Computer Networks 269 (2025) 111414

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Optimal adaptive scheduling to maximize throughput for battery constrained time-varying RF-powered systems

Fangyu Zhou, Feng Shan *, Weiwei Wu, Runqun Xiong, Junzhou Luo

School of Computer Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, PR China

ARTICLE INFO

Keywords:
Wireless power transfer
RF-powered systems
Time varying wireless power
Battery capacity limitation

ABSTRACT

Radio Frequency (RF) wireless power transfer is a novel technique to address the energy hunger problem of modern wireless devices, for which power transfer and data transmission are coordinated by the "harvest-then-transmit" (HTT) protocol. Time-varying RF-powered systems is becoming a research trend and a significant progress has been made recently that proposes an optimal HTT-scheduling algorithm. However, previous research assume the core time-varying charging power function to be continuous, and the battery to be infinitely large, which ease the theoretical analysis. This paper considers a more practical discretely time-varying charging power function and battery overflow caused by limited capacity, and attack an even harder but important problem. We establish a set of optimality properties for the offline problem where the time-varying power transfer is known in advance. Based on these optimality properties, we propose a novel splitting line system, and an optimal iteration-based method to locate the s-lines for the Energy Critical Point (ECP) and Battery Full Point (BFP), respectively and adaptively. Following the optimality principles learned from the offline problem, we design an online heuristic, and its superior performance is demonstrated by simulations.

1. Introduction

Maintaining sufficient available energy has become a major issue for today's wireless devices, affecting their operational lifetime and user experience. This problem is even more severe when these devices are performing critical tasks. Energy harvesting (EH) and wireless power transfer (WPT) techniques have therefore been developed to address this issue. EH technology enables wireless devices to harvest energy from the surrounding environment [1] such as solar energy harvesting for electric autonomous vehicles [2], and underwater ultrasonic energy transfer [3]. Since WPT technology transfers energy wirelessly to the receiving devices, WPT has become an increasingly important option for wireless devices to harvest energy, and it has attracted growing interest from the research community. A number of major achievements have been obtained and reported, including magnetic wireless power transfer [4–6], and harvesting energy from radio frequency [7–10].

For most wireless devices in human habitats, energy from radio frequency (RF) is a crucial power source, because wireless signals carry not only information but also power. Hardware for RF power harvesting captures energy from everyday signals like TV broadcasts [11], WiFi [12], and Bluetooth [13]. Radio wave interference has also been used to charge multiple devices simultaneously [8,9]. Commercial products for RF wireless power transfer (WPT) are available from companies like Powercast [14] and WISP [15].

In RF-powered systems, devices typically harvest energy first and then transmit data. This sequential operation occurs for three main reasons [16]. Firstly, low-cost sensors often share crucial hardware like antennas between harvesting and transmission modules, preventing simultaneous use. Secondly, most energy harvesting devices use supercapacitors, which cannot support concurrent discharging and recharging. Lastly, limited bandwidth must be shared between operations. Hence the 'harvest-then-transmit' (HTT) principle is widely adopted.

With the rapid development of hardware for RF-powered systems, efficient HTT scheduling is becoming an increasingly important research area [12,17–24].

Our previous research addressed the HTT scheduling problem [16], aiming to maximize data transfer with limited dynamic collection power. However, it assumes that the charging power varies continuously over time and that the battery is infinite. In contrast, many studies of energy harvesting systems recognize the limitations of battery capacity and discrete time (a detailed survey is presented in Section 2). Combining the assumptions of finite battery capacity and discrete time complicates the problem, but is critical for practical applications (see Fig. 1).

* Corresponding author.
E-mail addresses: zhoufangyu@seu.edu.cn (F. Zhou), shanfeng@seu.edu.cn (F. Shan), weiweiwu@seu.edu.cn (W. Wu), rxiong@seu.edu.cn (R. Xiong), jluo@seu.edu.cn (J. Luo).

https://doi.org/10.1016/j.comnet.2025.111414
Received 2 November 2024; Received in revised form 15 April 2025; Accepted 25 May 2025
Available online 11 June 2025

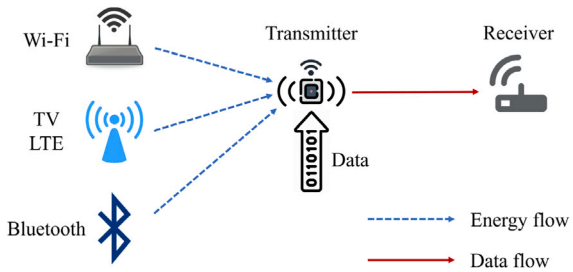


Fig. 1. The figure demonstrates a system that captures energy from ambient RF signals. It then uses this harvested energy to transmit its data.

1.1. Challenges and contributions

From the above discussion, we are very much motivated by the need of design an adaptive *HTT-scheduling* algorithm to transmit the maximum data within a given time duration for battery constrained time-varying RF-powered systems.

There are twofold challenges. **Firstly**, solving a problem in discrete solution space is generally harder than in continuous solution space. Discrete time power functions impose more restrictions than continuous time power functions. **Secondly**, the battery may overflow if it is charged with too much energy, adding another restriction where the optimal solution cannot store as much energy as it could with an infinitely large battery.

Our contributions in this paper are summarized as follows.

- We formulate an adaptive *HTT-scheduling* problem aiming at throughput maximization for battery constrained discretely-time-varying power transfer WPT channels.
- We observe a set of optimality properties for the offline problem where the time-varying power transfer is known in advance. Based on these optimality properties, we propose a novel splitting line system and s-line locating method that solves a special case optimally.
- For the general offline problem, the proposed splitting line system is utilized to iteratively locate the s-lines for the Energy Critical Point (ECP) and Battery Full Point (BFP), respectively and adaptively. To the best of our knowledge, it is the first adaptive *HTT-scheduling* algorithm that considers battery capacity.
- Following the optimality principles learned from the offline problem, we have designed an online heuristic algorithm for the throughput maximization problem. Its superior performance is demonstrated by simulations.

The remainder of this paper is organized as follows. In Section 2, we conducted a literature review. In Section 3, we formally define the system model and the optimization problems. An optimal scheduling algorithm for the defined problem is presented in two sections, Sections 4 and 5. The investigation, algorithms and simulation results for the online scheduling problem are presented in Section 6. Sections Section 7 concludes this paper.

2. Related work

Time-variation is a key characteristic of ambient energy harvesting, including RF power sources [20,21] and other energy sources [25]. In recent years, time-varying RF-powered systems have emerged as a prominent research focus [23,26–37].

In the field of the Internet of Things (IoT), Thakur et al. [31] and Potnis et al. [32] have respectively investigated the performance of RF-powered networks in multi-device and densely deployed environments, highlighting the potential of RF power transfer to enhance both energy and spectral efficiency. Lin et al. [33] explored RF-powered systems in rural areas, founding that IoT devices deeper in rural areas have higher coverage probability than those near the edges. This underscores the significance of energy harvesting in rural wireless networks. In the agricultural sector, Zahari et al. [34] analyzed the potential value of RF power transfer for sustainable farming, emphasizing its capability to reduce reliance on conventional batteries.

Moreover, RF-powered systems demonstrate broad application potential in battery-free devices and low-power sensors. Colell et al. [35] proposed a battery-free RF-powered circuit designed for non-contact voltage monitoring in power systems, validating the feasibility of remote power measurement using RF energy harvesting. Van Mulders et al. [36] designed battery-free RF-powered electronic shelf labels (ESLs), showcasing their feasibility for energy-efficient updates in retail environments. Sapurov et al. [37] investigated the application of RF energy harvesting in smart homes, using Wi-Fi signals to power leakage sensors and evaluating the impact of distance on energy collection efficiency.

Efficient scheduling strategies are essential to enhance the performance of these RF-powered systems. For time-varying RF-powered networks, the Harvest-Then-Transmit (HTT) protocol has become a widely adopted scheduling method. Shi et al. [38] combined HTT with backscatter networks, achieving improved network efficiency through optimized resource allocation. Kishore et al. [39] integrated opportunistic spectrum sensing (SS), ambient backscatter communication (ABC), and HTT into a unified framework, proposing an algorithm to optimize both throughput and energy efficiency in RF-powered cognitive radio systems, significantly boosting overall network performance.

Although HTT strategies have been extensively used in prior studies to optimize the energy efficiency of power transfer networks, throughput optimization has often been overlooked. To address this gap, with the introduction of an optimal HTT scheduling algorithm, our work achieves significant improvements in network throughput [16].

Most recently, our previous work [16] studied a fundamental HTT-scheduling problem aiming to transmit the maximum data within a given time duration, assuming the wireless harvesting power is time-varying. We propose an optimal method to determine when the transmitter should harvest energy (charging), when it should transmit data (sending), and what transmission power to use, in order to maximize data transmission with limited and dynamic harvested power. However, the previous work assume the dynamic time-varying charging power function is continuous in time, making it easier to analyze and solve. Moreover, the battery is assumed to be infinitely large, so battery overflow does not have to be considered, simplifying the problem.

In energy harvesting systems, it is widely accepted that batteries have limited capacities [28,40,41], and system time is discrete [42–44]. Feng et al. [28], Wang et al. [40], and Zhang et al. [41] assume the battery has constrained capacity and propose reinforcement learning-based methods to develop model-free solutions. Similarly, Kim et al. [42], Feng et al. [28], and Mohammadi et al. [44] assume time is discrete and also propose reinforcement learning-based methods to achieve their respective objectives.

Applying both assumptions makes the problem even harder to handle, but it is practically important.

3. Problem formulation

3.1. System model

We consider a communication channel model that comprises solely a data receiver and a wireless-powered data transmitter. The transmitter sends data to the receiver over an AWGN wireless communication channel, a model extensively utilized in existing literature [18,20,21]. At any given time, the transmitter can either harvest power or transmit data, but it is incapable of performing both actions simultaneously. The power transfer rate is characterized as a discrete, time-varying function.

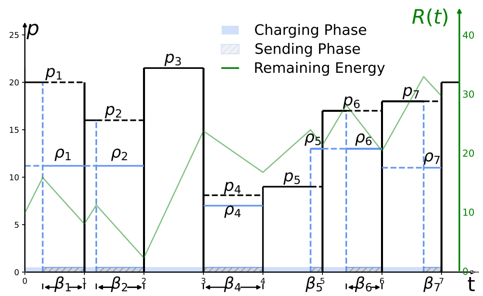


Fig. 2. Time is slotted and wireless supply power p_i varies from slot to slot. Each time slot is split into a charging phase and a sending phase. Transmission power is the power consumed in the sending phase. A natural idea is to charge when supply power is high and send data when it is low. However, defining *high* and *low*, and deciding what to do when the supply power is moderate, can be challenging.

We define a time axis partitioned into equal time slots, where each slot represents a unit of time, such as seconds or milliseconds, depending on the smallest discrete time step of the RF system, similar to the internal clock cycles in computer systems. The time-discrete charging power function is assumed to maintain a constant power transfer rate within each slot. Let T represent the total number of time slots to be scheduled and define the wireless power transfer rates (the unit of the transfer rates is consistent with the transmission power discussed later in the paper, such as watt) by the vector $\mathbf{p} = \{p_1, p_2, \dots, p_T\}$, where p_i denotes the power transfer rate in slot i, $i = 1, 2, \dots, T$. This vector is referred to as the supply power vector. For simplicity, we assume that all power supply rates are distinct. If two power supply rates are identical, we consider the one at the earlier time to be larger. This artificial ordering is introduced solely to enforce a consistent processing sequence in subsequent operations. However, it is important to emphasize that the two power supply rates remain equal in the set $\mathbf{p}$, the assigned ordering does not alter their actual values.

Definition 1. Each Slot i is split into two phases, the *Charging Phase* i and the *Sending Phase* i, with the corresponding lengths to be $1 - \beta_i$ and β_i respectively.

Moreover, if $\beta_i = 0$, Slot i is called a *charging slot*. It is called a *sending slot* if $\beta_i = 1$.

Definition 2. In *Sending Phase* i, the transmission power is denoted as ρ_i, which is subject to the range constraint of Eq. (1), where ρ_{max} is the maximum transmission power imposed by the hardware.

$$0 \le \rho_i \le \rho_{max}, \forall i \in [0, T]. \tag{1}$$

Fig. 2 shows the notions of time slot, charging (sending) phase and transmission power.

Let E_C and E_I represent the battery capacity and the initial battery energy, respectively (the unit of them depend on the units of the transmission power and the time slots). To facilitate the design and analysis of algorithms, we adopt the following assumptions. (1). $\rho_{max} < E_c \le \rho_{max} T$, meaning that a fully charged battery will not be depleted in a single time slot, even when transmitting at the maximum power level. (2). $E_I + \sum p_i \ge E_c$, implying that the battery can be fully charged if all time slots are used for power harvesting. We summarize the introduced notions in Table 1.

3.2. Problem formulation

Let $A(t)$ represent the total energy accumulated in the battery during the first t time slots. It can be expressed as:

$$A(t) = \sum_{i=1}^{t} p_i (1 - \beta_i), \quad t = 1, 2, \dots, T. \tag{2}$$

Table 1
Symbols of system models with their definitions.

Notation	Definition
T	total number of time slots
i	slot number
p_i	power transfer rate
β_i	sending phase length
ρ_i	transmission power
ρ_{max}	maximum transmission power
E_C	battery capacity
E_I	initial battery energy
r_i	transmission rate
N	thermal noise level
D	data throughput

Let $C(t)$ denote the total energy consumed during the first t time slots, which is given by:

$$C(t) = \sum_{i=1}^{t} \rho_i \beta_i, \quad t = 1, 2, \dots, T. \tag{3}$$

Let $R(t)$ represent the remaining energy in the battery at the end of slot t. It is calculated as:

$$R(t) = E_I + A(t) - C(t), \quad t = 1, 2, \dots, T. \tag{4}$$

For any t, the remaining energy $R(t)$ must be non-negative. This requirement is known as the *energy causality constraint* and can be expressed as:

$$R(t) \ge 0, t = 1, 2, \dots, T. \tag{5}$$

Due to the battery capacity, the remaining energy $R(t)$ must satisfy the constraint $R(t) \le E_c$. In any time slot t, since the charging phase occurs before the sending phase, the energy stored in the battery reaches a local peak value given by $R(t-1) + p_t(1-\beta_t) = E_I + A(t) - C(t-1)$ at time $t - \beta_t$. We therefore have the follow so called *battery capacity constraint*.

$$E_I + A(t) - C(t-1) \le E_c, t = 1, 2, \dots, T. \tag{6}$$

In the sending phase i, the transmission power ρ_i is related to the transmission rate r_i through the function of Eq. (7), which is commonly used for a single-user point-to-point AWGN channel [18,20,21].

$$r_i = \log(1 + \frac{\rho_i}{N}), \tag{7}$$

Where N represents the thermal noise level, which is often assumed to be one [45]. Consequently, the total amount of data transmitted over T time slots can be computed using the following equation.

$$D = \sum_{i=1}^{T} r_i \beta_i = \sum_{i=1}^{T} \beta_i \log(1 + \rho_i). \tag{8}$$

D in (8) is called the data throughput.

Definition 3 (*Adaptive HTT-Scheduling for Maximum Throughput*)**.** Let $\mathbf{p}$ be a wireless supply power vector in a WPT system as described above, the adaptive *HTT-scheduling* problem for maximum throughput is to determine the lengths $\{\beta_i\}$ and transmission powers $\{\rho_i\}$ for each sending phases i, $1 \le i \le T$ such that the data throughput D in (8) is maximized under the constraints defined by Eqs. (1), (5) and (6).

The scheduling problem described in Definition 3 is referred to as the offline case if the vector $\mathbf{p}$ is fully known prior to scheduling. Conversely, it is called the online problem when any p_t is unknown until slot t is being scheduled, for $t = 1, 2, \dots, T$.

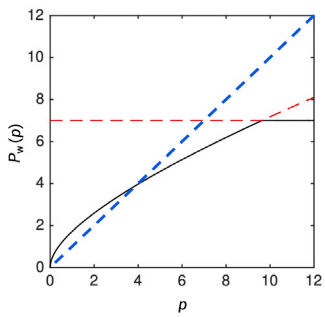


Fig. 3. The curve of function $\rho = P_s(p)$ when $\rho_{max} = 7$ [16] (The meanings of $P_s(p)$ and $P_w(p)$ are the same).

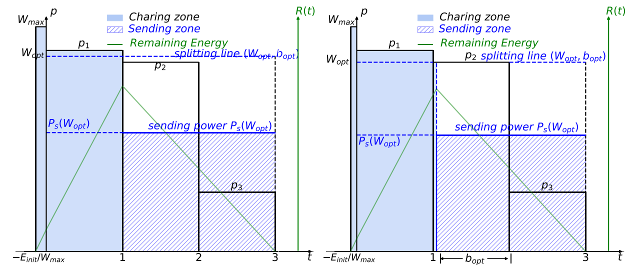


Fig. 4. Examples of the *splitting line system*. Initial energy is treated as pre-charged during a virtual charging phase. A splitting line (w, b) splits the curve into two parts. Above the splitting line is the *charging zone*, below the splitting line is *sending zone*, which is of height $P_s(w)$. The *up-the-stairs* algorithm is proposed to locate the optimal splitting line.

4. An optimal offline solution with unlimited battery capacity

Before tackling the complete problem, we first examine the case where the battery capacity is considered unlimited. This scenario is addressed in two steps. In the first step, we focus on a monotonically decreasing case, where the power supplies in the vector **p** are assumed to form a decreasing sequence. In the second step, we relax this assumption. We then handle the general case where power supplies need not be decreasing.

4.1. Preliminary research results for a single slot

In this case we consider only one time slot, our previous work [16] indicates that the transmission power is independent of the initial energy E_I and the duration of the time slot T_{slot}. Instead, it solely depends on the charging power. We quote this finding as follows.

Theorem 1 (*[16]*)**.** *The optimal sending power ρ_w is independent of E_I and T_{slot}, and it depends only on the harvesting power p when $E_I \le \rho_w T_{slot}$.*

Thus when we let β_{slot} denote the proportion of the sending phase in that slot, the optimal transmission power during the $\beta_{slot} T_{slot}$ time is as follows:

Definition 4 (*sOPT power Function $P_s(p)$ [16]*)**.** For any given wireless supply power p, the *sOPT power* $P_s(p)$ is defined as follows.

$$P_s(p) = \min\{ \frac{p-1}{\mathcal{W}(\frac{p-1}{e})} - 1, \rho_{max} \}, \tag{9}$$

where ρ_{max} is the maximum available transmission power which is imposed by the hardware.

Fig. 3 illustrates the curve of this function for the range $p \in (0, 12)$ and $\rho_{max} = 7$, providing an intuitive understanding of the relationship between the harvesting power and transmission power. For any given *sOPT power* ρ, where ρ is less than ρ_{max}, the inverse function can be computed to find the corresponding harvesting power as $p = P_s^{-1}(\rho)$, as discussed in [16].

4.2. Monotonically decreasing supply power case

In this case, the supply power vector $\mathbf{p} = \{p_1, p_2, \dots, p_T\}$ is assumed to follow a decreasing sequence, such that $p_1 > p_2 > \cdots > p_T$, which is different from previous work.

Theorem 2. *In the optimal solution, there must be a variable w_{opt} such that any slot with power $p > w_{opt}$ ($p < w_{opt}$) must be a charging (sending) slot. Meanwhile, the transmission power in all sending phases must be $P_s(w_{opt})$ and battery is empty at time T. Note, if in a slot $p = w_{opt}$, there must be another variable $b_{opt} \in [0, 1]$ such that the sending phase length is b_{opt}.*

Proof. See Appendix A. □

Depending on E_I and **p**, there are two distinct types of optimal solutions, *e.g.,* $w_{opt} \ne p_i$ for $\forall i$, and $w_{opt} = p_i$ for $\exists i$. Examples of these two solution types are illustrated in Fig. 4.

Once the optimal two-tuple (w_{opt}, b_{opt}) is determined, the optimal schedule is fully defined. The remaining challenge is to find the values of (w_{opt}, b_{opt}).

Definition 5 (*Splitting Line*)**.** Define two-tuple (w, b) to be the splitting line, denoted as $\Omega = (w, b)$, where $0 \le w \le w_{max} = P_s^{-1}(\rho_{max})$ and $0 \le b \le 1$. Define $(w_1, b_1) > (w_2, b_2)$, if $w_1 > w_2$ or $w_1 = w_2$ but $b_1 > b_2$.

Definition 6 (*Splitting Line Conversion*)**.** For the two-tuple (w, b), Ω raising corresponds to b increasing when $w = p_i$ for $\exists i$, and Ω raising corresponds to w increasing when $w \ne p_i$ for $\forall i$, and vice versa.

Definition 7 (*$P_s(w, b)$-Schedule*)**.** Given E_I, **p** and a splitting line $\Omega = (w, b)$, the $P_s(w, b)$-schedule is determined as:

- if $p_i > w$, we set $\beta_i = 0$, $\rho_i = 0$,
- if $p_i < w$, we set $\beta_i = 1$, $\rho_i = P_s(w)$,
- if $p_i = w$, we set $\beta_i = b$, $\rho_i = P_s(w)$.

The $P_s(w_{opt}, b_{opt})$-schedule represents the optimal schedule that maximizes throughput before the total time T. Therefore, the goal is to test various splitting lines $\Omega = (w, b)$ to identify the optimal one that yields the highest throughput.

Inspired by Algorithm 1 from [16], we propose an *up-the-stairs* algorithm to compute the optimal schedule, as illustrated in Fig. 4. In this figure, the monotonically decreasing supply power is depicted like a staircase. Initially, the splitting line ω is positioned at the base of the staircase, and we ascend one step at a time in a single loop. The process continues until we reach a point where $R(t) < 0$, which causes the loop to stop, indicating that the optimal splitting line lies between the two steps where the failure occurs. At this point, we can efficiently use the $P_s(p)$ function and its inverse from Definition 4 to quickly determine the exact location of the splitting line, and thus find the optimal solution.

4.3. The general p case

For a general **p** case, we can apply the *up-the-stairs* algorithm to find a splitting line by first sorting the power vector into a monotonically decreasing sequence. After obtaining the splitting line, we restore the original slot order to derive a solution, which is optimal if the energy causality principle is satisfied. This process is referred to as *s-line_raising_general*. However, in the general case, a single splitting line may not suffice to solve the problem. For instance, as shown in Fig. 5, when

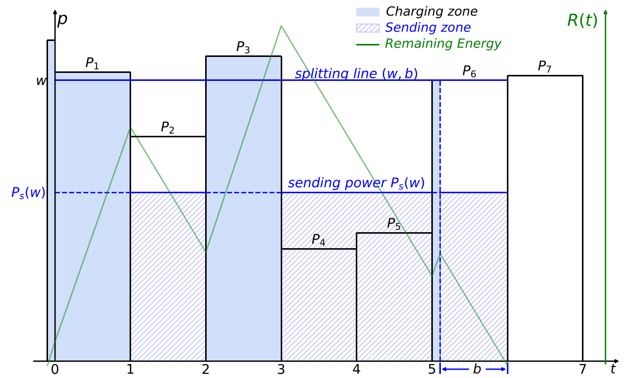


Fig. 5. Algorithm *s-line_raising_general* is illustrated. Its procedure is, starting from $(0, 0)$, the splitting line slowly arises, it stops when $R(t) = 0$. In this illustration, only the first 6 slots are considered. The rational is, if we sort these 6 slots and then apply *up-the-stairs*, we got the same results which is optimal. And when we consider slot 7, since $p_7 > w$, we should raise the splitting line, which will result in $R(6) < 0$, so there is no single splitting line for the entire duration [0, 7] such that $R(7) = 0$.

considering slot 7, since it exceeds the splitting line, we would need to raise the line. However, doing so causes $R(6) < 0$ indicating that there is no single splitting line over the entire interval $[0, 7]$ that satisfies $R(7) = 0$.

Consequently, we conclude that the optimal splitting line does not necessarily remain constant over the entire duration $[0, T]$. Instead, it may change at multiple points. Let the splitting line remain constant at $\Omega_i = (w_i, b_i)$ during the interval $[\tau_{i-1}, \tau_i)$, and change at time $t = \tau_i$. The objective now is to identify all the changing points τ_i and their corresponding splitting lines Ω_i.

Before proceeding further with the algorithm, it is essential to explore certain optimality properties of the splitting lines.

Lemma 1. *Suppose in the optimal splitting lines, Ω_i is in duration $[\tau_{i-1}, \tau_i)$ and Ω_j is in duration $[\tau_{j-1}, \tau_j)$, where $i < j$. If $\Omega_i \neq \Omega_j$, then we can find a more efficient shared splitting line for both durations to transmits more data, unless infeasible solution results.*

First, since Eq. (7) is a convex function, maximizing data transmission while satisfying the energy causality constraint requires that the transmission power ρ remains equal across all sending phases. This ensures optimal energy allocation. A detailed proof of this result can be found in Ref. [46]. Second, based on Theorem 2, there must exist a single splitting line Ω that applies to all slots within the two durations. If such a shared Ω exists, it must lie between Ω_i and Ω_j, as the battery energy consumption remains consistent across these intervals.

Lemma 2. *The optimal splitting line increases only.*

Lemma 3. *The optimal splitting line increases at battery empty points.*

The proofs of Lemmas 2 and 3 follow similar reasoning to the ones presented in [16] for continuous functions. Due to space constraints, we will omit these proofs here.

We are now ready to present the algorithm. The high level idea is quite simple, we want to find the first changing point for the optimal splitting line, it is obvious that the battery should be drained at that point, we name it Energy Critical Point (ECP). After this point the same problem repeats. Hence, we repeatedly call algorithm *s-line_raising_general* to compute the feasible single splitting line that empty the battery for each time instance $t = 1, 2, \ldots, T$. Amongst all the feasible single splitting lines, we choose the smallest one and its corresponding time instance to be the first changing point.

However, this is inefficient because for each additional slot, all current slots must be re-traversed to raise the splitting line, which results in the final algorithm taking $O(T^3)$ steps to complete the scheduling. Therefore our algorithm works in iteration so as to reduce the time complexity. In iteration t, it computes the splitting line for the first t slots based on previously computed splitting line for the first $t-1$ slots. The efficiency is built on the incremental style of work in each iteration. The core idea of the proposed algorithm is illustrated in Fig. 6.

For the first one slot, we can easily find the optimal splitting line $\Omega^{(1)}$ such that $\mathrm{R}(1) = 0$. Suppose $\Omega^{(t-1)}$ is the splitting line for the first $t-1$ slots. Depending on the value of p_t, there are two cases. (1) $p_t > w^{(t-1)}$. According the definition of s-line raising system, it is a charging zone in slot t, hence $R(t) > 0$ and we need to rise the splitting line to make $R(t) = 0$. However, this will cause $R < 0$ at some time before $t-1$. We hence conclude no single splitting line is feasible and move on to the next iteration. (2) $p_t \leq w^{(t-1)}$. So the slot t should be the sending slot (or temporarily considered as a sending slot). After going through that sending process if $R(t) > 0$ then we move on to the next iteration, if $R(t) < 0$ then we decrease the splitting line until $R(t) = 0$. The detailed pseudo code for each iteration is presented in Algorithm *Energy_Critical_Point_Iteration*.

Algorithm 1: *Energy_Critical_Point_Iteration*

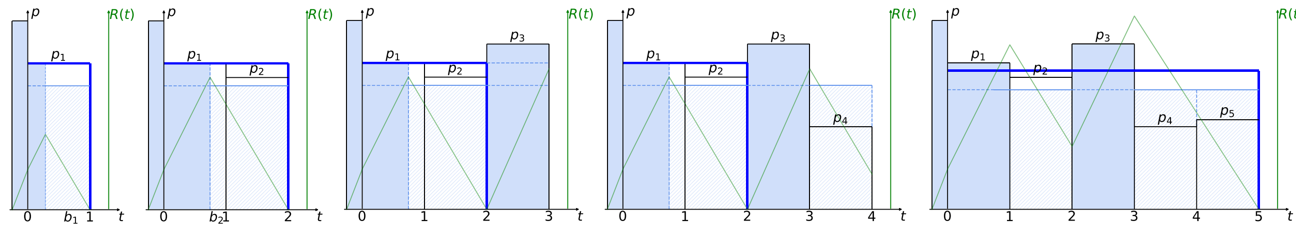
Input : starting time t_0, current time t, previous splitting line (w, b) depletes battery at τ, remain energy R (at $t-1$), max heap H that stores powers smaller than w, supply power p_t at time t

1 **if** $p_t > w$ **then**
2 $\quad R += p_t$;
3 **else**
4 $\quad R -= P_s(w)$;
5 $\quad$ Insert-MaxHeap(H, p_t);
6 **end**
7 **if** $R > 0$ **then** **return** $((w, b), \tau, R, H)$;
8 **while** *isEmpty-Heap(H)==false* **do**
9 $\quad p_x$=Top-MaxHeap(H) ; // Do not extract
10 $\quad l$=size-Heap$(H) + (\lceil t_0 \rceil - t_0)$;
11 $\quad \Delta E = (w + P_s(w)) * b - (P_s(p_x) - P_s(w)) * l$;
12 $\quad$ **if** $R + \Delta E \geq 0$ **then** break;
13 $\quad$ Extract-MaxHeap(H);
14 $\quad R += \Delta E, \quad w = p_x, \quad b = 1$;
15 **end**
16 $b_{next} = \frac{[(w+P_s(w))b-R]^+}{w+P_s(w)}$;
17 $w_{next} = P_s^{-1}(P_s(w) - \frac{[R-(w+P_s(w))b]^+}{l})$;
18 **return** $((w_{next}, b_{next}), t, 0, H)$

Output: updated splitting line (w, b) and battery depletion point τ, new remain energy R and new max heap H

For the execution of iteration t, we use R to denote the remain energy in battery, such as the area of the charging zone subtracts that of the sending zone. We must have $R \geq 0$ before slot t. Whether slot t is a charging slot or a sending slot is determined by comparing w with p_t, and R is modified accordingly in Line 1–6. After the modification, if $R > 0$ still holds, there is no feasible single constant splitting line that empty the battery at time t. If $R < 0$, we start to lower the splitting line. When the splitting line drops, it meets powers one by one in the order of from high to low, we thus need an efficient data structure to manage powers. We choose the maximum heap, because both extract the largest power and adding a power are with $O(\log(T))$ time complexity. Power p_t will be inserted into the heap if $p_t < \rho$, which will be used when splitting line (w, b) drop. The **while** in Line 8 repeatedly computes how much extra energy ΔE we can gain by dropping the splitting line to the next largest power below the current position. As long as $R + \Delta E < 0$ still holds, we continue to lower the splitting line. The **while** loop exits if lower the splitting line directly to the next power will make R go greater than 0. Then we use general formulas in Line 16 and 17 to compute the optimal position of the splitting line.

And the entire iteration process is shown in Algorithm *Varying-_Source_WPT_B*, which calculates all the changing points and the splitting lines between them.



(a) $\Omega^{(1)}$, $\tau^{(1)}$ (b) $\Omega^{(2)}$, $\tau^{(2)}$=2 (c) $\Omega^{(3)}$=$\Omega^{(2)}$, $\tau^{(3)}$=$\tau^{(2)}$ (d) $\Omega^{(4)} = \Omega^{(3)}$, $\tau^{(4)} = \tau^{(3)}$ (e) $\Omega^{(5)} = (w_5, 0)$, $\tau^{(5)} = 5$

Fig. 6. An example of the execution of our algorithm. In (a), $\Omega^{(1)} = (p_1, b_1)$ makes $R(1) = 0$, and battery is empty at $\tau^{(1)} = 1$. In (b), since $p_2 < p_1$, sending zone is enlarged in slot 2, hence splitting line drops from $\Omega^{(1)}$ to $\Omega^{(2)} = (p_1, b_2)$ to make $R(2) = 0$ and empty battery at $\tau^{(2)} = 2$. In (c), since $p_3 > p_1$, charging zone is enlarged, but battery at $t = 3$ cannot be emptied by any single splitting line. So we set $\Omega^{(3)} = \Omega^{(2)}$ which empties the battery at $\tau^{(3)} = \tau^{(2)}$. In (d), although $p_4 < p_1$ which means sending zone is enlarged, however $R(4) > 0$, so still no single splitting line can empty battery at $t = 4$, hence $\Omega^{(4)} = \Omega^{(3)}$ and $\tau^{(4)} = \tau^{(3)}$. In (e), Slot 5 is a sending zone and will result in $R(5) < 0$. We hence lower the splitting line from $\Omega^{(4)} = (p_1, b_2)$ to $\Omega^{(5)} = (w_5, 0)$ that empties battery $\tau^{(5)} = 5$.

Algorithm 2: *Varying_Source_WPT_B*

Input : the supply power vector **P** (used to provide the p_t needed by the *Energy_Critical_Point_Iteration*)

$t_0 = 1$;
while $t_0 \le T$ **do**
  $w = w_{max}$, $b = 1$, $R = 0$, init-maxHeap(H);
  for $t = t_0$ **to** T **do**
    $((w, b), \tau, R, H)$=*Energy_Critical_Point-_Iteration*$(t_0, t, (w, b), \tau, R, H)$;
    if $R > 0$ **then** continue;
    $\Omega_{min} = (w, b)$, $\tau_{min} = \tau$;
  end
  Set splitting line Ω_{min} in $[t_0, \tau_{min}]$;
  $t_0 = \tau_{min} + 1$;
end

Output: all splitting lines (w, b) throughout the process

Algorithm *Varying_Source_WPT_B* keeps calling the algorithm *Energy-_Critical_Point_Iteration* to find the smallest Ω in $t_0 = 1$ to T and record its energy depletion point τ_{min}, and after that performs the above operation for $t_0 = \tau_{min} + 1$ to T until $t_0 > T$.

Theorem 3. *The algorithm Varying_Source_WPT_B computes the optimal schedule for the offline problem in $O(T^2 \log T)$ steps.*

Proof. See Appendix B. □

An example of the execution of Algorithm *Varying_Source_WPT_B* is illustrated Figs. 5 and 6.

5. The optimal offline solution

In this section, we consider a more general case that the battery has a limited capacity. One significant difference from previous section is that too much energy will cause battery overflow and energy wastage. Let us look back at the example in Fig. 2. When charging in time slots 1–3, we may charge more energy than the battery capacity, at which point we should raise the splitting line of time slots 1–3 so that the battery fills up at $t = 3$, so that no energy is wasted.

Some optimal properties have changed and some have not. Firstly, Lemma 2 no longer holds, *e.g.,* the optimal splitting line may either increase or decrease. Secondly, Lemma 3 still holds, *e.g.,* the optimal splitting line increase only at battery empty points. Thirdly, as a supplementary, a new Lemma 4 is given below.

Lemma 4. *The optimal splitting line decreases at battery full points.*

Proof. We prove by contradiction. Suppose otherwise, the optimal splitting line decreases at time instance τ_i, but the battery is not full. More specifically, suppose $\Omega_i > \Omega_{i+1}$ while $R(\tau_i) < E_c$, where Ω_i and Ω_{i+1} are the optimal splitting lines in duration $[\tau_{i-1}, \tau_i)$ and $[\tau_i, \tau_{i+1})$ respectively, E_C is the battery capacity. By Lemma 1, we can decrease Ω_i and increase Ω_{i+1} to improve data transmission using the same amount of energy, for example, $R(\tau_{i+1})$ will not be effected. As long as $R(\tau_i) \le E_c$ after the modification, there is no violation of the battery capacity constraint, which is a contradiction. □

Now we have two types of splitting line changing point in the optimal solution. We refer the increasing (decreasing) point as ECP (BFP), stands for Energy Critical Point (Battery Full Point).

We are now ready to discuss the optimal algorithm. Algorithm *Energy_Critical_Point_Iteration* in the previous section can find one optimal ECP τ^e with the splitting line Ω^e, and if that dispatch overflows the battery before τ^e, then there must exist a BFP $\tau^f < \tau^e$ and a splitting line $\Omega^f > \Omega^e$. Starting from τ^f, the same problem repeats, so the same method applies. The key now is how to find τ^f and Ω^f. Similar to *Energy_Critical_Point_Iteration*, Algorithm *Battery_Full_Point_Iteration* is designed to incrementally computes the single splitting line that makes the battery full for a given slot. We can apply it to compute all BFPs before τ^e and select the latest one to be τ^f. Detailed pseudo code and correctness proof will be presented later.

If we firstly calculate Ω^e by iterating slots as the previous algorithm does in the pseudo code, and secondly compute Ω^f slot by slot from the first to the last to locate such an instance where $\Omega^f > \Omega^e$, then there is a computation inefficiency risk. Because each time Ω^f is computed from the first slot, and the same Ω^f would be recalculated for multiple times, making the process repetitive. However, as the slot increases, the corresponding calculated Ω^e only decreases and the computed Ω^f only increases throughout the process (explained later in Observation 1). The minimum value of Ω^e and the maximum value of Ω^f must occur in the latest slot. Therefore, if the latest slot have $\Omega^f < \Omega^e$, it implies that $\Omega^f > \Omega^e$ could not have occurred earlier. Thus, we can calculate both Ω^e and Ω^f simultaneously at each slot. This approach effectively lowers the algorithm's time complexity.

Therefore, the high level idea is, starting from the beginning, for each slot, we invoke both *Energy_Critical_Point_Iteration* and *Battery_Full-_Point_Iteration* to check whether the current time is a ECP or a BFP. We update the information, and keep tracking the latest ECP τ^e and BFP τ^f. If $\Omega^e > \Omega^f$, we continue to check the next time slot. If otherwise $\Omega^e < \Omega^f$, then $\min\{\tau^e, \tau^f\}$ must be the optimal changing point. The same problem thus repeats starting from this point.

We now discuss details by first presenting Algorithm *Battery_Full-_Point_Iteration*, which is similar to *Energy_Critical_Point_Iteration*.

Algorithm *Battery_Full_Point_Iteration* is the iteration step to compute the next battery full point. Not surprising, it is quite similar to *Energy-_Critical_Point_Iteration*. The battery remain energy R for both is updated in Line 1–6 according to the s-line raising schedule. There are two cases after the update, *e.g.,* $R < E_c$ and $R \ge E_c$. In the first case, since the splitting line cannot drop, we continue to the next iteration. In the second case, the splitting line arises until R reduces to E_C.

Algorithm 3: *Battery_Full_Point_Iteration*

Input : starting time t_0, current time t, previous splitting line (w, b) and its BFP τ, remain energy R (at $t-1$), max heap H that stores powers larger than w, supply power p_t at time t

1 **if** $p_t > w$ **then**
2 $\quad R+ = p_t$;
3 $\quad$ Insert-MinHeap(H, p_t);
4 **else**
5 $\quad R- = P_s(w)$;
6 **end**
7 **if** $R < E_c$ **then** **return** $((w, b), \tau, R, H)$;
8 **while** *isEmpty-Heap(H)==false* **do**
9 $\quad p_x$=Top-MinHeap(H) ; // Do not extract
10 $\quad l = (t - t_0)-$size-Heap(H);
11 $\quad \Delta E = (w + P_s(w)) * (1 - b) + (P_s(p_x) - P_s(w)) * l$;
12 $\quad$ **if** $R - \Delta E \le E_c$ **then** break;
13 $\quad$ Extract-MinHeap(H);
14 $\quad R- = \Delta E,\ w = p_x,\ \ \beta = 0$;
15 **end**
16 $b_{next} = 1 - \frac{[(w+P_s(w))(1-b)-R]^+}{w+P_s(w)}$;
17 $w_{next} = P_s^{-1}(P_s(w) + \frac{[R-(w+P_s(w))(1-b)]^+}{l})$;
18 **if** $w_{next} == p_t$ **then**
19 $\quad b_{next} = (R - E_c)/p_t$;
20 $\quad$ **return** $((w_{next}, b_{next}), t - b_{next}, E_c, H)$
21 **else**
22 $\quad$ **return** $((w_{next}, b_{next}), t, E_c, H)$
23 **end**

Output: updated splitting line (w, b) and its BFP τ, new remain energy R and new max heap H

Although most of *Battery_Full_Point_Iteration* is similar to *Energy_Critical_Point_Iteration*, there is a slightly difference. Since it is charging phase followed by a sending phase, it is possible that the energy fill point is not at the end of a slot. To address this problem, we add Line 18 to correct and update b_{next}. By setting $b_{next} = (R - E_c)/p_t$, the BFP is now at time $t - b_{next}$, not at slot border like other BFPs and ECPs. As a result, in the next round of iteration, although the starting time is set to be t, duration $[t - b_{next}, t)$ needs to be considered as well. Duration $[t - b_{next}, t)$ is hence called a *padding sending phase*.

Like *Energy_Critical_Point_Iteration*, Algorithm *Battery_Full_Point_Iteration* is designed to be repeatedly called to compute BFP for each slot, one after the other. The beginning padding sending phase is handled smoothly by Algorithm *Battery_Full_Point_Iteration*. If t_0 is not an integer, there must be a padding sending phase at the beginning, which belong the sending zone. In the computation of sending zone length in Line 10, its length is already included.

To find the optimal changing point efficiently, we invoke both *Energy_Critical_Point_Iteration* and *Battery_Full_Point_Iteration* for every time slot. Detailed pseudo code is presented in Algorithm *Varying_Source_WPT*.

Algorithm *Varying_Source_WPT* works in iteration. Each *while* loop in Line 2, namely the outer *while*, computes the next optimal changing point and the corresponding splitting line. At the beginning of each loop, t_0 is the starting time, while E is the energy remain in battery. Initially, $t_0 = 0$ and $E = E_I$ before the first loop. In each iteration, the *while* loop in Line 7, namely the inner *while* loop incrementally updates the ECP and BFP by algorithm *Energy_Critical_Point_Iteration* and *Battery_Full_Point_Iteration* respectively, for slots after $\lceil t_0 \rceil$ one by one. We initialize the splitting line for ECP $\Omega^e = (w_{max}, 1)$, and splitting line for BFP $\Omega^f = (0, 0)$, before the first invocation of *Energy_Critical_Point_Iteration* and *Battery_Full_Point_Iteration*. Duration the execution of inner *while*, we have the follow observation.

Observation 1. *As the inner* **while** *loop repeats,* $\Omega^e = (w^e, b^e)$ *returned by Energy_Critical_Point_Iteration in Line 8 keeps decreasing;* $\Omega^f = (w^f, b^f)$ *returned by Battery_Full_Point_Iteration in Line 9 keeps increasing.*

Algorithm 4: *Varying_Source_WPT*

Input : the supply power vector $\mathbf{P}$ (used to provide the p_t needed by the *Energy_Critical_Point_Iteration* and *Battery_Full_Point_Iteration*)

1 $t_0 = 1,\ E = E_I,\ p_{T+1} = \infty$;
2 **while** $t_0 \le T$ **do**
3 $\quad w^e = w_{max},\ b^e = 1,\ \tau^e = t_0,\ R^e = E$;
4 $\quad w^f = 0,\ b^f = 0,\ \tau^f = t_0,\ R^f = E$;
5 $\quad$ init-MaxHeap(H^e), init-MinHeap(H^f);
6 $\quad t = \lceil t_0 \rceil$;
7 $\quad$ **while** $(w^e, b^e) > (w^f, b^f)$ **do**
8 $\quad\quad ((w^e,b^e),\tau^e,R^e,H^e)$=*Energy_Critical_Point_Iteration*$(t_0,t,(w^e,b^e),\tau^e,R^e,H^e)$;
9 $\quad\quad ((w^f,b^f),\tau^f,R^f,H^f)$=*Battery_Full_Point_Iteration*$(t_0,t,(w^f,b^f),\tau^f,R^f,\ H^f)$;
10 $\quad\quad t = t + 1$;
11 $\quad$ **end**
12 $\quad$ **if** $\tau^f < \tau^e$ **then**
13 $\quad\quad$ Set splitting line (w^f, b^f) in $[t_0, \tau^f)$;
14 $\quad\quad t_0 = \tau^f + 1,\ E = E_c$;
15 $\quad$ **else if** $\tau^e < \tau^f$ **then**
16 $\quad\quad$ Set splitting line (w^e, b^e) in $[t_0, \tau^e)$;
17 $\quad\quad t_0 = \tau^e + 1,\ E = 0$;
18 $\quad$ **end**
19 **end**
20 **return** S

Output: all splitting lines (w, b) throughout the process

The first half can be observed in the previous section, the second half is symmetry. The inner *while* continues as long as $\Omega^e > \Omega^f$. When the loop terminates at $\Omega^e < \Omega^f$, there are two possibilities, $\tau^e < \tau^f = t$ or $\tau^f < \tau^e = t$, in either case, $\min\{\tau^e, \tau^f\}$ is the optimal changing point. Since we set $p_{T+1} = \infty$, whenever $t = T + 1$, Ω^f will be updated by *Battery_Full_Point_Iteration* to infinity, thus τ^e will be selected to empty the battery. As a result, the optimal schedule will eventually empty the battery at the end of slot T.

Note that our algorithms are designed to handle the padding phase case. This includes when a changing point is not at a slot border. Algorithm *Battery_Full_Point_Iteration* may return a BFP τ^f not at a slot border. Such BFP can become the next changing point, therefore, t_0 may not be an integer in the next iteration. We handle this case with *Varying_Source_WPT* in Line 6, which sets the starting time slot to $\lceil t_0 \rceil$. During the computation of ECP in *Energy_Critical_Point_Iteration*, the sending zone length is calculated by including such padding phase in Line 10. In *Battery_Full_Point_Iteration*, the computation of the sending zone length in Line 10 naturally includes such padding phase.

Theorem 4. *Algorithm Varying_Source_WPT computes the optimal schedule for the max-T adaptive HTT-scheduling problem in* $O(T^2 \log T)$ *steps.*

Proof. See Appendix C. ☐

Example of the execution of the *Varying_Source_WPT* is illustrated in Fig. 7. And Fig. 8 demonstrates the optimal solution for a given wireless supply power $\mathbf{p}$ and considering the battery capacity limitation.

6. Online algorithm and simulations

In this is section, we study the online max-T adaptive *HTT-scheduling* problem, where no future information about the wireless supply power is known. We propose a heuristic algorithm, namely *s-line guided online algorithm*, which is based on optimal properties for the offline problem. We then evaluate the performance of our online heuristic algorithm by comparisons with the optimal offline solutions.

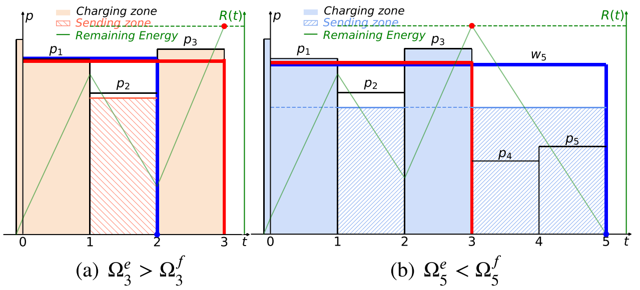


Fig. 7. $\Omega_3^e = \Omega_2^e$ is from Fig. 6(c), while Ω_3^f fulls the battery. Since $\Omega_3^e > \Omega_3^f$, we continues. Ω_5^e is from Fig. 6(e). Because $\Omega_5^f = \Omega_3^f$, and since $\Omega_5^e < \Omega_3^f$, Ω_3^f is selected to be the first splitting line and $\tau_3^f = 3$ is BFP.

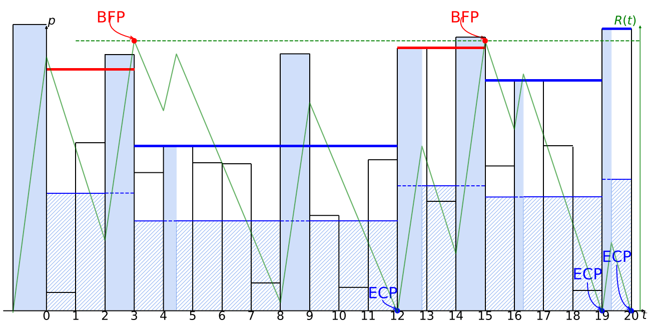


Fig. 8. A result of our final algorithm for the $T = 20$, $E_c = 50$, $E_I = 47$, $\rho_{max} = 47$ case, with two BFPs and three ECPs in the result.

6.1. Online algorithm

The core idea of *s-line guided online algorithm* is to utilize the average power supply from previous time periods to predict future power availability. This approach helps guide the scheduling process by applying the s-line-raising system to calculate the optimal schedule for the current time period, ensuring efficient energy usage and maximizing transmission while maintaining the energy causality constraint.

More specifically, at the beginning of slot t, the battery energy is assumed to be E_r and the supply power p for the current slot is known, while the supply powers for future slots remain unknown. For the slots after t, we assume the wireless supply power is the historical average $\bar{p}$. Given the initial battery energy E_r and the supply powers $\mathbf{p} = \{p, \bar{p}, \bar{p}, \dots\}$, the s-line raising system becomes straightforward, and calculating the optimal schedule is relatively simple. We explore the first optimal splitting line change in the following three cases.

(1). If $E_r - P_s(p) \leq 0$, we must have the end of current slot be a ECP. Suppose the corresponding splitting line $\Omega^e_{cur} = (p, \beta)$, where $0 < \beta < 1$. If $p < \bar{p}$, then splitting line Ω^e_{cur} cannot be lower down by *Energy_Critical_Point_Iteration* in subsequence invocation. So $\Omega^e_{cur} = (p, \beta)$ is the optimal splitting line that depletes battery at t. We compute β accordingly, $E_r + p(1-\beta) = P_s(p)\beta$, so $\beta = \frac{E_r+p}{p+P_s(p)}$. Meanwhile $\rho = P_s(p)$. If $t = T$, *e.g.,* the current slot is the last one, then we must have t be a ECP as well. So the calculation of ρ and β is the similar except a small correction: $\beta = \min\{1, \frac{E_r+p}{p+P_s(p)}\}$.

(2). If $E_r + p \geq E_c$, we must have a BFP before the end of the current slot. So similarly to (1) we can conclude that $E_r + p(1-\beta) = E_c$, so $\beta = \frac{E_r+p-E_c}{p}$. Meanwhile $\rho = P_s(p)$.

(3). Previously we have considered the case when t is the optimal splitting line change point. In other cases if the decision to schedule for a pure charging/transmitting time slot is based simply on the relationship between the power of the current time slot p and $\bar{p}$ is inflexible, so we modify the schedule for the current time slot as follows. This modification will not change transmission power, such as $\rho = P_s(\bar{p})$. It changes the sending phase length to be β. Compare to a pure charging slot, such sending phase costs $(p + P_s(\bar{p}))\beta$ energy loss, including $p\beta$ energy less charged and $P_s(\bar{p})\beta$ energy consumed. In slot t, an average of $\bar{p}$ energy can be charged in the battery, however, because of the sending phases, now $(p + P_s(\bar{p}))\beta$ energy is reduced from the battery. We want them equal, *e.g.,* $(p+P_s(\bar{p}))\beta = \bar{p}$, because the energy charged and energy consumed will equal in a long run. So, $\beta = \min\{1, \frac{\bar{p}}{p+P_s(\bar{p})}\}$. Meanwhile, to guarantee energy not depleted before t, we require $\beta \leq \frac{E_r+p}{p+P_s(p)}$, hence $\beta = \min\{1, \frac{\bar{p}}{p+P_s(\bar{p})}, \frac{E_r+p}{p+P_s(\bar{p})}\}$.

The details of the algorithm are shown in the pseudo code.

Algorithm 5: *s-line_guided_online*

Input : remaining energy E_r, current time t, current supply power p, historical supply power vector $\mathbf{P}$ (used to compute average receive power $\bar{p}$)

Compute the average receive power as $\bar{p}$;
if $p < \bar{p}$ **and** $E_r - P_s(p) \leq 0$ **or** $t = T$ **then**
  $\beta = \min\{1, \frac{E_r+p}{p+P_s(p)}\}$;
  $\rho = P_s(p)$;
else if $p > \bar{p}$ **and** $E_r + p \geq E_c$ **then**
  $\beta = \frac{E_r+p-E_c}{p}$;
  $\rho = P_s(p)$;
else
  $\beta = \min\{1, \frac{\bar{p}}{p+P_s(\bar{p})}, \frac{E_r+p}{p+P_s(\bar{p})}\}$;
  $\rho = P_s(\bar{p})$;
end
$D = \beta * \log(1 + \rho)$;
$E_r \leftarrow \min\{E_r + p * (1 - \beta_t), E_c\} - \rho * \beta$;

Output : data throughput D at slot t, remaining energy E_r at the end of slot t

---

If neither the battery becomes fully charged during slot t nor becomes empty due to sending in slot t, then the larger the supply power p, the smaller the value of β_t. If the future supply power is indeed $\bar{p}$, the historical average, then our algorithm will produce the optimal schedule for the given conditions.

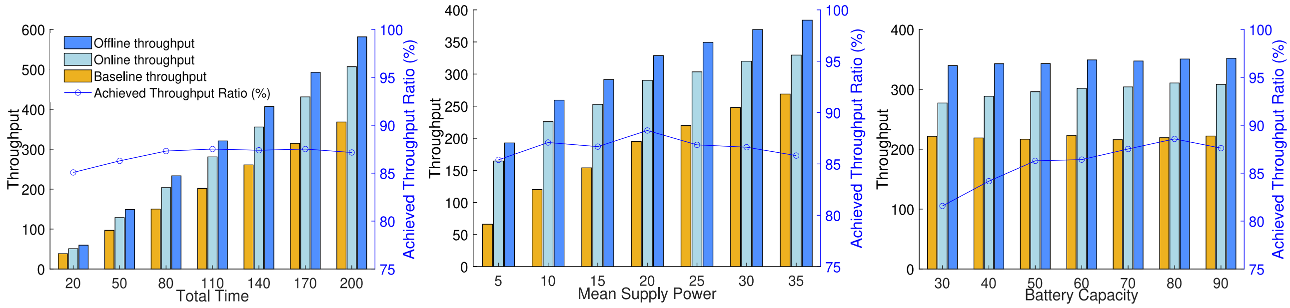
6.2. Simulations

In this subsection, we implement the proposed *s-line_guided_online* algorithm and study its efficiency. Since there are no other algorithms in the literature that study the same throughput-maximizing HTT scheduling problem under dynamic wireless supply power, we additionally design a **baseline algorithm**. The idea of the baseline algorithm is very simple, first using the maximum transmission power to deplete the energy in battery. Subsequently, in each time slot, it splits the slot into two phases — the first phase is dedicated to energy harvesting, while the second phase uses the harvested energy for data transmission, again at the maximum transmission power, ensuring that all the energy harvested in the first phase is fully consumed. We compare the performance of the online algorithm with the baseline algorithm and the optimal offline solution.

In simulations, a total of $T = 120$ time slots are considered. In each slot, the wireless supply power is assumed to be a random variable following the Rayleigh distribution with scale parameter $\sigma = \frac{25}{\sqrt{\pi/2}}$, resulting in a mean value of $\hat{p} = 25$. The battery capacity is assumed to be $E_c = 60$ and an initial charge of $E_I = 25$. The maximum transmission power is assumed to be $\rho_{max} = 47$. In this simulation, we change the total time slot T, the mean supply power $\hat{p}$ and the battery capacity E_C, one at a time, to evaluate their impact on the algorithm performance.

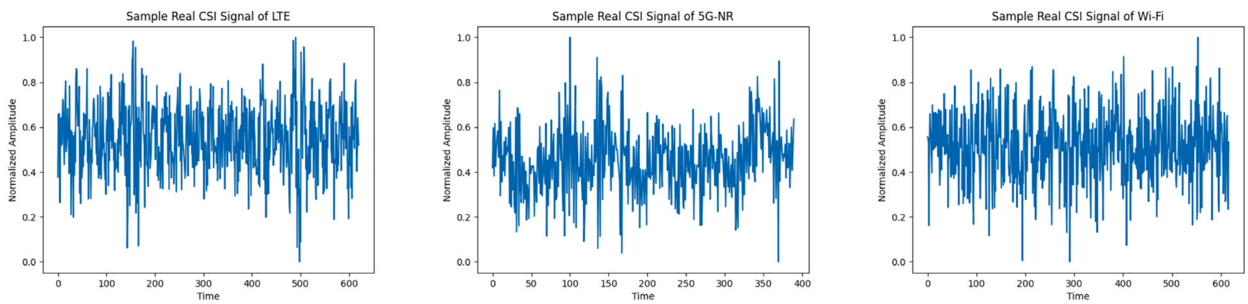
Each value shown in figures of this section is the mean value of simulation results from 20 random instances, and in each instance, a total of T supply powers are generated according to the above model.

In Fig. 9, we show the throughput of the baseline algorithm, the online algorithm versus the offline optimal solution and its ratio.



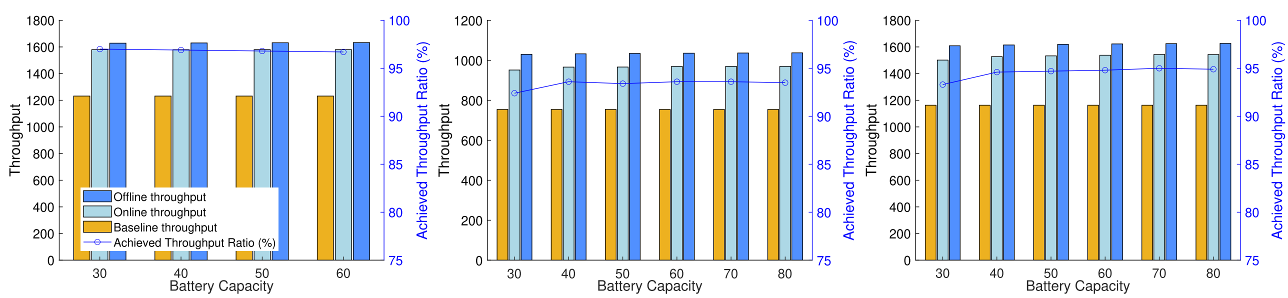
(a) (b) (c)

Fig. 9. Comparison of throughput achieved by baseline algorithm, throughput achieved by online algorithm and offline optimal throughput. The default setting is total time $T = 120$, mean wireless supply power $\hat{p} = 25$ and battery capacity $E_c = 60$. In (a), the total time changes from 20 to 200 with step 30. In (b), the mean wireless supply power changes from 5 to 35 with step 5. In (c), the battery capacity changes from 30 to 90 with step 10.



(a) (b) (c)

Fig. 10. Sample from Real-world Wireless Communication Dataset [47] released by IEEE. (a) LTE signal data with 621 time slots, (b) 5G-NR signal data with 390 time slots, and (c) Wi-Fi signal data with 617 time slots. All data are normalized and displayed.



(a) (b) (c)

Fig. 11. Comparison of throughput achieved by the baseline algorithm, throughput achieved by the online algorithm, and offline optimal throughput under real-world data. The default settings are an initial power $E_I = 25$, a charging baseline value $\hat{p} = 50$, and a maximum transmission power $\rho_{max} = 47$. LTE signaling data is used in (a), and the battery capacity E_C ranges from 30 to 60, in steps of 10. 5G-NR signaling data is used in (b), and the battery capacity E_C ranges from 30 to 80, in steps of 10. Wi-Fi signaling data is used in (c), and the battery capacity E_C ranges from 30 to 80 in steps of 10.

In Fig. 9(a), the total time slot T varies from 20 to 200. The performance of our online algorithm improves with larger total time slot, as it relies on historical average supply power to predict future power more accurately. However, even the total time slots is only 20, the achieved ratio is over 85%, and with larger total time, the ratio is around 87%.

We can see from Fig. 9(b) that the efficiency of the proposed algorithm increases as the average supply power $\hat{p}$ increases. This is because when $\hat{p}$ varies, it takes less time to charge the same amount of energy, and the reduced charging time of our heuristic algorithm is not necessarily optimal, so the efficiency fluctuates. However, the realized rate remains above 85% in all test cases.

In Fig. 9(c), the battery capacity E_C varies from 30 to 90. We can see that for larger battery capacities, our online algorithm performs better. This is due to the fact that with smaller battery capacities, data is often forced to be transferred due to a full charge. However, in most cases the achieved ratio is greater than 85%.

6.3. Real world scenario

To evaluate the performance of our online algorithm under realistic conditions, we utilized data from the Real-world Wireless Communication Dataset [47] released by IEEE. Specifically, we randomly selected sample segments from LTE, 5G, and Wi-Fi signal traces, as illustrated in Fig. 10. These traces were used as input under a consistent set of default parameters: an initial energy level $E_I = 25$, a charging baseline value $\hat{p} = 50$, and a maximum transmission power $\rho_{\max} = 47$. In each time slot, the charging power is calculated as the product of $\hat{p}$ and the normalized signal value. We then varied the battery capacity E_C to test the algorithm's performance. The corresponding results are presented in Fig. 11.

As shown in the results, our online algorithm demonstrates strong performance across all three types of signals. Under all tested battery capacity conditions, the achieved ratio consistently exceeds 90%, reaching close to 95% in many cases. Moreover, the performance of the algorithm improves with increasing battery capacity, which aligns well with our simulation results. These observations indicate the potential of our algorithm for practical deployment in real-world wireless power transfer scenarios.

7. Conclusion and future work

In this paper, we first formulate the throughput maximization HTT-scheduling problem for dynamic wireless power supply. We then introduced the basic property $sOPT$ power, the concept of the *splitting line*, and the *s-line raising system*. Next, we investigated the scenario with unlimited battery size and monotonically decreasing charging power, observing some optimality properties and designing the optimal offline algorithm based on the splitting line and the s-line raising system. These properties and the algorithm were then extended to the general scenario with limited battery capacity. Finally, an online heuristic algorithm was proposed and evaluated through simulations.

Next, we plan to integrate the algorithm into Internet of Things (IoT) devices, providing a novel power transfer solution for small-scale IoT systems. This integration will require addressing real-world challenges and limitations, such as hardware constraints: the limited processing power and storage capacity of IoT devices; synchronization issues: ensuring accurate timing for power harvesting and data transmission; the impact of environmental conditions: noise and obstacles that may affect the actual power harvesting efficiency of the RF system; the nonlinear behavior of batteries in practice: reduced charging efficiency at higher states of charge or when the battery is near full capacity.

CRediT authorship contribution statement

Fangyu Zhou: Writing – original draft, Software, Data curation. **Feng Shan:** Writing – review & editing, Writing – original draft, Project administration. **Weiwei Wu:** Visualization. **Runqun Xiong:** Visualization, Funding acquisition, Conceptualization. **Junzhou Luo:** Supervision, Methodology.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Feng Shan reports financial support was provided by National Natural Science Foundation of China. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the National Natural Science Foundation of China Grants 62472090, 62172091, the National Natural Science Foundation of Jiangsu Grants BK20242026, BK20230024, Jiangsu Provincial Key Laboratory of Network and Information Security Grant BM2003201, and Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China Grant 93K-9.

Appendix A. Proof of Theorem 2

It is a brief proof, we prove by contradiction. Assuming that there exists $[t_i, t_i+\sigma]$ time $p > w_{opt}$ in the optimal solution and this time is the transmitting phase, and $[t_j, t_j+\sigma]$ time $p < w_{opt}$, which is the charging phase, then we can always exchange these two phases thus in the case that the throughput D is the same. The residual energy $R(T) > 0$ is clearly not an optimal solution, which is a contradiction. The proof of the existence of $b_{opt} \in [0, 1]$ is trivial, since any slot is defined to be with charging phase length $b \in [0, 1]$.

Appendix B. Proof of Theorem 3

In every iteration of the **while** loop, the problem repeats: starting from t_0, find the next optimal changing point and the corresponding splitting line. Therefore, we only need to show that in the first loop, where $t_0 = 1$, the computed changing time τ_{min} and splitting line Ω_{min} (Line 9) are the first optimal changing point and optimal splitting line respectively.

Suppose, instead, the first optimal changing point $\tau_1^{opt} \neq \tau_{min}$, we then have the following two cases. (1) $\tau_1^{opt} > \tau_{min}$. Since the **for** loop has already checked in Line 5–6, there is no feasible splitting line that would make $R(t) = 0$ in all $t > \tau_{min}$. This is impossible. (2) $\tau_1^{opt} < \tau_{min}$. Since the splitting line only decreases in Algorithm *Energy_Critical_Point-_Iteration*, we must have $\Omega_1^{opt} > \Omega_{min}$. Since Lemma 2, splitting line $\Omega_i^{opt} (i > 1)$ have $\Omega_i^{opt} > \Omega_1^{opt} > \Omega_{min}$, which leads to the duration $[\tau_1^{opt} + 1, \tau_{min}]$ in which the remaining power $R < 0$ occurs.

During the execution of *Energy_Critical_Point_Iteration*, the dominate computation are two heap operations, both take $O(\log T)$ steps at most. In each **while** loop iteration, one changing point is found by the execution of **for** loop, which has at most T iterations. During these iterations, at most T Insert-MaxHeap operation takes place. Because each powers inserted into the Heap can be extracted no more than once, there are at most T Extrac-MinHeap operations too. Hence, to find one changing point, it needs at most $O(T \log T)$ steps. Since there are no more than T changing points, the algorithm can be completed in $O(T^2 \log T)$ steps.

Appendix C. Proof of Theorem 4

Similarly to Theorem 3, since our algorithm is iterative, we only need to show that we can find the next optimal changing point, given the current one and the battery energy.

In each iteration of the outer **while**, there is a inner **while** and a **if** statement setting the next changing point. We now show that when the inner **while** terminates at condition $(w^e, b^e) < (w^f, b^f)$, the **if** statement considers all possibility, *e.g.,* $\tau^e \neq \tau^f$. Suppose at time $t-1$, we have $\Omega^{e'} > \Omega^{f'}$, while in the next loop, *e.g.,* time t, we have $\Omega^e < \Omega^f$, thus exits the loop. Since at time t the splitting line that empties the battery must be greater than the splitting line that fulls the battery, we must have $\Omega^e > \Omega^f$ when $\tau^e = \tau^f = t$. So there are only two cases for updates in the loop t. **(1)**. $\Omega^{e'}$ is updated to Ω^e, while $\Omega^{f'}$ is not. We must have $\tau^{f'} = \tau^f < \tau^e = t$. **(2)**. $\Omega^{f'}$ is updated to Ω^f, while $\Omega^{e'}$ is not. We must have $\tau^{e'} = \tau^e < \tau^f = t$.

We only show the first branch, such as case (1), is correct, because the other case is similar and thus left to the readers.

We show τ^f is the optimal changing point and (w^f, b^f) is the optimal splitting line (Line 13–14) by contradiction. Suppose τ_{opt} is the optimal changing point, but $\tau_{opt} \neq \tau^f$. Since τ_{opt} must be a ECP or a BFP according to Lemmas 3 and 4, we consider the following cases.

(a). $\tau_{opt} = \tau^{e'} \geq \tau^e$. This is impossible, because $\Omega^{e'} \leq \Omega^e < \Omega^f$ and battery is full at τ^f, so splitting line in $[t_0, \tau^f]$ cannot drop any lower, let alone to $\Omega^{e'}$. **(b)**. $\tau_{opt} = \tau^{f'} > \tau^e$. This is impossible, because we have $\Omega^e < \Omega^f < \Omega^{f'}$ and battery is empty at τ^e, so splitting line in $[t_0, \tau^e]$ cannot raise any higher, let alone to $\Omega^{f'}$. **(c)**. $\tau^f < \tau_{opt} = \tau^{f'} < \tau^e$.

This is impossible, because at time $\tau^{f'}$, the latest BFP must have been updated to $\tau^{f'}$ by the algorithm, contradicting to the fact BFP $\tau^f < \tau^{f'}$ is the latest BFP at time $t > \tau^{f'}$. **(d).** $\tau^f < \tau_{opt} = \tau^{e'} < \tau^e$. Since $\Omega^e < \Omega^{e'}$, in duration $[\tau^{e'}, \tau^e)$, there must be a sub-duration a such that its splitting line $\Omega^a < \Omega^e$, because τ^e is a energy critical point and Ω^e empties the battery at τ^e. Then splitting line $\Omega^{e'}$ in duration $[t_0, \tau^{e'})$ and splitting line Ω^a in duration a can be equalized by Lemma 1, contradict to $\tau_{opt} = \tau^{e'}$. **(e).** $\tau_{opt} < \tau^f$. Since τ^f is a battery full point, there must be a sub-duration in $[t_0, \tau^f]$ with splitting line greater than Ω^f. For otherwise, energy will overflow at time τ^f. Suppose $[t_a, t_b]$ is the first of such sub-duration, such that its splitting line $\Omega^{ab} > \Omega^f$. Either $t_a \neq t_0$ or $t_b \neq \tau^f$ holds, for otherwise there is no changing point before τ^f. If $t_a \neq t_0$, then there must be a ECP $\tau^{e'} \leq t_a$, such that $\Omega^{e'} < \Omega^f$. According to the algorithm, the inner **while** loop must have already stopped at τ^f. If $t_a = t_0$ and $t_b \neq \tau^f$, then t_b is a BFP, its splitting line Ω^{ab} cannot drop to Ω^f.

Battery_Full_Point_Iteration and *Energy_Critical_Point_Iteration* are similar in structure, so both take at most $O(\log T)$ steps. And the algorithm approximately performs an additional *Battery_Full_Point_Iteration* in each inner loop of the Algorithm *Varying_Source_WPT_B*, so the algorithm can be completed in $O(T^2 \log T)$ steps.

References

[1] Q. Lin, J. Su, M. Chen, Competitive online age-of-information optimization for energy harvesting systems, in: IEEE INFOCOM 2024-IEEE Conference on Computer Communications, IEEE, 2024, pp. 1–10.

[2] P. Zhou, C. Wang, Y. Yang, Design and optimization of electric autonomous vehicles with renewable energy source for smart cities, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1399–1408.

[3] R. Guida, E. Demirors, N. Dave, T. Melodia, Underwater ultrasonic wireless power transfer: A battery-less platform for the internet of underwater things, IEEE Trans. Mob. Comput. 21 (5) (2020) 1861–1873.

[4] J. Li, Y. Meng, L. Zhang, G. Chen, Y. Tian, H. Zhu, X.S. Shen, MagFingerprint: A magnetic based device fingerprinting in wireless charging, in: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, IEEE, 2023, pp. 1–10.

[5] W. Zhou, H. Zhou, X. Cui, X. Wang, X. Wang, Z. Liu, Roland: Robust in-band parallel communication for magnetic MIMO wireless power transfer system, in: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, IEEE, 2023, pp. 1–10.

[6] W. Zhou, X. Wang, H. Zhou, S. Jiang, Z. Liu, Y. Ji, Safety guaranteed power-delivered-to-load maximization for magnetic wireless power transfer, in: IEEE INFOCOM 2024-IEEE Conference on Computer Communications, IEEE, 2024, pp. 1–10.

[7] M. Ren, D. Wu, J. Xue, W. Xu, J. Peng, T. Liu, Utilizing the neglected back lobe for mobile charging, in: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, IEEE, 2023, pp. 1–10.

[8] Y. Ma, D. Wu, M. Ren, J. Peng, J. Yang, T. Liu, Concurrent charging with wave interference, in: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, IEEE, 2023, pp. 1–10.

[9] T. Liu, Y. Ma, M. Ren, J. Yang, J. Peng, J. Yang, D. Wu, Concurrent charging with wave interference for multiple chargers, IEEE/ACM Trans. Netw. (2024).

[10] Y. Ma, D. Wu, J. Gao, W. Sun, J. Yang, T. Liu, Dynamic power distribution controlling for directional chargers, in: IEEE INFOCOM 2024-IEEE Conference on Computer Communications, IEEE, 2024, pp. 1–10.

[11] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, J.R. Smith, Ambient backscatter: Wireless communication out of thin air, ACM SIGCOMM Comput. Commun. Rev. 43 (4) (2013) 39–50.

[12] S. Mohanti, E. Bozkaya, M.Y. Naderi, B. Canberk, G. Secinti, K.R. Chowdhury, WiFED mobile: WiFi friendly energy delivery with mobile distributed beamforming, IEEE/ACM Trans. Netw. 29 (3) (2021) 1362–1375.

[13] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, J. Smith, Inter-technology backscatter: Towards internet connectivity for implanted devices, in: Proceedings of the 2016 ACM SIGCOMM Conference, 2016, pp. 356–369.

[14] Powercast Corporation, Powercast, 2024, https://www.powercastco.com.

[15] University of Washington, Wireless identification and sensing platform (WISP), 2024, https://sites.google.com/uw.edu/wisp-wiki/home.

[16] F. Shan, J. Luo, Q. Jin, L. Cao, W. Wu, Z. Ling, F. Dong, Optimal harvest-then-transmit scheduling for throughput maximization in time-varying RF powered systems, IEEE J. Sel. Areas Commun. Early Access.

[17] H. Ju, R. Zhang, Throughput maximization in wireless powered communication networks, IEEE Trans. Wirel. Commun. 13 (1) (2013) 418–428.

[18] F. Shan, J. Luo, W. Wu, X. Shen, Optimal wireless power transfer scheduling for delay minimization, in: IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.

[19] F. Shan, J. Luo, W. Wu, X. Shen, Delay minimization for data transmission in wireless power transfer systems, IEEE J. Sel. Areas Commun. 37 (2) (2018) 298–312.

[20] T.A. Zewde, M.C. Gursoy, Optimal resource allocation for energy-harvesting communication networks under statistical QoS constraints, IEEE J. Sel. Areas Commun. 37 (2) (2018) 313–326.

[21] K.-G. Nguyen, Q.-D. Vu, L.-N. Tran, M. Juntti, Energy efficiency fairness for multi-pair wireless-powered relaying systems, IEEE J. Sel. Areas Commun. 37 (2) (2018) 357–373.

[22] S. Li, S. He, K. Hu, L. Fu, S. Chen, J. Chen, Operation state scheduling towards optimal network utility in RF-powered internet of things, IEEE Trans. Mob. Comput. 20 (11) (2020) 3117–3130.

[23] X. Song, K.-W. Chin, Maximizing packets collection in wireless powered IoT networks with charge-or-data time slots, IEEE Trans. Cogn. Commun. Netw. 9 (4) (2023) 1067–1079.

[24] C. Jin, F. Hu, Z. Ling, Z. Mao, Z. Chang, C. Li, Transmission optimization and resource allocation for wireless powered dense vehicle area network with energy recycling, IEEE Trans. Veh. Technol. 71 (11) (2022) 12291–12303.

[25] M.A. Qureshi, C. Tekin, Online Bayesian learning for rate selection in millimeter wave cognitive radio networks, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1449–1458.

[26] K. Zheng, G. Jiang, X. Liu, K. Chi, X. Yao, J. Liu, DRL-based offloading for computation delay minimization in wireless-powered multi-access edge computing, IEEE Trans. Commun. 71 (3) (2023) 1755–1770.

[27] L. Zhao, Y. Yao, J. Guo, Q. Zuo, V.C. Leung, Collaborative computation offloading and wireless charging scheduling in multi-UAV-assisted MEC networks: A TD3-based approach, Comput. Netw. (2024) 110615.

[28] Y. Feng, X. Zhang, R. Jia, F. Lin, J. Lu, Z. Zheng, M. Li, Intelligent trajectory design for mobile energy harvesting and data transmission, IEEE Internet Things J. 10 (1) (2022) 403–416.

[29] I. Valiulahi, A. Salem, C. Masouros, Resource allocation policies for battery constrained energy harvesting communication systems with co-channel interference, IEEE Trans. Commun. 70 (12) (2022) 8342–8354.

[30] Y. Xiao, Q. Yuan, K. Jiang, X. Jin, J. He, L. Zhang, C.-w. Lin, Local-global temporal difference learning for satellite video super-resolution, IEEE Trans. Circuits Syst. Video Technol. 34 (4) (2023) 2789–2802.

[31] C. Thakur, S. Chattopadhyay, Performance analysis of RF-powered multi-device diamond relay IoT network using adaptive NOMA, Arab. J. Sci. Eng. 49 (5) (2024) 7163–7177.

[32] V. Potnis Kulkarni, R.D. Joshi, Modeling and performance analysis of LBT-based RF-powered NR-u network for IoT, Sensors 24 (16) (2024) 5369.

[33] H. Lin, M.A. Kishk, M.-S. Alouini, Performance evaluation of RF-powered IoT in rural areas: The wireless power digital divide, IEEE Trans. Green Commun. Netw. 8 (2) (2024) 716–729.

[34] M. Zahari, D. El Pebrian, S. Shamsi, H. Sulaiman, S. Mustaffha, N. Shairi, Powering the future of farming: RF energy harvesting for environmental sustainability, in: IOP Conference Series: Earth and Environmental Science, 1397, (1) IOP Publishing, 2024, 012022.

[35] R. Colella, L. Catarinucci, G. Grassi, Battery-less RF-powered circuits for non-contact voltage monitoring of electric systems: Circuit modeling and SPICE analysis, Int. J. Circuit Theory Appl. 53 (2) (2025) 840–860.

[36] J. Van Mulders, G. Callebaut, Designing RF-powered battery-less electronic shelf labels with COTS components, 2025, arXiv preprint arXiv:2503.13009.

[37] M. Sapurov, A. Baskys, K. Slivka, V. Bleizgys, K. Stasys, R. Simniskis, V. Zlosnikas, K. Urbonas, A. Dervinis, S. Paulikas, et al., RF power saving system for smart homes, Heliyon 10 (21) (2024).

[38] L. Shi, R.Q. Hu, J. Gunther, Y. Ye, H. Zhang, Energy efficiency for RF-powered backscatter networks using HTT protocol, IEEE Trans. Veh. Technol. 69 (11) (2020) 13932–13936.

[39] R. Kishore, S. Gurugopinath, P.C. Sofotasios, S. Muhaidat, N. Al-Dhahir, Opportunistic ambient backscatter communication in RF-powered cognitive radio networks, IEEE Trans. Cogn. Commun. Netw. 5 (2) (2019) 413–426.

[40] Y. Wang, Q. Wang, Q. Wang, Z. Zheng, HRL-based access control for wireless communications with energy harvesting, IEEE Trans. Autom. Sci. Eng. 21 (1) (2023) 1000–1011.

[41] W. Zhang, C. Pan, T. Liu, J. Zhang, M. Sookhak, M. Xie, Intelligent networking for energy harvesting powered IoT systems, ACM Trans. Sens. Netw. 20 (2) (2024) 1–31.

[42] H. Kim, T. Cho, J. Lee, W. Shin, H.V. Poor, Optimized shallow neural networks for sum-rate maximization in energy harvesting downlink multiuser NOMA systems, IEEE J. Sel. Areas Commun. 39 (4) (2020) 982–997.

[43] C. Xu, X. Zhang, H.H. Yang, X. Wang, N. Pappas, D. Niyato, T.Q. Quek, Optimal status updates for minimizing age of correlated information in IoT networks with energy harvesting sensors, IEEE Trans. Mob. Comput. 23 (6) (2024) 6848–6864.
[44] R. Mohammadi, Z. Shirmohammadi, RLS2: An energy efficient reinforcement learning-based sleep scheduling for energy harvesting WBANs, Comput. Netw. 229 (2023) 109781.
[45] J. Yang, S. Ulukus, Optimal packet scheduling in an energy harvesting communication system, IEEE Trans. Commun. 60 (1) (2011) 220–230.
[46] F. Shan, J. Luo, W. Wu, M. Li, X. Shen, Discrete rate scheduling for packets with individual deadlines in energy harvesting systems, IEEE J. Sel. Areas Commun. 33 (3) (2015) 438–451.
[47] S. Subray, Real-world Wireless Communication Dataset: IEEE 802.11ax, LTE, and 5G-NR Signals, IEEE Dataport, 2023, Dataset.