i, d_i^1, \cdots, d_i^k, \cdots\right\}$: here, d_i^{-1} represents the data uploading in the last period and d_i denotes the data to be uploaded in the current period; the accurate value of $d_i^1, \cdots, d_i^k, \cdots$ is unknown, and we only infer whether they are different from the current data d_i using the data transition matrix Γ.
- The energy sequence $e_i, e_i^1, \cdots$: e_i is the current energy state, and $e_i^1, \cdots$ represents the energy state in the following periods and can be inferred by energy transition matrix Ξ.
- The data uploading period T_i: we denote the remaining slots in the current period as n_i, then the available slots can be $n_i + T_i$ if there is no data to be uploaded in the next period.
- The data length s_i: it is apparent that devices with larger data volume would endure lower successful uploading than others without considering data length s_i. Thus, we may incorporate data length s_i as a factor for data uploading scheduling to balance the fairness among devices.

Based on these parameters above, we need to derive the urgency function for devices in $\mathcal{M}_j$ as $U_i\left(T_i, n_i, d_i^{-1}, d_i, e_i\right)$. Depending on the available time slots and the depth κ, we have

$$U_i = f(s_i, T_i) \cdot \sum_{k=1}^{\kappa} \theta(n_i + k \cdot T_i) U_i(n_i + k \cdot T_i), \tag{15}$$

here $\theta(n_i + k \cdot T_i)$ is a decreasing factor referring to the number of slots, which reflect the inaccuracy of $U_i(n_i + k \cdot T_i)$.

We begin with deriving $U_i(n_i)$, that is new data maybe uploaded in the next period, then only n_i slots can be used for uploading the current data; this can be classified into four cases:

1. The last period sensing data d_i^{-1} was sent successfully, and also the data is unchanged in the current period, that is $d_i^{-1} = d_i$. In this case, device i need not to participate in transmission competitions, thus denote $U_i = 0$.

2. The sensing data satisfies $d_i \neq d_i^{-1}$; thus, IoT device i needs to transmit. The urgency function of device i depends on the remaining slots for transmitting its sensing data. We analyze the situation in the following cases:

(a) Device i also needs to transmit new data d_i^1 in the next period; thus, we have $e_i^1 \geq e_i^{req}$ and $d_i^1 \neq d_i$. Hence, we have $U_i(n_i) = P\left(e_i^1 \geq e_i^{req}, d_i^1 \neq d_i\right)$.

(b) Device i need not to transmit in the next period but to transmit in the period following the next. Hence, we have $U_i(n_i + T_i) = P\left(e_i^1 < e_i^{req} \text{or} d_i^1 = d_i\right) \cdot P\left(e_i^2 \geq e_i^{req}, d_i^2 \neq d_i\right)$.

Similar to the derivation of $U_i(n_i)$, $U_i(n_i + T_i)$ above, we can derive $U_i(n_i + k \cdot T_i), \forall k = 2, \cdots$.

3. The last period sensing data d_i^{-1} is not transmitted successfully, and the data in the current period is unchanged or the energy is below the threshold; this case is the same as the case above.

4. The last period sensing data d_i^{-1} is not transmitted successfully, and also $d_i^{-1} \neq d_i$; thus, data d_i^{-1} is discarded, and the data d_i in the current period needs to be sent. Hence, this case is also the same as the second case above.

Next, we need to calculate the expression for $U_i(n_i + k \cdot T_i)$. We begin with $U_i(n_i) = P\left(e_i^1 \geq e_i^{req}, d_i^1 \neq d_i\right)$. As data transition and energy transition are independent, we have $U_i(n_i) = P\left(e_i^1 \geq e_i^{req}\right) \cdot P\left(d_i^1 \neq d_i\right) = P\left(e_i^1 \geq e_i^{req}\right) \cdot \sum_{j \neq d_i} \gamma^i_{d_i,j}$. The energy state e_i^1 can be divided into two cases: First, $\bar{e}_i \geq e_i^{req}$, thus no matter whether device i obtains energy outside the sensing can be performed with probability 1. Second, $\bar{e}_i < e_i^{req}$, whether device i performs the data sensing depending on obtaining energy to achieve the minimum energy e_i^{req}, the probability can be calculated as $\sum_{j=e_i^{req}}^{K} \xi^i_{\bar{e}_i,j}$. Based on the analysis above, we derive the expression of $U_i(n_i)$ as in Eq. (16).

$$U_i(n_i) = \begin{cases} \sum\limits_{j\neq d_i} \gamma^i_{d_i,j}\,, & \bar{e}_i \geq e_i^{req}, \\ \sum\limits_{j=e_i^{req}}^{K} \xi^i_{\bar{e}_i,j} \cdot \sum\limits_{j\neq d_i} \gamma^i_{d_i,j}\,, & \bar{e}_i < e_i^{req}. \end{cases} \tag{16}$$

Similarly, we derive the expression of $U_i(n_i + T_i)$ as in Eq. (14). Hence, we derive the urgency function of device i as in Eq. (15).

5 AP selection among IoT devices

In this section, we deal with the AP selection problem for IoT devices. As we denoted before, the AP selection process is performed by IoT devices distributively. For each device, it will join the other APs once it can increase the data uploading probability. The main concern here is that how would a device know joining other APs can bring a larger successful data uploading probability. Although we aim to derive a distributive algorithm for AP selection, some key information is also needed and broadcast to IoT devices. We list the main information needed for AP selection as follows:

- The slot utilization sequence $\Theta_1, \cdots, \Theta_j, \cdots, \Theta_N$ of AP $j, j \in [1, N]$. As for multiple devices that joined AP j for data uploading, AP j assigned the slots to a device with the highest utility of urgency function. Hence, AP j has the accurate statics on slot utilization irrelevant to the data uploading competitions. Note that the slot utilization E_j is easy to derive and broadcast through control channels by APs
- The data uploading probability P_i^s of device $i, i \in [1, M]$. This probability can be derived by device i using energy harvesting matrix and also data transition matrix as in Lemma 1
- The actual data uploading probability A_i^s of device $i, i \in [1, M]$. This probability is derived by counting on the actual uploading data.

These parameters are the overall information needed for AP selection, the first one is derived by AP broadcasting, and the latter two are derived by devices. For devices and APs to derive these parameters accurately, the AP selection should be operated periodically with a relative larger slots.

Basically, we need to find a metric for IoT devices, facilitating their selection among APs. In order to reflect the actual slot usage for devices, the metric must be accurate and brief. Fortunately, the actual data uploading probability $A_i^s, i \in [1, M]$ can be used as a metric to measure the data uploading performance of APs. Suppose that device i selects AP j randomly at the beginning, then perform its data uploading through AP j. After a certain time T_s, each AP broadcasts its slot utilization probability as $\Theta_1, \cdots, \Theta_N$. At the same time, data uploading probability P_i^s and actual data uploading probability $A_{i,j}^s$ of device $i, i \in M$ are derived by device i. Before device i switches to another APs, it must derive the expected actual data uploading probability $\tilde{A}_{i,k}^s, k \in [1, N], k \neq j$.

Lemma 2 *The expected actual data uploading probability $\tilde{A}_{i,k}^s, k \in \{1, N\}, k \neq j$ is*

$$\tilde{A}_{i,k}^s = (1 - \Theta_k)\left(1 + \frac{s_i \ln\left(1 - P_i^s\right)}{T_i}\right). \tag{17}$$

Proof As the transmission rate is different between APs, thus the number of slots for data uploading is different either. For AP $k, k \neq j$, the number of slots is denoted as $s_{i,k}$. Further, we consider that the devices access AP k as a whole, with the data uploading probability Θ_k. Also, the uploading period is denoted as 1, that is, data would be uploaded in each slot. Suppose that device i has data to be uploaded in the current period, and the number of slots is s_i, we consider the following cases:

- Case 1: device i has data to be uploaded in the next period; thus, the remaining slots for data uploading is $T_i - s_i$. Hence, we can derive the successful uploading probability as the product of case 1 and other devices do not upload at these slots, that is, $P_i^s \cdot (1 - \Theta_k)\,\frac{T_i - s_i}{T_i}$.
- Case 2: device i does not have data to be uploaded in the next period but in next two periods; this case has the probability of $P_i^s\left(1 - P_i^s\right)$. Also, the remaining slots for data uploading is $2T_i - s_i$. Hence, the overall probability is $P_i^s\left(1 - P_i^s\right) \cdot (1 - \Theta_k)\,\frac{2T_i - s_i}{2T_i}$.
- Case k: device i has only 1 data to be uploaded in k period, thus has the probability of $P_i^s\left(1 - P_i^s\right)^{k-1}$. Hence, the overall probability is $P_i^s\left(1 - P_i^s\right)^{k-1} \cdot (1 - \Theta_k)\,\frac{kT_i - s_i}{kT_i}$.

Summarizing all the cases $1, 2, \cdots, k, \cdots$, we can derive the expected actual data uploading probability as

$$\begin{aligned} \tilde{A}_{i,k}^s =& P_i^s \cdot (1 - \Theta_k)\frac{T_i - s_i}{T_i} + P_i^s(1 - P_i^s) \cdot (1 - \Theta_k) \\ & \cdot \frac{2T_i - s_i}{2T_i} + \cdots + P_i^s\left(1 - P_i^s\right)^{k-1} \\ & \cdot (1 - \Theta_k)\frac{kT_i - s_i}{kT_i} + \cdots \\ =& (1 - \Theta_k)\left(1 + \frac{s_i}{T_i}\ln(1 - P_i^s)\right) \end{aligned} \tag{18}$$

□

Hence, we finish the proof.

Therefore, device i will calculate the data utilization vector $\tilde{A}^s_{i,1}, \cdots, \tilde{A}^s_{i,N}$ when APs broadcast their slot utilization $\Theta_1, \cdots, \Theta_N$, except for its current selection, that is, AP j. It is apparent that the best AP to be selected is $\max_k \tilde{A}^s_{i,k}$. We summarize the AP selection process as in Algorithm 2 (given in Table 1). The variable ϵ is a minimal positive number, which avoids the ping-pong effect between AP selections.

Then, the next question arises, that is, whether this AP selection process would eventually reach a stable state, no any devices change their selected AP and switch to another one. We begin with the slot utilization Θ_j of AP j. Considering that devices set $\mathcal{M}_j = \{1, \cdots, M_j\}$ selected AP j currently and have the actual data uploading probability as $A^s_{1,j}, \cdots, A^s_{M_j,j}$, thus we can derive that the slot utilization Θ_j is a summation of data uploading probability of device set $\mathcal{M}_j$, that is, $\Theta_j = \sum_{i=1}^{M_j} \frac{s_i}{T_i} P^s_i$. Hence, slot utilization Θ_j is a function of uploading period T_i, data volume s_i, and uploading probability P^s_i, that is, $\Theta_j = g(T_i, s_i, P^s_i)$. As we denoted in data uploading scheduling inside AP above, each device acquires the uploading opportunity by sending its urgency function, no collision is derived, but some data can be delayed and eventually discarded when new data is acquired, thus bringing the decreasing in actual data uploading probability. Hence, we conclude that the function $g(\cdot)$ has $g^{'}(\cdot) \geq 0, g^{''}(\cdot) \leq 0$. Based on this analysis, we derive the following proposition:

Proposition 2 *The AP selection process leads to a stable state in finite rounds.*

Proof Consider that the current AP selected by device i is AP j, with actual data uploading probability as $A_{i,j}$. As the slot utilization vector broadcasting by AP is $\Theta_1, \cdots, \Theta_N$, device i selects AP k with $\max_k \tilde{A}^s_{i,k}$ for data uploading. It is apparent that device i will increase its successful data uploading probability, that is, $\tilde{A}^s_{i,k} > A_{i,j}$. We further denote the slot utilization of AP j without device i is $\bar{\Theta}_j$ and the slot utilization of AP k by joining device i as $\bar{\Theta}_k$. Thus, we have $A_{i,j} = \left(1 - \bar{\Theta}_j\right)\left(1 + \frac{s_i \ln(1-P^s_i)}{T_i}\right)$. By incorporating $\tilde{A}^s_{i,k} > A_{i,j}$, we have $\bar{\Theta}_j > \Theta_k$.

Table 1 AP selection process of IoT devices

	Algorithm 2 : distributive AP selection process of IoT devices.
1:	The APs broadcast their data utilization $\Theta_1, \cdots, \Theta_j, \cdots, \Theta_N$ to devices through CC channel every T_{AP}, $T_{AP} \gg T_i, \forall i \in \mathcal{M}$ period.
2:	The devices compute the expected data uploading probability $\tilde{A}^s_{i,k}$ over APs according to equation (16).
3:	For device $i, i \in 1, \cdots, M$ who selected AP j for data uploading
4:	IF $\max_k \tilde{A}^s_{i,k} \geq A^s_{i,j} + \epsilon$ do
5:	Device i sends the request to AP k for data uploading.
6:	end
7:	end

As we defined before, the slot utilization is a function of data uploading load, that is, $\Theta_j = \sum_{i=1}^{M_j} \frac{s_i}{T_i} P^s_i, j \in \mathcal{N}$. Suppose that the current data uploading load in AP j, k are L_j and L_k, thus we have $\Theta_j = g(L_j), \Theta_k = g(L_k)$. Moreover, we have $\bar{\Theta}_j = g\left(L_j - \frac{s_i}{T_i} P^s_i\right), \bar{\Theta}_k = g\left(L_k + \frac{s_i}{T_i} P^s_i\right)$. Hence, we have

$$\begin{aligned} &\bar{\Theta}_j + \bar{\Theta}_k - \Theta_j - \Theta_k \\ =&(\bar{\Theta}_k - \Theta_k) - (\Theta_j - \bar{\Theta}_j) \\ =&\left(g\left(L_k + \frac{s_i P^s_i}{T_i}\right) - g(L_k)\right) - \left(g(L_j) - g\left(L_j - \frac{s_i P^s_i}{T_i}\right)\right) \end{aligned} \tag{19}$$

□

As we derived that $\bar{\Theta}_j > \Theta_k$ and $g^{'}(\cdot) \geq 0, g^{''}(\cdot) \leq 0$, we conclude that $\bar{\Theta}_j + \bar{\Theta}_k - \Theta_j - \Theta_k > 0$. This result indicates that device i changes its AP selection for increasing its own successful data uploading probability and also increases the overall slot utilization of both APs. However, the slot utilization of all APs, that is, $\sum_j \Theta_j$, is finite, thus may achieve its maximization by finite rounds.

6 Results and discussions

Along with the two-tier data uploading process, the performance of slot scheduling inside AP and AP selection among IoT devices is verified in this section. In the slot scheduling process, we need to figure out, by using proposed Algorithm 1 given in Table 2, whether the slot utilization increases and all IoT devices increase their successful data uploading probabilities compared with the transitional contention-based mechanism. In the AP selection process, we need to figure out whether the selection process halts in finite rounds and if the iterative process may increase the slot utilization on all APs gradually.

6.1 Slot utilization inside AP

In this case, a set of IoT devices $\mathcal{M}_j$ selects an identical AP j for data uploading. Firstly, we verify the performance of the proposed algorithm by gradually increasing the data uploading load. Given each IoT device has almost equal data load, that is, $\frac{P^s_m}{T_m} \cdot s_m \approx \frac{P^s_n}{T_n} \cdot s_n, \forall m, n \in \mathcal{M}_j$. Suppose that there is only 1 IoT device connected to AP j at the beginning, then we increase the number of IoT devices by 1 at a time and derive the simulation results as in Fig. 3. Here, each IoT device's data uploading load

Table 2 Data uploading process of the proposed mechanism

	Algorithm 1 : data uploading process of proposed mechanism.
1:	The IoT devices broadcast their data transmission requests to AP through CC channel.
2:	The AP decides whether devices' requests can be approved according to equation (14).
3:	while the current slot is idle, set as $T = 0$ **do**
4:	The AP informs IoT devices for bids submission.
5:	The IoT devices compute their bids by urgency function G_i and $f(s_i, T_i)$ according to equation (7), then submit bid b_i to AP for devices $i \in M$.
6:	The AP selects the highest bid among devices, and the winning device begins its data transmission.
7:	After the data uploading finished, set $T = T + \bar{s}_i$, where $\bar{s}_i$ is the data length of winning device.
8:	end

is set to 0.06, and the maximum load is 1.8 with 30 IoT devices, which is larger than the capacity of AP. As we depicted in the above red curve, the ratio of successful data uploading decreases with the variation on data uploading load. Even with data uploading load of 1.8, successful data uploading of our proposed algorithm achieves 0.58, much larger than 0.09 of the contention-based algorithm. Moreover, the above blue curve indicates that our proposed algorithm reaches almost 100% slot utilization, larger than 60% of the contention-based algorithm, which means that almost each slot is used for data uploading through AP. Besides, the above blue curve coincides with the assumption in Section 5 that the slot utilization on AP is a convex function of the data uploading load.

Secondly, we focus on the successful data uploading probabilities of IoT devices individually instead of the overall performance. Consider that there are nine devices that selected this AP and are classified into three classes. Different class has unequal sensing data length and also transmission period, due to heterogeneous energy harvesting matrices and data transition matrices of IoT devices. By generating their data sequences and calculation, we list their parameters, such as uploading period, data volume, and data uploading probability as in Table 3.

In Fig. 4, we compared the performance with different urgency functions. Note that we take random selection among nine devices for transmission as a benchmark scheme for comparison, which is depicted as the red curve. We observe that the class C devices with short period time have the lowest successful transmission rate, about 55%. If we take the simple method, by using the weighted vector, multiply the function $f(\cdot)$ only, as depicted in the green curve, the ratio of class C devices increases a lot by severely damaging the class A devices. Furthermore, we perform the simulations under the urgency function but depth $k = 0$, and depth $k = 1$ only, and also the urgency function with depth $k = 1$. The results indicate that using urgency function and setting depth $k = 1$ achieve better performance and balance among devices, with ratio above 85% for each class devices.

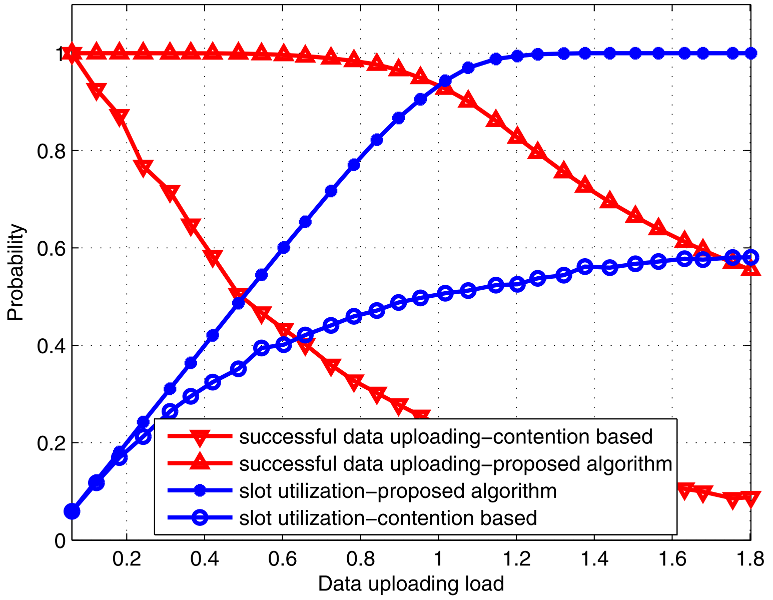

Fig. 3 The probability of successful data uploading and slot utilization of AP j. Performance comparison between our algorithm and the contention-based algorithm in terms of the successful data uploading probability and the slot utilization probability *j*

Table 3 Parameters of nine IoT devices

Classes	Period (slots)	Data (slots)	P_i^s
Class A	10	4	0.35
Class B	6	2	0.42
Class C	3	1	0.23

In Fig. 5, we compare the performance with different depth k. As expected, by considering the future data sensing and energy harvesting state, all devices increase their successful data transmission. However, further increments on depth k cannot bring obvious improvement in performance, but increase the computation complexity. It is apparent that the optimal depth is $k = 1$ under this simulation scenario.

In Fig. 6, we compared the efficiency of slot usage and data transmission under different urgency functions, where 1–7 in the x-axis represent cases listed as random selection, weighted, $G(\cdot)$ only, and urgency function with $k = 0, 1, 2, 3$. We observed that the proposed mechanism not only increases the successful data transmission, from 85% to above 90%, but also increases the ratio of slot usage. However, by different sensing period constraints, the slot usage cannot increase even by using higher depth k.

6.2 AP selection among IoT devices

In this subsection, we evaluate the performance on the AP selection algorithm. Specifically, we need to figure out whether the proposed algorithm leads to a stable state and if the overall successful data uploading probability increases along with the iterative process. In this case, we suppose that there are 20 IoT devices and 2 AP arranged in a given area. From the scratch, 20 IoT devices select AP 1 for data uploading and no device selects AP 2, then devices decide whether to switch to another AP in each round distributively after APs broadcast their current slot utilizations.

The slot utilization probabilities derived by 50 rounds are depicted in Fig. 7. Here, the blue curve indicates that the slot utilization of AP 1 decreases due to the departure of IoT devices. Meanwhile, the green curve increases as the number of devices gradually increased. Overall, the summation of slot utilization in both APs increases in the whole iterative rounds, which is coincided with the theoretical analysis in Proposition 2. Besides, the selection of IoT devices on APs steadied in less than 20 rounds, which indicates that our proposed algorithm has a well-convergence performance.

In Fig. 8, the successful data uploading probabilities of five devices out of these IoT devices are depicted. It is easy to find that the red curves indicate devices switched to AP 2 and the blue curves indicate devices persisted in AP 1. As AP 2 has less devices, thus users 1, 2, and 3 joined by increasing their successful data uploading probabilities directly to almost 1. Meanwhile, IoT devices which remain in AP 1 also increase their successful data uploading probabilities due to lower data uploading load. Hence, we conclude that the successful data uploading probabilities of IoT devices always increase during the iterative process.

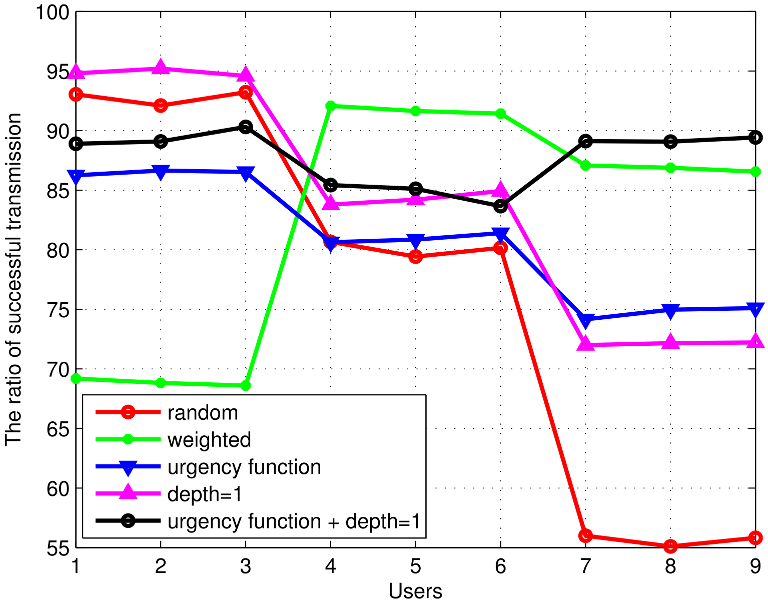

Fig. 4 Successful transmission of nine devices referring to different G_i. The figure shows the successful transmission ratio when different methods are applied for calculating G_i

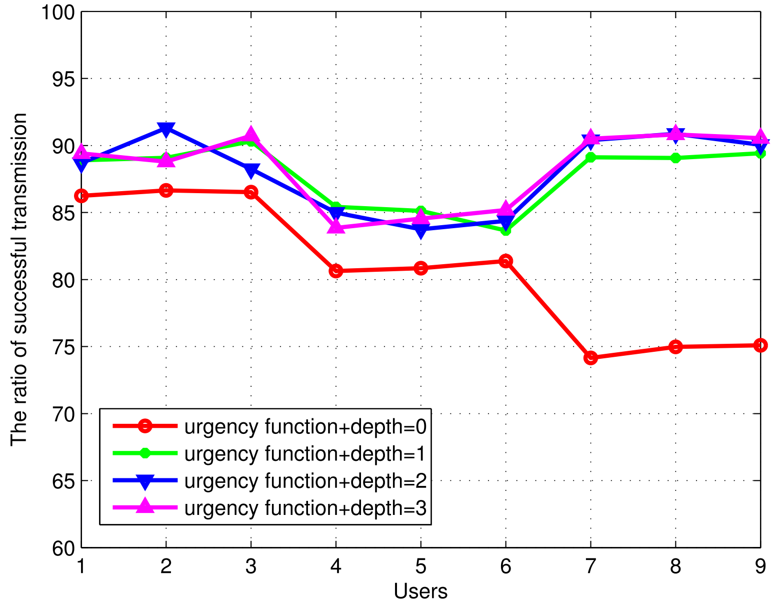

Fig. 5 Successful transmission of nine devices referring to different *k*. The figure shows the successful transmission ratio when different depth *k* are used in the emergency function

7 Conclusions

In this work, we analyze the scenario that how to schedule the data uploading of multiple energy harvesting enabled IoT devices among multiple APs. Firstly, a multi-layer data uploading process is proposed, which devices access the proper AP for data uploading and APs relay devices' data through different network access technologies. Secondly, a low-complexity slot allocation algorithm is proposed for devices which select the identical AP; the urgency function of devices ensures that the uploading period, data length, and remaining slots are considered and achieve the fairness among devices. Thirdly, a distributive AP selection algorithm is proposed and thus can achieve the stable selection among devices. Finally, the simulation results indicate that our proposed algorithms well balance the data uploading requirements and also afford multiple

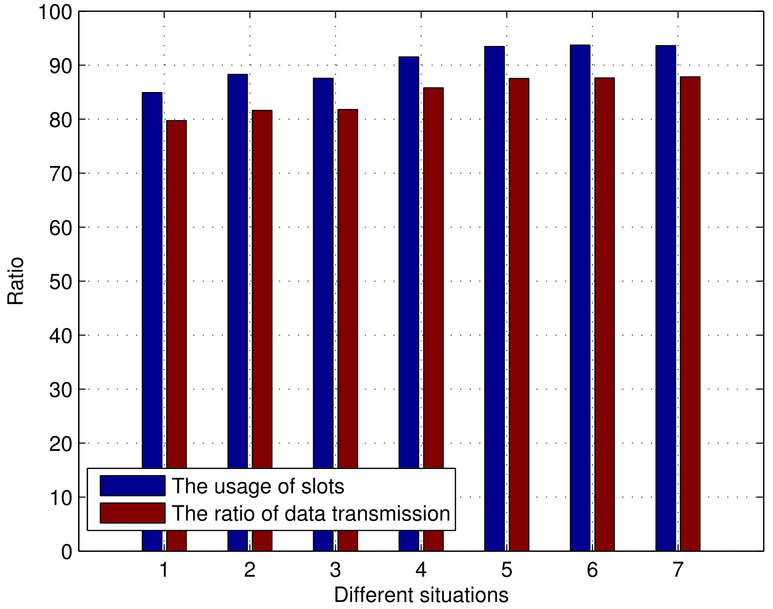

Fig. 6 The efficiency of slot usage and data uploading. The figure shows the efficiency of slot usage and data uploading under different situations

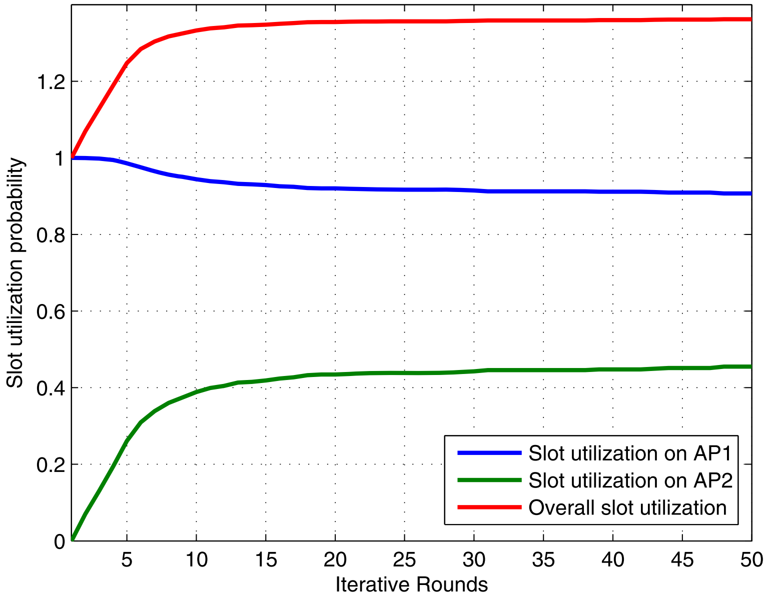

Fig. 7 The slot utilization of APs in 50 rounds. The figure shows the slot utilization of each AP and the total slot utilization of the existing two APs

access choices in heterogeneous networks with low cost and complexity.

8 Methods/experimental

In this paper, we designed a three-layer network framework and scheduled the data uploading from two tiers. In the first tier, we proposed a scheduling algorithm where an emergency function is defined for each data uploading requirement by considering the freshness of the data, the data length, the data uploading period, and the energy state of the IoT device. Then, the priority of the data uploading requirements is decided according to their emergency functions. In the second tier, we proposed an AP selection algorithm where each IoT device accesses to the AP that can provide the highest expected data uploading probability. Experimental results were performed using MATLAB R2009b on Intel? Core i5 system. The requirement of uploading data and the harvested energy were all constructed by an appropriate MATLAB code.

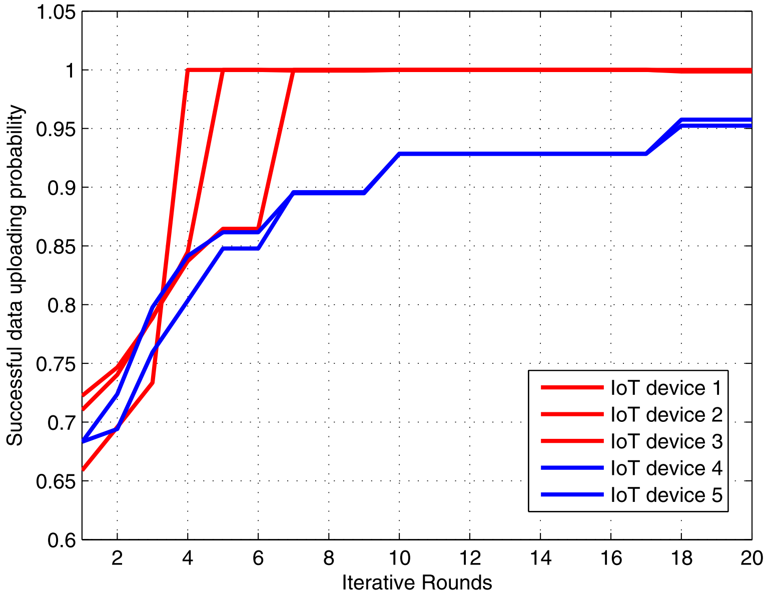

Fig. 8 The successful data uploading probabilities of individual IoT devices. The figure shows the successful data uploading probabilities of IoT device 1 to IoT device 5

Abbreviations

AP: Access point; IoT: Internet of Things; NB-IoT: Narrow band- Internet of Things; LORA: Long-range radio; WiFi: WIreless FIdelity; WEH: Wireless energy harvesting; RF: Radio frequency; DC: Direct current; PMU: Power management unit; MAC: Media access control; ISM: Industrial scientific medical; TDMA: Time division multiple access; CSMA/CA: Carrier sense multiple access/collision avoidance

Funding

The authors would like to thank the support from the Natural Science Foundation of China (61702056, 61602062), Educational Commission of JiangSu Province (17KJB520001), the Natural Science Foundation of JiangSu Province (BK20160410), the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1521, KJS1522), CERNET innovation project (NGII20160322).

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Authors' contributions

GS and XX conceived the scenario to be considered and carried out the theoretical analysis. GS carried out the simulations. All authors contributed to writing this paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]Changshu Institute of Technology, Hushan Road, Changshu, China.
[2]Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Shizi Road, Suzhou, China.

Received: 24 October 2018 Accepted: 3 April 2019
Published online: 11 June 2019

References

1. M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, L. Ladid, Internet of Things in the 5G era: enablers, architecture, and business models . IEEE J. Sel. Areas Commun. **34**(3), 510-527 (2016)
2. Z. Cai, X. Zheng, A Private and Efficient Mechanism for Data Uploading in Smart Cyber-Physical Systems. IEEE Trans. Netw. Sci. Eng., 1–1 (2018)
3. X. Lu, P. Wang, D. Niyato, D. I. Kim, Z. Han, Wireless networks with RF energy harvesting: a contemporary survey. IEEE Commun. Surv. Tutor. **17.2**, 757-789 (2015)
4. T. Shi, S. Cheng, Z. Cai, Y. Li, J. Li, Exploring Connected Dominating Sets in Energy-Harvest Networks. IEEEACM Trans. Netw. **25**(3), 1803-1817 (2017)
5. Q. Chen, H. Gao, Z. Cai, L. Cheng, J. Li, in *Book Energy-Collision Aware Data Aggregation Scheduling for Energy Harvesting Sensor Networks*. Energy-Collision Aware Data Aggregation Scheduling for Energy Harvesting Sensor Networks, (2018), pp. 117–125
6. K. Pouya, et al., Wireless energy harvesting for the Internet of Things. IEEE Commun. Mag. **53**(6), 102-108 (2015)
7. Y. Gang, C. K. Ho, Y. L. Guan, Dynamic resource allocation for multiple-antenna wireless power transfer. IEEE Trans. Sig. Process. **62**(14), 3565-3577 (2014)
8. H. Ju, R. Zhang, Throughput maximization in wireless powered communication networks. IEEE Trans. Wirel Commun. **13**(1), 418-428 (2014)
9. L. Liu, R. Zhang, K. Chua, Multi-Antenna Wireless Powered Communication With Energy Beamforming. IEEE Trans. Commun. **62**(12), 4349–4361 (2014)
10. Q. Sun, et al., A low-complexity algorithm for throughput maximization in wireless powered communication networks. arXiv preprint arXiv:1403.3665 (2014)
11. Q. Gao, T. Jing, X. Xing, X. Cheng, Y. Huo, D. Chen, *Simultaneous energy and information cooperation in MIMO cooperative cognitive radio systems*. (IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, 2015), pp. 351–356
12. Z. He, Z. Cai, Q. Han, W. Tong, L. Sun, Y. Li, An energy efficient privacy-preserving content sharing scheme in mobile social networks. Personal Ubiquit. Comput. **20**(5), 833-846 (2016)
13. T. Shi, S. Cheng, Z. Cai, L. Li, in *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*. Adaptive connected dominating set discovering algorithm in energy-harvest sensor networks
14. H. Ju, R. Zhang, User cooperation in wireless powered communication networks. available online at arXiv:1403.7123
15. W. Wang, L. Li, Q. Sun, J. Jin, in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*. Power Allocation in Multiuser MIMO Systems for Simultaneous Wireless Information and Power Transfer (IEE, Las Vegas, 2013), pp. 1–5
16. K. Huang, E. Larsson, Simultaneous information and power transfer for broadband wireless systems. IEEE Trans. Sig. Process. **61**(23), 5972-5986 (2013)
17. D. W. K. Ng, E. S. Lo, R. Schober, in *Proc. IEEE WCNC, Shanghai, China*. Energy-efficient resource allocation in multiuser OFDM systems with wireless information and power transfer, (2013), pp. 3823-3828
18. Q. Sun, L. Li, J. Mao, Simultaneous information and power transfer scheme for energy efficient MIMO systems. IEEE Commun. Lett. **18**(4), 600-603 (2014)
19. D. Niyato, D. I. Kim, P. Wang, L. Song, in *2016 IEEE International Conference on Communications (ICC)*. A novel caching mechanism for Internet of Things (IoT) sensing service with energy harvesting, (Kuala Lumpur, 2016), pp. 1–6
20. J. Kim, J.-W. Lee, in *Proc. IEEE 3rd ICUFN Dalian, China*. Energy adaptive MAC protocol for wireless sensor networks with RF energy transfer, (2011), pp. 89-94
21. P. Nintanavongsa, M. Y. Naderi, K. R. Chowdhury, in *Proc. IEEE INFOCOM-Mini Conf.* Medium access control protocol design for sensors powered by wireless energy transfer, (Turin, Italy, 2013), pp. 150-154