2023

Data Collection and Information Freshness in Energy Harvesting Networks

Lei Zhang

Data Collection and Information Freshness in Energy Harvesting Networks

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Lei Zhang

Bachelor of Engineering (Telecommunications)

School of Electrical, Computer and Telecommunications Engineering

March 2023

Statement of Originality

I, Lei Zhang, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institutions.

Signed

Lei Zhang

March, 2023

Abstract

An Internet of Things (IoT) network consists of multiple devices with sensor(s), and one or more access points or gateways. These devices monitor and sample targets, such as valuable assets, before transmitting their samples to an access point or the cloud for storage or/and analysis. A critical issue is that devices have limited energy, which constrains their operational lifetime. To this end, researchers have proposed various solutions to extend the lifetime of devices. A popular solution involves optimizing the duty cycle of devices; equivalently, the ratio of their active and inactive/sleep time. Another solution is to employ energy harvesting technologies. Specifically, devices rely on one or more energy sources such as wind, solar or Radio Frequency (RF) signals to power their operations. Apart from energy, another fundamental problem is the limited spectrum shared by devices. This means they must take turns to transmit to a gateway. Equivalently, they need a transmission schedule that determines when they transmit their samples to a gateway.

To this end, this thesis addresses three novel device/sensor selection problems. It first aims to determine the best devices to transmit in each time slot in an RF Energy-Harvesting Wireless Sensor Network (EH-WSN) in order to maximize throughput or sum-rate. Briefly, a Hybrid Access Point (HAP) is responsible for charging devices via downlink RF energy transfer. After that, the HAP selects a subset of devices to transmit their data. A key challenge is that the HAP has

neither channel state information nor energy level information of device. In this respect, this thesis outlines two centralized algorithms that are based on cross-entropy optimization and Gibbs sampling.

Next, this thesis considers information freshness when selecting devices, where the HAP aims to minimize the average Age of Information (AoI) of samples from devices. Specifically, the HAP must select devices to sample and transmit frequently. Further, it must select devices without channel state information. To this end, this thesis outlines a decentralized Q-learning algorithm that allows the HAP to select devices according to their AoI.

Lastly, this thesis considers targets with time-varying states. As before, the aim is to determine the best set of devices to be active in each frame in order to monitor targets. However, the aim is to optimize a novel metric called the age of *incorrect* information. Further, devices cooperate with one another to monitor target(s). To choose the best set of devices and minimize the said metric, this thesis proposes two decentralized algorithms, i.e., a decentralized Q-learning algorithm and a novel state space free learning algorithm. Different from the decentralized Q-learning algorithm, the state space free learning algorithm does not require devices to store Q-tables, which record the expected reward of actions taken by devices.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Associate Professor Kwan-Wu Chin, for his patient guidance, encouragement, brilliant ideas, dedication, and hard work throughout my PhD studies. In the past years, Professor Kwan-Wu Chin has always lit the way for me. I sincerely appreciate his help in improving my research ability, critical thinking, and writing skills. He has truly set a great example for me in both study and life.

Special thanks to my co-supervisor Associate Professor Raad Raad for his help in refining my research work.

Sincere thanks to all of my friends and lab mates in building B39-201 for all the encouragement, advice, constructive criticisms, joyful conversations, and awesome Squad sessions.

Next, I would like to thank my family. Without their love, hard work, support, guidance, and encouragement, it would be impossible for me to even begin my journey in Australia. Particularly, I want to express my gratitude to my parents, whose hard work supported me financially throughout my study abroad.

Lastly, I would like to appreciate Miss Siyu Zhang, who is the most important girl in my life and the brightest star in my universe. I am grateful that Siyu appeared in my life and has helped me navigate through life and motivated me to the terminus of my PhD journey.

Contents

List of Figures

List of Tables

Abbreviations

AP	Access Point
AoI	Age of Information
AoII	Age of Incorrect Information
BS	Base Station
CSI	Channel State Information
CE	Cross-Entropy
EH	Energy Harvesting
HAP	Hybrid Access Point
IoT	Internet of Things
MAC	Medium Access Control
MDP	Markov Decision Process
OFDM	Orthogonal Frequency Division Multiplexing
SNR	Signal-to-Noise Ratio
QoS	Quality of Service
QL	Q-Learning
RL	Reinforcement Learning
RR	Round Robin
RMAB	Restless Multi-Armed Bandit
SWIPT	Simultaneous Wireless Information and Power Transfer

TDMA	Time Division Multiple Access
WPCN	Wireless Power Communication Network
WPT	Wireless Power Transfer
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Background

Internet of Things (IoT) networks interconnect humans with objects instrumented with transceivers, processors, data storage, power source(s), sensors or/and actuators [1]. These smart objects or 'things' are thus capable of sensing their surroundings and communicate with other smart devices/objects to help each other accomplish tasks [2]. As a result, these 'things' enable a broad range of applications. Example application domains include (i) transportation and logistics, (ii) healthcare, (iii) smart cities/homes, (vi) agriculture, and (v) social, see details in Table 1.1. To elaborate, applications in transportation include next generation vehicles with driving assistance [1]. In healthcare, wearable devices are used to monitor the health of patients and provide automatic health records [3]. Further, devices are capable of automatically transmitting collected data to hospitals during emergencies [4]. In smart environments such as homes, devices are used to monitor and control appliances to improve energy usage [5]. On the other hand, IoT facilitates precision agriculture, whereby farmers use an IoT network to monitor their crops or/and protect their farms against pests efficiently [5]. Lastly, smartphones and wearable devices are now capable of gathering information relating to user activities [6], which in turn helps users better manage their daily life.

Domain	Application
Transportation and logistics	Vehicles exchange information to provide assisted driving and safe driving conditions [1], [7].
	Systems that consist of smarts devices automatically monitor and optimize every link of supply chain to provide better customer services [8].
Healthcare	Multiple wireless devices collaborate to provide continuous bio-signal and health monitoring of patients [3], [9].
	Telemedicine systems monitor and transmit patient vital signs to hospitals [4].
Smart environment	Sensors and actuators distributed in houses to control smart household applications and provide comfortable living environment [5].
	Monitor and analyze users' behaviors to schedule the operation time of smart household applications [10].
Agriculture	Smart systems that automatically monitor soil moisture and control watering time [5].
	Monitoring pollination process to ensure the origin of agricultural products [11], [12].
Social	Smart wearable devices automatically update users information such as locations and activities to Facebook or Twitter [6].
	IoT-enabled smartphones check predefined dating and friendship information to automaticallytransfer contact information to other smartphones [13].

Table 1.1: IoT applications.

A majority of the said applications involve targets monitoring, which is a key step to achieving automated services and improved user experience. For example, smart transportation, logistics, and storage systems require the capture of real-time state of stocks such as fruits, vegetables, and meat in order to provide the freshest products to customers [14], [15]. For smart healthcare applications, it is critical to continuously monitor the state of patients such as their blood pressure, heart rate, and oxyhemoglobin saturation [1], [16]. Another example is to determine the state, e.g., temperature or occupancy, of smart homes in order to reduce their energy consumption [17]. In terms of precision agriculture, farmers may track soil moisture level [5]; this information can then be used to regulate water usage.

A key issue in sensor networks is energy management and consumption. The amount of energy at sensors/devices and their energy consumption jointly determine their lifetime. In particular, various sensors have different energy consumption when they are working in different modes, see Table 1.2 for details. In this respect, a dynamic power management strategy [18] that switches sensors/devices between sleep and active mode according to their surroundings is a viable approach to extend their lifetime [19]. An alternative approach is energy harvesting, which will be discussed in the next section.

Sensor	**Nominal voltage**	**Power consumption**		**Response time**
		Sleep mode	Measuring	
Prism photo sensor (KP1430) [20]	5 V	N/A	75 mW	24 μs
Humidity & Temperature (HTU31 RH/T) [21]	5 V	1 μW	2.25 mW	5-10 s
Flow meter (SFM3003) [22]	3.3 V	3.6 μW	18.15 mW	3 ms
Prism photo sensor (KP1650) [23]	5 V	N/A	75 mW	22 μs
Magnetic sensor (TMAG5328) [24]	1.65-5.5 V	1.65 μW	10 mW	N/A
Digital image sensor (AR0130CS) [25]	12-20 V	N/A	270 mW	N/A
Pressure sensor (P1A) [26]	5 V	N/A	25 mW	5 ms

Table 1.2: Example of different types of sensors and their energy consumption for different operational modes.

1.2 Energy Harvesting

To date, as shown in Table 1.3, devices can harvest energy from various sources, namely solar [27], wind [28], thermal [29], mechanical motion [30], and kinetic [31]. Harvesting solar energy has been considered in many works; see [32] for details. During daytime, the power density of solar energy is capable of achieving 100 mW/cm^2 [32]. For indoor environments, the power density of solar energy significantly decreases to 3.2 μW/cm^2 [33]. Wind energy is also popular. When the wind speed is between 2 m/s and 9 m/s, a wind turbine is capable of generating 100 mW of power. Another source of energy includes thermoelectric, where it has a power density of around 20-60 mW/cm^2.

The aforementioned energy sources have the following disadvantages: (i) they are unpredictable or only partially predictable, (ii) they are uncontrollable, and (iii) the amount of available energy is a function of device location and time. In contrast, kinetic and mechanical motion are controllable and predictable. However, they require machinery, e.g., car engines, or continuous movements, to sustain energy generation. In this respect, Radio Frequency (RF) is advantageous as it is a controllable energy source that can be used to power devices/sensors at any time and place [34].

Energy source	**Work**	**Characteristics**	**Energy available**	**Applications**
Solar - outdoor	[27]	Uncontrollable, partially predictable	100 mW/cm^2	Wireless sensors, cellular base station
Solar - indoor	[33]	Uncontrollable, partially predictable	3.2 μW/cm^2	Wireless sensors
Wind	[28]	Uncontrollable, unpredictable	100 mW at wind speeds 2 m/s - 9 m/s	Wireless sensors, cellular base station
Thermal	[29], [35]	Uncontrollable, unpredictable	10 μW/cm^2 - 1 mW/cm^2	Human body, wearable, consumer devices
Mechanical motion	[30]	Controllable, predictable	30 mW	Vehicle, wireless sensors
Kinetic	[31], [36]	Controllable, predictable	1 μW - 7 W	Wearable devices

Table 1.3: Examples of energy sources.

1.2.1 RF Charging

RF charging uses far-field wireless power transfer technology. In general, the system consists of one or more energy transmitters, receivers/devices equipped with an RF-energy harvester. The first step for a device is to collect RF energy. It then converts RF power into Direct Current (DC) by using an impedance matching circuit with a rectifier. Finally, the direct current is used to charge a battery/capacitor or/and drive a load [37] [38].

There are several factors that dictate the amount of harvested RF energy. They include (i) a transmitter's transmit power, (ii) channel gains, and (iii) energy harvester efficiency. In particular, an energy harvester's efficiency is affected by the following factors: (a) whether a harvester is constructed using Complementary Metal Oxide Semiconductor (CMOS) transistor or Metal Oxide Semiconductor Field Effect Transistor (MOSFET), (b) carrier frequency, and (c) input power, see examples in Table 1.4. As shown in Table 1.4, the energy conversion efficiency of RF energy harvesters that are based on MOSFET is at most 83.7% when the input power is

15 dBm [39]. On the other hand, CMOS based harvesters have an efficiency of at most 69.5% when the input power is 5.2 dBm [40].

Literature	Minimum input power	Output voltage of minimum input power	Peak conversion efficiency @ RF input power	Frequency	Technology
M. Stoopman et al. [41]	-27 dBm	1 V	40% @ -17 dBm	868 MHz	90 nm CMOS
M.A. Abouzied et al. [42]	-21.7 dBm	1 V	N/A	850 MHz	180 nm CMOS
P.H. Hsieh et al. [43]	-17 dBm	2 V	44.1% @ -12 dBm	900 MHz	180 nm CMOS
Z. Hameed et al. [44]	-20.5 dBm	1 V	32% @ -12 dBm	902-928 MHz	130 nm CMOS
L. Fadel et al. [45]	-19.5/-25 dBm	1 V	27% @ -16 dBm	915 /2440 MHz	N/A
V. Kuhn et al. [46]	-20 dBm	N/A	84% @ 5.8 dBm	Multi-Band	N/A
Z. Wang et al. [39]	-15 dBm	N/A	83.7% @ 15 dBm	2450 MHz	MOSFET
D. Michelon et al. [47]	-24 dBm	1.2 V	68% @ N/A	900 MHz	CMOS
A.K. Moghaddam et al. [40]	-35 dBm	N/A	69.5% @ 5.2 dBm	953 MHz	130 nm CMOS
Y. Yu et al. [48]	-17.7 dBm	1 V	36.5% @ -10 dBm	900 MHz	65 nm CMOS

Table 1.4: Examples of RF energy harvesters and their energy conversion efficiency.

To date, prior works have considered dedicated or ambient RF charging. A dedicated RF source is controllable, where its transmit power, transmission time and duration can be optimized to supply energy to devices. For example, the dedicated RF source in [49] is capable of adjusting its energy transmission duration. In [50], a dedicated RF source can control both its transmit power and charging duration. Ambient RF energy sources include analog/digital TV signals, AM/FM radios, and Wi-Fi signals [51]. Unlike dedicated energy sources, these ambient energy sources are not aware of RF-energy harvesting devices operating in their vicinity. Further, it is not possible for devices to request energy from these energy sources. To date, past works have considered ambient RF sources that operate between 0.2 and 2.4 GHz. Example works include [52], where devices rely on RF signals broadcasted by a television tower located 6.5 km from them. In addition, in [53], the authors present a prototype that harvests energy from a WiFi signal. The aforementioned prototype is capable of harvesting energy at a rate of 2.77 μJ/s when it is located around 8.5 meters from a source.

Given the advances in RF charging, Wireless Power Communication Networks (WPCNs) are now a reality, where devices are charged by a Hybrid Access Point (HAP) or Power Beacon (PB). Advantageously, WPCNs help realize the delivery of both energy and data over the same wireless medium [54].

Figure 1.1 shows two different WPCN architectures. In Figure 1.1(a), there are RF energy harvesting devices and an access point (AP). There is also a PB, which is used to charge wireless devices. In the downlink, wireless devices collect RF energy from a power beacon [55]. The harvested RF energy is then used by devices to transmit their data to the AP. Another architecture is shown in Figure 1.1(b), where there is a HAP, which serves as a PB and an AP. In other words, the HAP is responsible for delivering energy to wireless devices in the downlink and receiving data from these devices in the uplink.

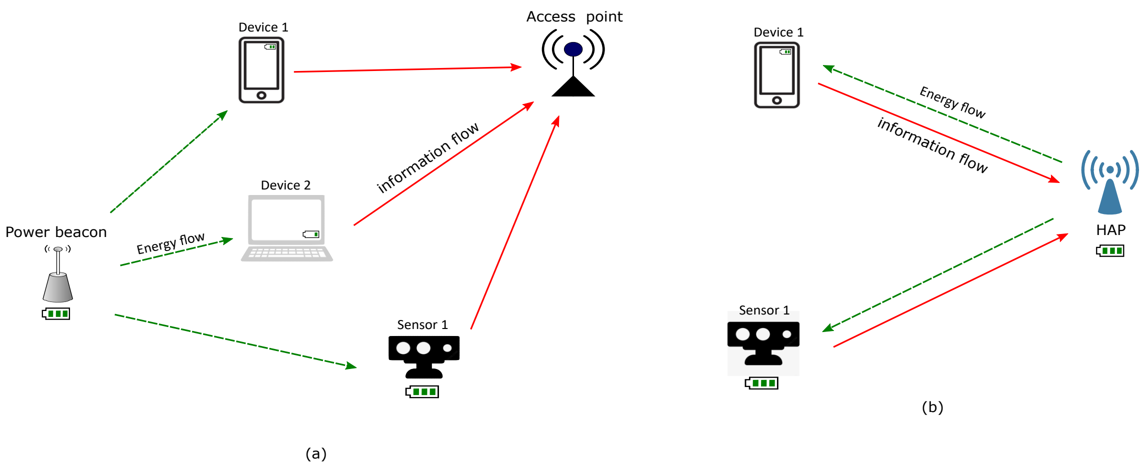

Figure 1.1: WPCNs architectures: (a) separated energy transmitter and information receiver, and (b) an HAP-based WPCN.

1.3 Research Statement

This thesis studies devices/sensors selection problems in single-hop WPCNs that use the harvest-then-transmit protocol [56], where in each frame, the HAP first broadcasts energy to devices/sensors via downlink RF energy transfer, see Figure 1.2 (a). After that, it selects a set of devices to transmit data or request them to collect samples, see Figure 1.2 (b). Here, a key research problem is to select the best set of devices to transmit or sample in each frame in order to optimize a performance metric. The performance metrics of interest include (i) sum rate, (ii)

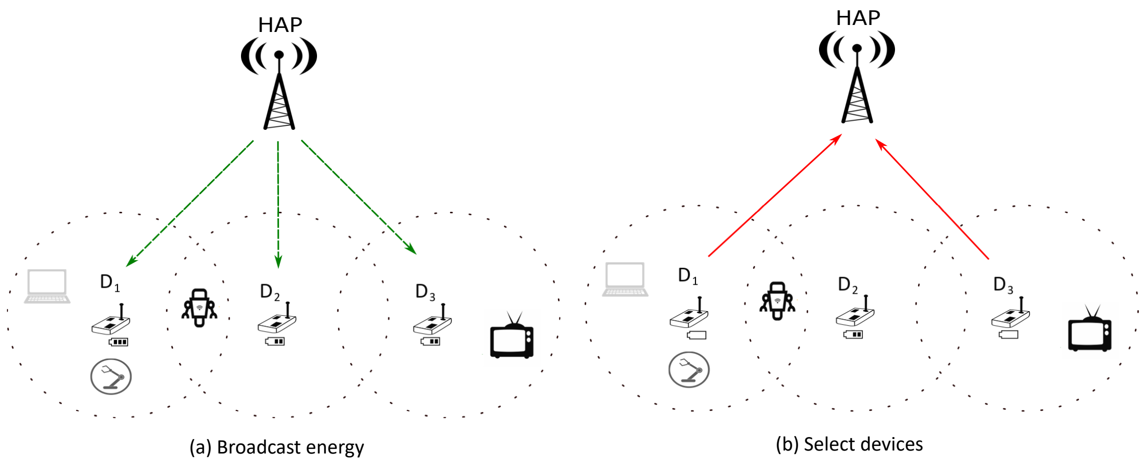

Figure 1.2: An example WPCN. The red and green arrows represent energy flow and data flow respectively.

Age of Information (AoI), and (iii) Age of Incorrect Information (AoII) of multiple targets. These aims are elaborated in the following sections.

1.3.1 Sum rate

This thesis first considers devices that transmit their samples via Time Division Multiple Access (TDMA) to a HAP. The main problem is to determine the set of devices that have the highest data rate, i.e., devices that have the highest energy level and the best channel state, in each frame.

There are several key challenges: (i) each device has an imperfect battery, meaning in each frame, devices lose some of their energy over time, (ii) random channel gains, which cause different amounts of harvested energy at devices, (iii) a HAP/scheduler has imperfect channel state information and energy level information of devices, and (iv) the double near-far problem [56], where devices located far from a HAP require more frames to harvest energy to run their operations.

1.3.2 AoI

The second research aim of this thesis considers selecting devices in order to minimize AoI which is defined as the number of frames that have elapsed since the sample stored at the HAP was generated at the device. The key problem is to determine the set of devices that have sufficient energy to generate and send a sample to a HAP and the set of devices that should save energy for future use.

There are several key challenges: (i) a HAP has neither uplink CSI nor energy level of devices, (ii) the number of schedules increases exponentially with increasing number of devices and time frames, and (iii) the double near-far problem, meaning a device may have a large AoI because it requires a longer period to harvest sufficient energy to transmit a sample successfully.

1.3.3 AoII

Lastly, this thesis considers devices/sensors that monitor one or more targets, see Figure 1.2. The state of targets, e.g., machines or the ingresses of a building, is driven by a stochastic process. In each frame, selected devices/sensors are responsible for monitoring targets and transmitting the state of monitored targets to a HAP over an orthogonal channel.

Given the said assumptions, the main aim is to select the best set of devices to be active, i.e., sample and transmit, and placing other devices to sleep mode. The aim is to minimize the AoII of targets. Specifically, the AoII of a target is defined as the number of frames that have elapsed since a state mismatch exists between the HAP and a target. In this respect, a key problem is to determine (i) whether devices have sufficient energy to sense targets and then transmit successfully to the HAP, and (ii) whether a device that has sufficient energy monitors targets with a high AoII.

There are several challenges. First, the HAP has no channel state information and energy level of devices. Second, the HAP/scheduler has to select devices/sensors without knowing which set of targets is under the monitor of a device. Moreover, the HAP does not know whether other devices/sensors are monitoring the same target at the same time. Lastly, the HAP is not aware of the stochastic process governing the state of each target.

1.4 Contributions

This thesis contains three major contributions. They include (i) two centralized algorithms for device selection to maximize the sum rate at a HAP, (ii) applying reinforcement learning to select devices to sample and transmit in order to minimize AoI, and (iii) two decentralized reinforcement learning algorithms that can be used to determine the best devices without knowing the channel state information and energy level of devices. The following sections further detail the aforementioned contributions.

1.4.1 Sum Rate Maximization

First, Chapter 3 outlines the problem of selecting devices in a WPCN in order to maximize sum-rate. Specifically, when selecting devices, a HAP/scheduler does not have perfect channel state information nor the energy state of devices, and each device is equipped with an imperfect battery. In this respect, Chapter 3 outlines a cross-entropy approach to identify the best set of devices to transmit data in each time frame over random channel gains to maximize sum rate. In addition, Chapter 3 contains a fast Gibbs sampling [57] approach, called Gibbs$^+$. It iteratively selects devices by evicting non-competitive devices. The proposed approaches are compared against random pick, round robin, original Gibbs sampling, and perfect information selection, which select devices according to known channel state information and energy level of devices. The results show that cross-entropy and Gibbs$^+$ produce a higher sum rate than the said benchmark algorithms. In addition, Gibbs$^+$ is faster than cross-entropy but capable of achieving approximately the same sum rate as cross-entropy.

1.4.2 AoI Minimization

Next, Chapter 4 considers minimizing AoI. A HAP is responsible for charging devices and instructing some devices to carry out sampling. After that, devices transmit their samples to the HAP. It considers a challenging and practical question: how to select devices without uplink channel state information and energy level of devices in order to minimize AoI?

To this end, Chapter 4 contains a novel decentralized Q-learning algorithm that can be used by a HAP to determine the best set of devices to sample and transmit their data. Advantageously, when using a decentralized Q-learning algorithm, each device will determine whether to send a request to the HAP according to its energy level, channel state, buffer state, and Q-table. Instead of selecting devices according to their channel and energy state, the proposed decentralized Q-learning algorithm allows the HAP to select devices according to their request and AoI. The results show that the decentralized Q-learning algorithm consistently achieves a much lower average AoI than round-robin, random pick, and AoI-greedy strategies. In addition, the average AoI of the outlined decentralized Q-learning algorithm is only a little bit higher than the optimal selection strategy which requires perfect channel state information and energy level of devices.

1.4.3 AoII Minimization

Lastly, Chapter 5 studies device selection for the purpose of minimizing AoII. Each device is responsible for monitoring one or more targets with time-varying states. A HAP is responsible for charging devices and instructing some devices to sample targets that are within their sensing range to transmit samples. During this process, there are the following challenges: (i) the HAP does not have uplink channel state information and energy level of devices, and (ii) devices do not know if there any other devices that monitor the same targets.

Henceforth, Chapter 5 outlines two decentralized reinforcement learning algorithms, namely (i) decentralized Q-learning, and (ii) a novel state space free learning algorithm to determine the best devices to monitor targets. Briefly, for both algorithms (i) and (ii), each device independently decides its probability to sample and then transmit, i.e., probability to be active, in each frame. However, for the decentralized Q-learning algorithm, devices determine the aforementioned probability according to their historical uplink channel state information, energy level, and buffer state, while the state space free learning algorithm does not require the devices to know the said information. In simulation studies that compare the proposed two algorithms against random pick, round robin, ϵ-greedy strategy, the results show that decentralized Q-learning and state space free learning algorithm achieve much lower AoI than the aforementioned benchmark algorithms. In addition, the average AoII of the proposed two algorithms is only slightly higher than the optimal selection strategy which requires perfect channel state information and energy level of devices.

To summarize, this thesis includes three contributions that study different devices/nodes selection problems in WPCNs. Figure 1.3 shows the relationship between the aforementioned three contributions.

1.5 Publications

The aforementioned contributions have resulted in the following publications:

1. **L. Zhang** and K-W Chin, *On Devices Selection in RF-Energy Energy Harvesting Wireless Networks*, in IEEE Systems Journal, 15(4), pp 4816-4826, December, 2021.

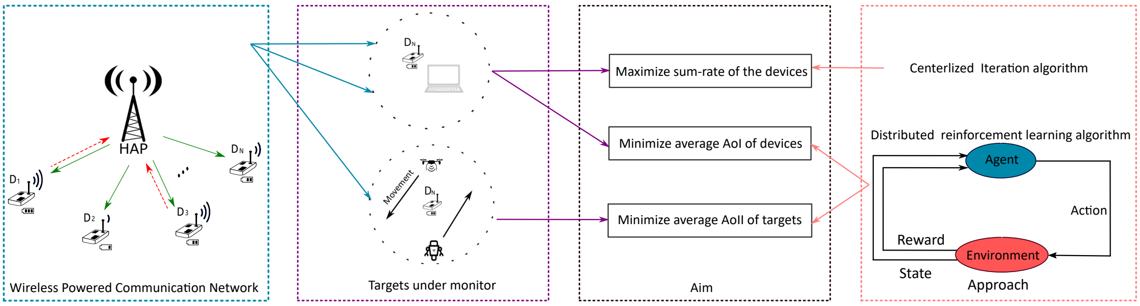

Figure 1.3: The three contributions in this thesis. All of them consider devices selection problems in WPCNs. The first contribution aims to maximize sum-rate of devices and apply centralized algorithm as its solution. The second and third contributions aim to minimized the average AoI of devices and average AoII of targets. They apply distributed reinforcement learning algorithms as their solutions.

2. **L. Zhang** and K-W Chin, *A Distributed Learning Device Selection Method for Minimizing AoI in RF-Charging Networks*, in IEEE Communications Letter, 25(11), pp 3733-3737, November, 2021.

3. **L. Zhang** and K-W Chin, *On Device Selection for Optimizing AoII in Wireless Powered IoT Networks, IEEE Internet of Things, 2023.* Under review.

1.6 Thesis Structure

1. *Chapter 2.* This chapter provides a comprehensive survey of past works that consider device selection in energy harvesting sensor networks. Moreover, it focuses on works aim to optimize sum-rate, AoI and AoII.

2. *Chapter 3.* This chapter outlines a cross-entropy based algorithm and a fast Gibbs sampling approach that aim to maximize the sum-rate of multiple devices in an energy harvesting WSN.

3. *Chapter 4.* This chapter proposes a reinforcement learning-based method to determine the set of transmit devices so as to minimize the average age of information in an RF-charging WSN.

4. *Chapter 5.* This chapter outlines a state-space free reinforcement learning

method and a Q-learning method to determine the active device set to minimize the average age of incorrect information in an energy harvesting WSN.

5. *Chapter 6.* This chapter concludes the thesis, provides a summary of key contributions and outlines possible future research direction.

Chapter 2

Literature Review

This chapter reviews past works that consider device selection in energy harvesting WSNs or WPCNs.

2.1 Device selection in EH WSN

Many works have considered device selection in an energy harvesting wireless sensor network. These works aim to select the best set of devices/sensors to transmit/receive data under limited channel resources in order to achieve a specific goal, e.g., maximize throughput. The following subsections classify past works according to their aim(s); each subsection then further classifies past works according to their energy source, see Figure 2.1 .

2.1.1 Throughput Maximization

A number of works have considered devices powered by an ambient energy source, e.g., [58–68]. In these works, the fusion center or scheduler is responsible for selecting the best K sensors out of M sensors to transmit over orthogonal channels. For example, in [58], each sensor is equipped with a unit capacity battery and they used a two-state Markov model to formulate the energy evolution at sensors. The fusion

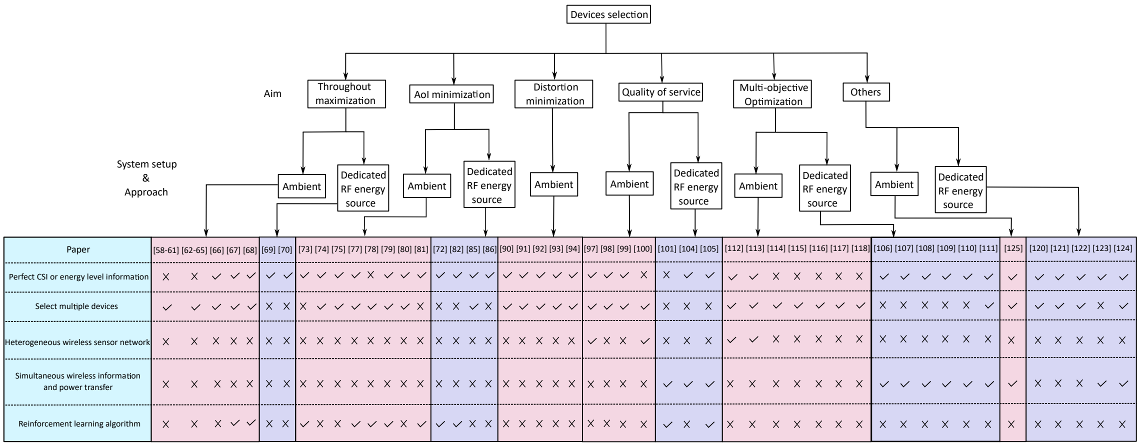

Figure 2.1: Taxonomy of prior works that study devices selection problem in EH WSN.

center has no knowledge of the battery state of sensor devices. Nodes always have data to transmit. A node that has sufficient energy can transmit in a time slot when it is selected. Battery leakage is considered in [58], where the device selection problem is modeled as a partially observable Markov decision process (POMDP). It is then solved using the restless multi-armed bandit (RMAB) framework [58], where the goal is to maximize the number of packets received by the central node.

Different from [58], in [59] and [60], each sensor node is equipped with a battery with an arbitrary capacity. A similar myopic policy to the work in [58] is studied by Pol et al. in [59] and [60]. The aforementioned two works, i.e., [59] and [60], have the following differences: (i) reference [59] considered backlogged sensor devices and static channel while the work in [60] assumed un-saturated sensors and considered the influence of uplink channel state, and (ii) references [59] and [60] proved the optimality of myopic policy in different cases, respectively. Specifically, in [59], a myopic policy is proved to be optimal when sensors cannot harvest energy and transmit simultaneously. The work also showed that the transition probability of energy harvesting processes is influenced by the scheduling policy. In [60] the authors proved the optimality of their myopic policy when the energy harvesting process of a sensor device is independent in each slot, and when sensor devices have no battery. The aim of [61] is similar to [58], where the latter work considered devices with a finite and infinite battery. Battery leakage is ignored in [61], and the authors used the same myopic policy as the work in [61]. The authors of [61] further investigated the throughput performance of the said myopic policy under general energy harvesting processes, i.e., Markovian, i.i.d., non-uniform and uniform.

In [62], [63], [64], and [65], Pol et al. studied a scheduling algorithm for energy harvesting WSNs. Specifically, in [62], the fusion center has no information about the charging process and the battery state of sensors. Instead, it knows the previous transmission attempts of sensor nodes. The fusion center uses a so-called uniformizing random ordered policy (UROP) algorithm. Its basic idea is to select sensor devices based on a predefined random priority list and the outcomes of previous transmission attempts. Specifically the fusion center initially orders sensors randomly and generates a priority list. In the first time slot, it schedules the first K sensors on the list. If a scheduled sensor is able to transmit in a time slot, it will be selected in the next time slot again. Otherwise, the fusion center will replace the sensor with the next sensor in a priority list. The authors of [62] then showed that UROP is a near optimal policy assuming infinite battery. Reference [63] extended the work in [62] to the un-saturated case. Different from [62], in [63], a selected node can only transmit when it has sufficient energy and it has one packet in its buffer. Otherwise, the channel allocated to the node will be idle. The fusion center in [63] does not know the battery and buffer state of nodes. Further, it has no knowledge of the charging and data arrival process at nodes. In [64] and [65], based on the work of [62] and [63], Pol et al. further investigated the throughput performance of UROP. Specifically, instead of considering a certain energy harvesting process, i.e., Poisson arrival and Markovian process, UROP is proved to be asymptotically optimal for general energy harvesting processes in [65].

The work in [66] considered a device selection problem with perfect information. Specifically, the following information is known: (i) energy arrival rate at devices, (ii) battery state, and (iii) uplink channel state of devices. The problem at hand is to select the best set of devices to transmit in each time slot so as to maximize sum rate. This problem is solved using an online policy that selects transmitting devices according to (i), (ii), and (iii).

In [67], and [68], a reinforcement learning (RL) method is used to select devices. Specifically, in [67], the authors studied multi-access control and battery prediction in an energy harvesting in IoT system. The base station or fusion center has channel state information. The work in [67] addressed three aims: (i) determine the access control policy that maximizes sum rate, (ii) obtain the prediction policy such that the prediction loss is minimum, and (iii) consider access control and battery prediction simultaneously in order to maximize long-term discounted sum rate and minimize cumulative battery prediction loss simultaneously. Specifically, reference [67] proposed a long short-term memory deep Q-network based approach to achieve aim (i). The state space of the aforementioned approach consists of channel and battery states. The action is to select a set of users to transmit. The reward is the sum rate. The authors propose a deep long short-term memory neural network-based battery prediction algorithm to minimize the prediction loss so as to achieve aim (ii). Specifically, the input of the neural network is a three tuple which consists of scheduling history, predicted battery state of users, and selected users' true battery state. The output of the neural network is a prediction of battery state. The authors then proposed a two-layer deep Q-network to achieve aim (iii).

In contrast to [58–67], the authors of [68] considered an energy harvesting communication network that consists of an energy harvesting access point (AP) and multiple devices. Specifically, in each time slot, the energy harvesting access point is responsible for harvesting energy from the environment and selecting a set of devices to deliver information. The AP selects the best set of devices to receive data in order to maximize downlink sum rate. The AP runs a deep reinforcement learning approach that considers channel state information, its battery state information, and received energy. The AP learns a policy to select a subset of devices and allocate channels to these selected devices. The reward is its sum rate.

A dedicated energy source or HAP can be used to power devices, e.g., [69], [70].

In this context, both [69] and [70] have studied a device selection problem in a full-duplex WPCN that consists of a HAP and multiple energy harvesting devices. In each time slot, one device will be selected to transmit data to the HAP. Other devices will harvest energy whenever the HAP transmits an RF signal. In [69], the HAP has two antennas. It is capable of broadcasting energy and receiving data from a single device simultaneously. Further, it runs a scheduler that has the energy state of devices. The scheduler (i) selects the best transmit device in each time slot so as to maximize the average throughput, and (ii) selects the best transmit device so as to trade off system throughout and device fairness. To achieve (i), the scheduler use a throughput-oriented scheduling scheme. Its basic idea is to always select the device with the maximum weighted residual energy to transmit in each time slot. To achieve (ii), the scheduler uses a fairness-oriented scheduling scheme. For each device, the scheduler will calculate the ratio between its current energy level and its average energy level within the period from its last transmission to the current time slot. In each slot, the scheduler selects the device with the highest ratio to transmit. Similarly, in [70], the HAP has multiple antennas. Each device has one antenna. In each time slot, the HAP is responsible for transmitting energy. It allocates some antennas for energy delivery and data reception. Note, the scheduler of [70] has perfect channel state information. Its aim is to select the best transmit device, decide the set of antennas to transmit energy, and optimize the beamforming weight of the hybrid AP so as to maximize the average sum rate of energy harvesting devices while satisfying their minimum average data rate requirement. The HAP decides the best set of antennas to transmit energy and optimize beamforming in each time slot. It then selects the transmitting device according to the decided beamforming vector and an antenna selection vector.

Table 2.1 summarizes the aforementioned works. All works consider device selection in an energy harvesting communication network. Their aim is to maximize throughput or sum rate. We see that only reference [69] and [70] have considered a dedicated charger, i.e., a HAP that broadcasts a radio frequency signal. These works studied the influence of downlink channel state on received energy at devices. Their solution, however, only selects a single device, which means the complexity of their problem does not grow exponentially with increasing number of time slots and the number of selected devices. The scheduler in [66–70] has channel state, battery state of device or/and energy arrival rate of devices information.

In this respect, a key issue is that the scheduler is required to poll devices to collect the aforementioned information, which is impractical in an energy harvesting wireless sensor network that contains a large number of devices.

Paper	**Operation Mode**	**Number of selected devices**	**Energy source type**	**Information**	**Solution**
Iannello et al. [58] Blasco et al. [59], [60] Gul et al. [61]	Half-duplex	Multiple	Ambient	None	Myopic policy
Gul et al. [62], [63], [64] [65]	Half-duplex	Multiple	Ambient	Transmission attempts of sensors	Uniformizing random ordered policy algorithm
Yang et al. [66]	Half-duplex	Multiple	Ambient	Channel state, battery state, and energy arrival rate	Select devices according to known information
Chu et al. [67]	Half-duplex	Multiple	Ambient	Channel state	Reinforcement learning algorithm
Luo et al. [68]	Half-duplex	Multiple	Ambient	Channel state, battery state, and energy arrival rate	Deep reinforcement learning algorithm
Zhai et al. [69]	Full-duplex	Single	Dedicated RF source	Battery state	Select device according to battery state information
Park et al. [70]	Full-duplex	Single	Dedicated RF source	Channel state	Joint ID, Antenna, and Beamforming (IAB) algorithm

Table 2.1: A comparison of works that study device selection and aim to maximize throughput or sum rate.

2.1.2 Age of Information

The novel metric Age of Information (AoI) is now popular. Specifically, the AoI of a sensor or device is defined as the number of frames that have elapsed since the sample stored at the data receiver was generated at the device [71]. To minimize AoI, there are a few key challenges. First, the complexity of device selection increases exponentially with more devices and the number of time slots. Secondly, many works have employed a Markov decision process, e.g., [72]. In this respect, the key challenge is the curse of dimensionality, where the state space of agent, which usually models the channel and battery state of devices, grows exponentially with increasing number of devices.

Many AoI works have considered an ambient energy source [73], [74], [75], [76], [77], [78],[79],[80],[81]. For example, in [74], [75], [76], [78], Hatami et al. considered an energy harvesting sensor network that consists of multiple sensors, an edge node, and multiple users, i.e., data receivers. Data receivers will send their request to the edge node to ask for a sample. Next, the edge node selects sensors to transmit samples in order to update its cache. The aforementioned four works use a so-called on-demand AoI. In [78], a Q-learning algorithm is used select the best set of devices to transmit in each slot so as to trade off the energy consumption and on-demand AoI. Their scheduler only knows the AoI and the partial battery state of devices. Different from [78], in [76], the proposed solution selects the best set of devices to transmit in each time slot to minimize the weighted on-demand AoI of each device. The scheduler in [76] has perfect battery state information of each device, which is different from [78]. The work in [82] formulated the device selection problem as a Markov decision process and propose two solutions: (i) a model-based method, i.e., value iteration algorithm (VIA), and (ii) a model-free reinforcement learning (RL) method, i.e., Q-learning algorithm. The setting in [75] is similar to [76]. There are only two differences between the aforementioned two works. Firstly, the formula to calculate the on-demand AoI in [75] is different from the work in [76]. Secondly, in [75], the scheduler knows the request information of data receivers, while this information is unknown in [76]. In [74], there is a transmission constraint, which means in each slot, a scheduler selects at most M sensors to transmit due to limited channel resources. There is no such constraint in [75, 76] and [78].

In [73] and [81], an energy harvesting network consists of multiple energy harvesting users or sensors and multiple receivers. At most one sensor will be selected each time by a scheduler to transmit to a receiver. The remaining sensors harvest energy from their environment. In [73] and [81], the scheduler knows the AoI, the uplink channel state information (CSI), and the energy level of devices. Specifically, in [81], the proposed solution selects the best device to transmit in each slot so as to minimize long-term AoI. To achieve the aforementioned aim, the said solution uses a deep reinforcement learning algorithm to select a device according to the aforementioned information. Different from [81], in [73], the proposed solution (i) selects a user to transmit, (ii) determines the action of the selected user, i.e., whether to transmit a packet to its intended receiver, and (iii) determines the transmit power of the selected device so as to minimize the long-term age of information. The solution employed a neural network to make decision (i), while decision (ii) and (iii) are made based on the energy level, channel state, and Signal-to-Noise Ratio (SNR) threshold of a selected device.

The work in [79] and [77] considered a wireless sensor network with a fusion center, i.e., scheduler, and multiple energy harvesting sensors. The fusion center is responsible for selecting at most M devices from K devices to sample and transmit. Specifically, in [77], each sensor monitors a specific target and has its own AoI threshold. The fusion center knows the following information: (i) the battery state of each device, (ii) the transmission state in the last slot, i.e., when transmissions are successful, (iii) the age of information threshold of each device, and (iv) the selected devices in the last slot. The fusion center aims to select the best set of devices in each slot according to the aforementioned information in order to (i) minimize the total number of times that the AoI of devices exceeds a threshold, and (ii) minimize the sum AoI of devices. To achieve its aim, it runs a double deep Q-learning (DDQN) [83] based algorithm. Different from [79] and [77], multiple sensors may cooperatively monitor the same target in [84]. Both works used age of correlated information (AoCI), which is derived from the concept of AoI. Their aim is to (i) select a set of sensors to sample and transmit, and (ii) decide the sensed target for each selected sensor so as to minimize the average age of correlated information. Similar to [79], a deep reinforcement learning (DQL) based algorithm is used in [84] to solve (i) and (ii).

In [80], Liu et al. considered a communication network that consists of a base station that has limited computation resources and multiple heterogeneous energy harvesting devices. Specifically, there are the following two types of devices: (i) energy harvesting devices that have a battery, and (ii) energy harvesting devices without a battery. The aim of [80] is to jointly address the problem of selecting a set of devices to transmit, decide the transmit power of selected devices and allocate the computation resource of the base station to minimize the age of status updates, i.e., the average weighted age of information. This problem is solved using a stochastic gradient descent based online algorithm.

Many works have considered wireless powered communication networks (WPCNs). For example, in [85], [72], [86], the network has a HAP and multiple RF energy harvesting sensors or devices. A scheduler knows the energy state, uplink channel state, downlink channel state, and age of information of all devices. Moreover, the work in [85], [72], [86] aim to decide (i) whether each time slot is used for wireless energy transfer (WET) or wireless information transmission (WIT) and (ii) the best set of devices to transmit in the wireless information transmission phase so as to minimize the long-term average weighted age of information. There are several differences between [85], [72], [86]. Firstly, in [85], Jin et al. considered non-orthogonal multiple access (NOMA), which means multiple devices can transmit simultaneously. While in [72], [86], the scheduler only selects one sensor to transmit in the wireless information transmission phase. Secondly, the solution presented in [85], [72], [86] is different. Specifically, in [85], Jin et al. applied Lyapunov optimization [87] to dynamically decide (i) and (ii) according to the energy state, uplink channel state, downlink channel state, and age of information of all devices. In contrast, the authors of [86] and [72] respectively applied the policy iteration algorithm (PIA) and deep reinforcement learning (DRL) as a solution.

Recently, intelligent reflecting surface (IRS) [88] has been shown to significantly improve wireless communication network performance [89]. This is because it could enhance channel conditions. In this respect, in [82], Cui et al. studied an IRS-assisted wireless powered communication network, which consists of a hybrid access point (HAP), multiple energy harvesting sensors, and an IRS. The aim is to jointly decide (i) the time slot used for wireless energy transfer (WET) or wireless information transmission (WIT), and the transmitting sensor in the WIT phase, (ii) the HAP's beamforming vector, and (iii) the phase shifting matrices of the intelligent reflecting surface so as to minimize the average age of information of devices. To achieve the aforementioned aim, Cui et al. propose a hierarchical deep reinforcement learning algorithm as a solution.

Table 2.2 summarizes and compares all the aforementioned works. The aim of these works is to select one or multiple sensors to transmit so as to minimize the average AoI or its variation in an energy harvesting communication network. These works proposed centralized algorithms whereby their scheduler selects devices according to known information. However, collecting battery and channel state information from all sensors may be impractical, especially in a network that contains a large number of sensors, see section 2.1.1. Only references [85], [72], [86], and [82] consider a WPCN and downlink energy transfer. However, the work in [72], [86], and [82] does not consider the multiple device selection problem. The only work that considers multiple device selection in a WPCN is [85]. However, Jin et al. do not consider a learning algorithm.

2.1.3 Distortion

A number of works have aimed to minimize distortion [90–94], which is defined as the mean square error (MSE) between reconstructed samples at a fusion center and original samples. These works, i.e., [90–94], have considered an energy harvesting wireless sensor network that consists of multiple sensors and a fusion center. Sensor

Paper	Joint optimization	Number of selected devices	Energy source type	Information	Solution
Leng et al. [73]	Yes	Single	Ambient	AoI,channel state, and energy level of sensors	Reinforcement learning algorithm
Hatami et al. [74]	No	Multiple	Ambient	AoI of sensors, request of users, and battery state of sensors	Relative value iteration algorithm
Jin et al. [85]	No	Multiple	Dedicated RF source	Energy state, channel state, and AoI of all devices	Lyapunov optimization
Hatami et al. [75]	No	Multiple	Ambient	AoI of sensors, request of users, and battery state of sensors	Value iteration algorithm
Hatami et al. [76]	No	Multiple	Ambient	AoI and battery state of sensors	Value iteration algorithm and Q-learning algorithm
Feng et al. [77]	No	Multiple	Ambient	Battery state of sensors, transmission state in the last frame, selected sensors in the the last frame, and AoI threshold of each sensor	Double deep Q-learning algorithm
Mohamed et al. [72]	No	Single	Dedicated RF source	Channel state, battery state, and AoI of all senors	Deep reinforcement learning algorithm
Mohamed et al. [86]	No	Single	Dedicated RF source	Channel state, battery state, and AoI of all sensors	Value iteration algorithm
Hatami et al. [78]	No	Multiple	Ambient	Imperfect battery state of all sensors	Q-learning algorithm
Zhao et al. [79]	No	Multiple	Ambient	Battery state of devices	Deep reinforcement learning based algorithm
Liu et al. [80]	Yes	Multiple	Ambient	Battery state of devices	Online algorithm
Leng et al. [81]	No	Single	Ambient	AoI, channel state, and energy level of sensors	Actor-Critic deep reinforcement learning algorithm
Cui et al. [82]	Yes	Single	Dedicated RF source	Channel state and energy level of sensors	Hierarchical deep reinforcement learning algorithm

Table 2.2: A comparison of works that study device selection and aim to minimize AoI or its variation.

nodes monitor a stationary source and generate independent and identically distributed (i.i.d.) samples. A fusion center is responsible for reconstructing samples according to data from sensors.

To minimize reconstruction distortion, the key problems considered in prior works included (i) selecting the best subset of sensors to transmit in each time slot, and (ii) deciding the transmit power of sensors. Addressing problem (i) and (ii) is challenging because they are combinatoric in nature, NP-hard and results in a non-convex optimization problem.

A set of works, i.e., [91], [92], [94], have jointly considered problem (i) and (ii) and assumed there are finite channels. These works adopted a similar sensor network and same problem. However, they have a different solution. For example, in [94], a separate sensor selection and power allocation (SS-EH) algorithm is proposed to address problem (i) and (ii). The aforementioned algorithm has the following steps: (i) construct a vector where its elements are weights that determine the contribution of sensors to the reconstruction process, see details in [94], (ii) select the best K sensors according to the result of (i), i.e., select the largest K element in the aforementioned vector, and (iii) the transmit power of selected sensors by using the iterative algorithm in [95]. In [92], the authors extended their work in [94] and proposed a so-called joint sensor selection and power allocation (JSS-EH) algorithm that iteratively finds the optimal devices to transmit samples and their transmit power, see [92]. In [91], the authors formulated and transformed their joint device selection and power allocation problem into a convex problem. This problem is then solved using the Lagrangian duality approach.

Different from [91], [92], [94], the authors of [90] and [93] selected devices by controlling their transmit power, i.e., a device transmits when its allocated power is larger than zero, otherwise, a device does not transmit. In other words, the work in [90] and [93] has only considered problem (ii). To address problem (ii), the decentralized algorithm in [90] decides the transmit power of each device. Each device locally computes and reports its transmit power to a fusion center. Then the fusion center optimizes the transmit power of each device. In contrast, the centralized algorithm in [93] uses the Majorization-Minimization (MM) algorithm [96] to iteratively find the transmit power of each device.

Table 2.3, summarizes and compares all works in this section. All works consider ambient energy sources and none of them consider imperfect channel state and battery state information. This means no works have considered collecting channel and battery state information in a wireless sensor network that contains a large number of sensors, see discussion of the aforementioned challenge in Section 2.1.1. Moreover, only the work in [90] has applied a decentralized algorithm. No work has applied a reinforcement learning based solution.

Paper	**Energy storage loss**	**Joint Optimization**	**Energy source type**	**Information at fusion center**	**Solution**
Calov-Fullana et al. [90]	No	No	Ambient	Channel state, and amount of harvest energy of each sensor	Decentralized iteration algorithm
Du et al. [91]	Yes	Yes	Ambient	Channel power gain and battery state of sensors	Lagrangin duality approach
Calov-Fullana et al. [92]	No	Yes	Ambient	Channel state and amount of received energy of each sensor	Separate sensor and power allocation algorithm, joint sensor selection and power allocation algorithm
Calov-Fullana et al. [93]	No	No	Ambient	Channel state and amount of received energy of each sensor	Majorization -Minimization algorithm based centralized algorithm
Calov-Fullana et al. [94]	No	Yes	Ambient	Channel state and amount of received energy of each sensor	Separate sensor and power allocation algorithm

Table 2.3: A comparison of works that study device selection and aim to minimize reconstruction distortion.

2.1.4 Quality of Service

A number of works aim to achieve a specific objective, for example, maximize network lifetime, while meeting a quality of service (QoS) requirement [97–100]. In these works, there is a fusion center and multiple energy harvesting sensors. The problem at hand is device selection. There are two main challenges. First, the fusion center does not have perfect information of the energy level [97] or channel state information [101] of devices. Second, the considered device selection problem is NP-hard [97–99].

There is a set of past works, i.e., [97] and [100], that have considered a heterogeneous wireless sensor network. For example, there is a primary and secondary network in [97]. The sensors in the primary network harvest energy from solar and are responsible for monitoring the temperature and reporting to a fusion center. The sensors in the secondary network are responsible for monitoring solar irradiance and reporting their measurement to the fusion center. Based on the received data from the primary and the secondary network, the fusion center will forecast the energy level of sensors in the primary network. It aims to select the best set of sensors in the primary network to transmit according to predicted energy level so as to maximize network lifetime while meeting the QoS defined by users, see details in [97]. To address the sensor selection problem, Chen et al. proposed an algorithm based on the cross-entropy method [102]. Its basic idea is to maintain a probability distribution that is then adapted iteratively to determine the transmitting device.

Different from [97], in [100], two types of sensors, namely, (i) high-quality, and (ii) low-quality, collaborate to monitor the same physical phenomenon, see details in [100]. A fusion center aims to reconstruct a physical phenomenon according to the received data from sensors so as to accurately estimate and predict the monitored physical phenomenon. Further, it selects the best set of devices to transmit in order to minimize the cost of active sensors while meeting the QoS criterion required by users [100]. Similar to [97], the algorithm in [100] uses the cross-entropy method to address the sensor selection problem.

Different from [97] and [100], in [98] and [99] there is a single type of sensors. The fusion center knows the channel state and energy level of each sensor. Specifically, in [99], the fusion center selects the best set of sensors to transmit in each time slot in order to maximize the number of received samples while meeting the quality of service of users., i.e., ensure signal-to-noise ratio over a threshold. In [98], the authors extended their work in [99] to further consider transmission fairness. Similarly, the online algorithm in [98] and [99] selects devices according to their channel and battery state information.

Some works have also considered simultaneous wireless information and power transfer (SWIPT) [54, 103] and investigated a single device selection problem [101, 104, 105]. A key problem is whether devices should receive data or harvest energy. For example, in [104], the problem involved (i) selecting one device to receive data, and the other devices harvest energy from the RF-signal transmitted by a fusion center, and (ii) deciding the transmit power to each device so as to maximize the amount of harvest energy at devices and ensure the average data rate of devices over a threshold. The proposed joint device selection and power allocation algorithm selects a device and decides its transmit power according to known channel state information. In [105] and [101], there is a fusion center that has both grid power and renewable energy. It aims to (i) select one device to receive data. Other devices harvest energy from its RF-signal, and (ii) decide the amount of energy to draw from the power grid and battery to transmit a signal in order to maximize the average throughput, satisfy the energy requirement of a device , and ensure the energy consumption of the power grid lower than a threshold. Note that, in [105], the fusion center has perfect channel state information of devices, while in[101], it has imperfect channel state information. The work in [105] and [101] used a policy iteration algorithm and reinforcement learning algorithm to maximize average throughput while satisfying energy harvesting requirement of devices.

Table 2.4 summarizes and compares the works discussed in this section. Only the work in [97] and [100] has a fusion center that does not use the current battery state information of devices. They applied a cross-entropy method based algorithm to select the best devices to transmit. However, in [97], the fusion center has to forecast the energy level of sensors based on their historical energy level and observation of sensors in a secondary network. This means that the fusion center has to poll all sensors in the secondary network in each time slot for their observation, which is impractical in a large-scale wireless sensor network, see discussion in Section 2.1.1. Moreover, only reference [100] has studied mixed energy sources. However, the dedicated energy source in [100] is a power grid instead of an RF signal broadcast by a fusion center.

Paper	**SWIPT**	**Heterogeneous wireless sensor network**	**Energy source type**	**Infornation at fusion center**	**Solution**
Chen et al. [97]	No	Yes	Ambient	Historical battery state of sensors	Cross entropy method-based algorithm
Hentati et al. [98]	No	No	Ambient	Channel state and energy level of each sensor	Online algorithm
Hentati et al. [99]	No	No	Ambient	Channel state and energy level of each sensor	Online algorithm
Zhang et al. [100]	No	Yes	Mixed	None	Cross entropy method-based algorithm
Boshkovska et al. [104]	Yes	No	Dedicated RF source	Channel state information of devices	Online algorithm
Guo et al. [105]	Yes	No	Dedicated RF source	Channel state information of devices	Policy iteration and reinforcement learning algorithm
Guo et al. [101]	Yes	No	Dedicated RF source	Imperfect channel state information of devices	Policy iteration and reinforcement learning algorithm

Table 2.4: A comparison of works that study device selection and aim to meet QoS.

2.1.5 Multi-Objectives

This section presents two categories of works that have investigated device selection. The aim of these works is to optimize two objectives. The first category of works considered energy and data transfer in a WPCN. The second category of works considered an energy harvesting wireless sensor network, where sensors harvest energy from an ambient energy source and aims to trade-off energy usage and sensing accuracy. In the aforementioned works, one main challenge is the uncertain amount of harvested energy. In this respect, a fusion center/scheduler has to select the best set of sensors without knowing their current energy level.

The first category of works has investigated selecting devices/users and considered simultaneous wireless information and power transfer (SWIPT) [54, 103]. A key problem of this category of works is to consider whether devices should receive data or harvest energy. For example, in [106], [107], [108], a device is selected to receive data from a fusion center, and the other devices will harvest energy from the RF-signal broadcasted by the fusion center so as to trade-off the sum-rate and harvested energy of devices. To achieve their aim, prior works proposed strategies that select the best device according to their signal-to-noise ratio [106], [107], and perfect channel state information [108]. The work in [109] and [110] considers devices equipped with a power splitting unit. That is, a portion of a received signal is used for information decoding and another portion is used for energy harvesting. One device is selected to receive data and energy from a fusion center, and the other devices only harvest energy from the RF-signal broadcasted by the fusion center so as to trade-off the sum rate and harvested energy of devices. In order to achieve their aim, the authors of [109] and [110] proposed strategies that select the best device according to achievable data rate and harvested energy at devices. Different from [106–110], in [111], multiple devices/users are selected to receive data and other devices/users harvest energy to trade-off information transmission and energy harvesting. To select the best set of devices/uses, the authors of [111] proposed a so-called opportunistic communications-based user selection algorithm. The basic idea is to select devices/users according to their energy harvesting and information transmission capacity, which means the scheduler in [111] has perfect channel state information of devices/users.

The second category of work investigated selecting multiple sensors to sample and transmit data. For example, in [112, 113], the authors considered device selection in a heterogeneous sensor network. Specifically, in [112], there are multiple nodes. Each node is responsible for multiple environmental parameters. Each node consists of multiple sensors, and each sensor is responsible for sensing one environment parameter. In [112], the proposed solution selects (i) the best set of nodes, and (ii) the best set of sensors so as to trade-off sensing quality and energy efficiency. To achieve their aim, the solution uses a so-called adaptive multi-sensing (MS) strategy. Its basic idea is to select sensors according to spatio-temporal dynamics. In [113], the authors extended their work in [112] and proposed two new strategies: adaptive Multi-Sensing Spatial Proximity (MS-SP) and adaptive Multi-Sensing Cross-Correlation (MS-CC). These two strategies further reduce the number of active sensors, i.e., selected sensors, according to their location and sensed environmental parameters, to further reduce energy consumption.

There are a set of works, i.e., [114–118], that have considered a single target tracking problem and a network that consists of the same type of sensors. In order to select the best set of sensors so as to trade-off tracking error and energy consumption, they use an adaptive dynamic programming (ADP) algorithm [119]. The basic idea is to train an agent using an actor and critic network. In [115–117], a fusion center/sink node first predicts the state of a tracked target, i.e., the location of the target, according to the received data from sensors. It then selects the best set of sensors according to the predicted state. In [115, 116], the fusion center/sink node is only capable of predicting the state of the tracked target in the next frame. In another work, i.e., [117], the fusion center/sink node predicts the state of the tracked target in the next few frames. Different from [115–117], in [118], instead of predicting the location/trajectory of a tracked target, a fusion center predicts the amount of harvested energy and tracking performance of sensors in the next frame before ADP [119] to select sensors.

Table 2.5 summarizes works in this section. As shown in Table 2.5, only references [106], [107], [108], [109], [110], [111] have considered RF-charging. However, in the aforementioned works, the proposed solutions do not consider transmissions from devices/users to a fusion center, and they are centralized and run by a fusion center/scheduler. Lastly, these works assumed a fusion center has perfect channel state information of each device, which is impractical in a large-scale network since obtaining channel state information requires the fusion center to poll all devices.

Paper	**Heterogeneous wireless sensor network**	**SWIPT**	**Energy source**	**Information at fusion center**	**Solution**
Morsi et al. [106]	No	Yes	RF signal	Channel state of each device	Online algorithm
Morsi et al. [107]	No	Yes	RF signal	Channel state of each device	Online algorithm
Chynonona et al. [108]	No	Yes	RF signal	Channel state of each device	Online algorithm
Bang et al. [109]	No	Yes	RF signal	Channel state of each device	Adaptive multiuser scheduling algorithm
Kim et al. [110]	No	Yes	RF signal	Channel state of each device	Adaptive proportional scheduling algorithm
Zhao et al. [111]	No	Yes	RF signal	Channel state of each device	Opportunistic communications -based user selection algorithm
Gupta et al. [112]	Yes	No	Solar	Energy level of sensors in past frames	Adaptive multi-sensing algorithm
Song et al. [114]	No	No	Solar	None	Adaptive dynamic programming algorithm
Gupta et al. [113]	Yes	No	Solar	Energy level of sensors in past frames	Adaptive multi-sensing spatial proximity algorithm
Liu et al. [115], [116]	No	No	Solar	Location of sensors	Adaptive dynamic programming-based multi-sensor scheduling algorithm
Liu et al. [117]	No	No	Solar	Location of sensors	Multistep prediction-based adaptive dynamic programming algorithm
Jiang et al. [118]	No	No	Solar	None	Finite-horizon adaptive dynamic programming algorithm

Table 2.5: A comparison of works that study device selection and consider multi-objective optimization.

2.1.6 Others

This section presents device selection works that consider different performance functions, for example, spectral efficiency. This section classifies works into two categories according to their energy source, i.e., (i) works that have considered devices that harvest energy from the transmission of a transmitter/HAP, e.g., a WPCN [120–124], and (ii) works that have studied devices that harvest renewable energy such as solar or wind [125].

The first category of works can be further classified into two groups: (i) works that considered a harvest-then-transmit strategy [56], and (ii) works that studied SWIPT. A key problem addressed in the first set of works is to find the set of devices that harvest the most energy and achieve the highest data rate. For example, in [120], a HAP selects devices to receive energy from a fusion center or HAP. They then transmit data to the fusion center so as to maximize spectral efficiency. Moreover, the HAP aims to ensure fairness among devices. To this end, the harvesting-constrained scheduling scheme in [120] first selects devices that are capable of harvesting sufficient energy for transmission. It then selects devices either according to a greedy or round-robin strategy. In works such as [121] and [122], all devices harvest energy from RF-signals broadcasted by an energy transmitter and transmit data to an information receiver that is located in a different place with an energy transmitter. In each slot, the k-best devices are selected to transmit data to a receiver so as to minimize the outage probability at an information receiver. In order to achieve their aim, the proposed solution in [121] and [122] select devices according to the energy level, channel state information, and signal-to-noise (SNR) ratio at information receivers, see details in [121] and [122].

A key problem is to consider whether devices should harvest energy or transmit their data. For example, in [123], there is a multi-user orthogonal frequency division multiplexing (OFDM) system with an AP and multiple devices/users. The AP (i) selects one device to receive data and other devices harvest energy from its RF-signal, and (ii) allocates energy for each sub-carrier to maximize energy efficiency. The AP runs an algorithm that selects the best device and allocates energy according to known channel state information. In a different work, devices in [124] are equipped with a power splitting unit that divides the power of a received signal into two parts. A portion of the signal is used for information decoding and another portion is used for energy harvesting. The authors of [124] focused on selecting an area sector or a set of devices to receive energy and polling information simultaneously and then transmit data according to time division multiplexing (TDMA) to a fusion center. The problem at hand is to minimize energy outage probability and ensure the age of information (AoI) of devices is below a threshold. This problem is modeled as a Constrained Markov Decision Process (CMDP) that is then solved using a linear program.

Different from the first category of works, a key problem in the second category of works is to consider whether devices/sensors have sufficient energy to sense and transmit data to a fusion center. For example, in [125], the problem is to select a set of devices/sensors to sense their environment and transmit data so as to optimize average sensing utility while considering the energy budget of devices/sensors. A myopic policy is proposed to select devices according to the energy level of devices/sensors.

As shown in Table 2.6, all works required either channel state information or energy level of devices. However, as discussed in Section 2.1.5, obtaining the aforementioned information of all devices is impractical in large-scale networks since this requires a scheduler to poll all devices/sensors. Moreover, only reference [124] has considered RF charging and TDMA. However, it does not use a reinforcement learning approach and it is not a distributed approach.

Paper	SWIPT	TDMA	WPCN	Known information	Solution
Tabassum et al. [120]	No	No	Yes	Channel state information	Harvesting constrained scheduling scheme
Dimitropoulou et al. [121] and [122]	No	No	Yes	Channel state and energy level of each devices	Select devices according to known information
Kwan et al. [123]	Yes	No	Yes	Channel state information	Select devices according to channel state information
Ko et al. [124]	Yes	Yes	Yes	AoI, energy level and location of devices	Linear program
Yang et al. [125]	Yes	No	No	Energy level of devices	Myopic policy

Table 2.6: A comparison of works that study device selection and consider different performance functions with prior sections.

2.2 Summary

In summary, this chapter has reviewed device/sensor selection problems and their respective system, namely ambient, RF, or hybrid energy harvesting wireless sensor networks. The work in this thesis differs from past works in the following manners:

1. This thesis first aim to maximize sum rate. The majority of prior works that have investigated data rate maximization assumed that devices are powered by an ambient energy source, which means energy arrival at each device is independent and random. In contrast, in this thesis, devices have a dedicated charger, i.e., the hybrid access point. Secondly, past works such as [58–60, 62, 65] have considered the impact of random uplink channel gain on throughput. In this thesis, however, the amount of received data is affected by the channel gain from each transmitting device to a hybrid access point. Moreover, past works such as [59, 60, 62, 65–67] ignored battery leakage. Apart from that, in this thesis, a HAP is not required to poll devices to obtain their channel state or battery state information. Consequently, the problem considered in this thesis is more challenging than prior works.

2. This thesis then investigates the age of information minimization. In general, unlike prior works such as [73–77], [78–81] that considered devices that harvest from wind or solar energy, this thesis assumes that devices harvest energy from a HAP that broadcast an RF-signal. Further, the HAP selects the best set of devices without knowing their energy and channel state. This feature is distinct from prior works that have considered perfect information at a HAP [72], [85], [86], [82]. Apart from that, all past works have used a centralized reinforcement learning algorithm to select devices, e.g., [72], [73], [76], [77], [78], [79], [81], [82]. In contrast, this thesis proposes a decentralized reinforcement learning algorithm to select devices in a WPCN in order to determine a policy that minimizes the average age of information.

3. Lastly, this thesis investigates the optimization of a novel metric, i.e., age of incorrect information, in a WPCN. To date, there is no past work that has studied device selection and aimed to minimize the said metric for multiple targets. Moreover, all past works on device selection in a WPCN in order to improve information freshness assumed that each sensor/device only monitors one target or senses one environmental parameter [72], [74], [86], [82]. In contrast, this thesis considers one or more targets. Apart from that, there is no prior work that consider cooperate monitoring of targets in order to minimize the age of incorrect information.

Chapter 3

Throughput Maximization in RF Charging Networks with Imperfect CSI

This chapter considers an RF charging Wireless Sensor Network (WSN) that consists of a HAP and multiple EH devices with imperfect battery, i.e., energy leakage exists. The HAP does not have the battery state information and only has the imperfect uplink channel state information of devices. The main research question is to determine the set of devices that maximize throughput.

To illustrate the problem, consider the example Figure 3.1. There are three devices and a HAP. Time is slotted. HAP and devices are synchronized. The HAP aims to select two devices in each time slot. The HAP aims to maximize the sum rate over two time slots. Devices always have data to transmit. Further, in this example, devices lose all their stored energy if they do not transmit. This example only considers two channel states: good or bad. If the channel is good, a device has a link rate of 2 bps/Hz for each 1J of energy. Otherwise, the link operates at 1 bps/Hz for the same amount of energy. This example assumes that D_2 and D_3 have a good channel, while D_1's channel is bad. In a charging slot, the received energy of D_1, D_2 and D_3 is 1J, 2J, and 4J, respectively. Consider a round-robin device selection policy, where devices are selected in turn by the HAP. The HAP first selects device D_1 and D_2. They use all their energy to transmit data. The link rate of D_1 and D_2 is respectively 1 bits/Hz and 4 bps/Hz. After the second charging slot, D_1, D_2 and D_3 receive 1J, 2J and 4J again. The HAP then selects D_3 and D_1 to transmit in the second time slot. Their respective link rate is 1 bps/Hz and 8 bps/Hz. The sum rate over two time slots is thus 14 bps/Hz. Notice that in the given two time slots, device D_3 and D_2 always have a higher energy level and better channel state than D_1. This means D_3 and D_2 are capable of producing a higher rate than device D_1. Consequently, a better selection policy is to select D_2 and D_3 in the first and the second time slot. The corresponding sum rate of the best policy is 24 bps/Hz. However, as mentioned, the HAP is unaware of the channel state nor battery level of devices, which complicates the device selection process.

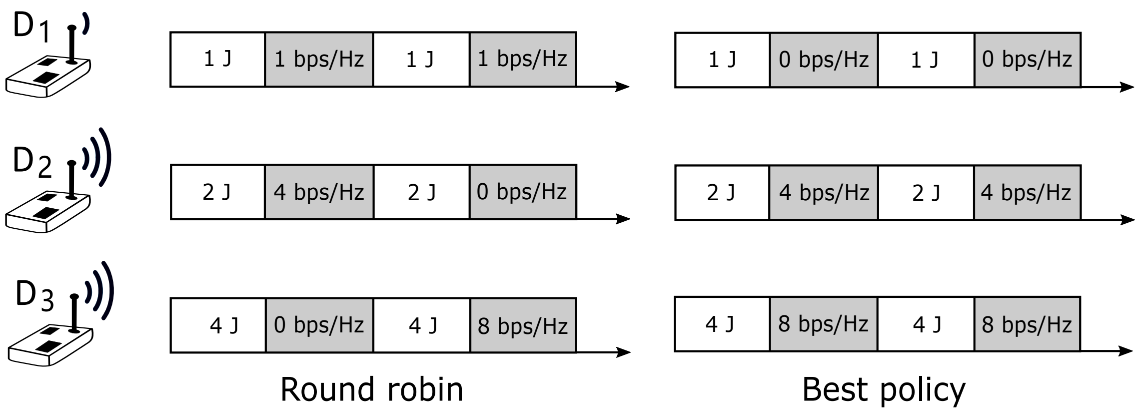

Figure 3.1: An example of device selection. The channel condition to/from device D_1 is poor, while that of D_2 and D_3 is good.

From the example, the problem at the hand is to determine which set of devices has the highest throughput in each frame. The are two main challenges. Firstly, the HAP/scheduler does not have perfect channel state information nor the energy level of devices. This is reasonable since it is impractical for the HAP to poll devices for the aforementioned information in a large-scale network. Secondly, the HAP has to take into consideration energy leakage at each device in order to avoid energy wastage.

The remainder of this chapter is organized as follows. Section 3.1 formalizes the RF-energy harvesting network and battery leakage model under consideration. This is followed by the problem formulation in Section 3.2. After that, Section 3.2.1 lists several properties concerning the problem. Section 3.3 and Section 3.4 outline the proposed approaches for the problem. The results are presented in Section 3.5. Finally, Section 3.6 concludes this chapter.

3.1 System Model

Let $N = \{D_1, D_2, \ldots, |N|\}$ be a set of devices. These devices are placed randomly around a HAP as shown in Figure 3.2. The HAP is responsible for charging via RF these $|N|$ devices and collecting data from them. Devices always have data to transmit. Time is discrete and each time slot is indexed by t. There are T time slots; each has a duration of one second. Each time slot is divided into a fixed charging slot and K data slots, where $K \ll N$; see Figure 3.2. The size of the charging slot is τ_C. Let us denote the data slots of time slot t as $\widehat{s^t_1}, \widehat{s^t_2}, \ldots, \widehat{s^t_K}$ and set the data slot size to $\tau_D = \frac{1-\tau_C}{K}$. In each charging slot, the HAP will transmit with power P^t (in Watts) [126]. After the charging slot, the HAP will select K devices to transmit data. Let $I^t_i \in \{1, 0\}$ denote whether device D_i is selected in time slot t. Specifically, if the HAP selects D_i to transmit in time slot t, then $I^t_i = 1$. Otherwise, $I^t_i = 0$. Let $s^t \in \{0, 1\}^N$ be a binary vector that has exactly K selected devices, denoted with a value of one, in time t. Formally, $s^t = \{I^t_1, I^t_2, \ldots, I^t_N\}$. Note that for each time slot t, there are $\binom{N}{K}$ possible number of such binary vectors. Let us define a *schedule* indexed by z, as $S_z = \{s^t \mid t = 1, \ldots, T\}$; i.e., the set of binary vector selected by the HAP in time slot $t = 1$ to $t = T$. The collection of schedules is denoted as $\widehat{\mathcal{S}}$, which has size $|\widehat{\mathcal{S}}|$, which is bounded by $\binom{N}{K}^T$. The channel gain between a device D_i and the HAP is denoted as g^t_{i0}, where 0 denotes the HAP. The channel gain from the HAP to a device D_i is g^t_{0i}. The path loss between the HAP and devices follow

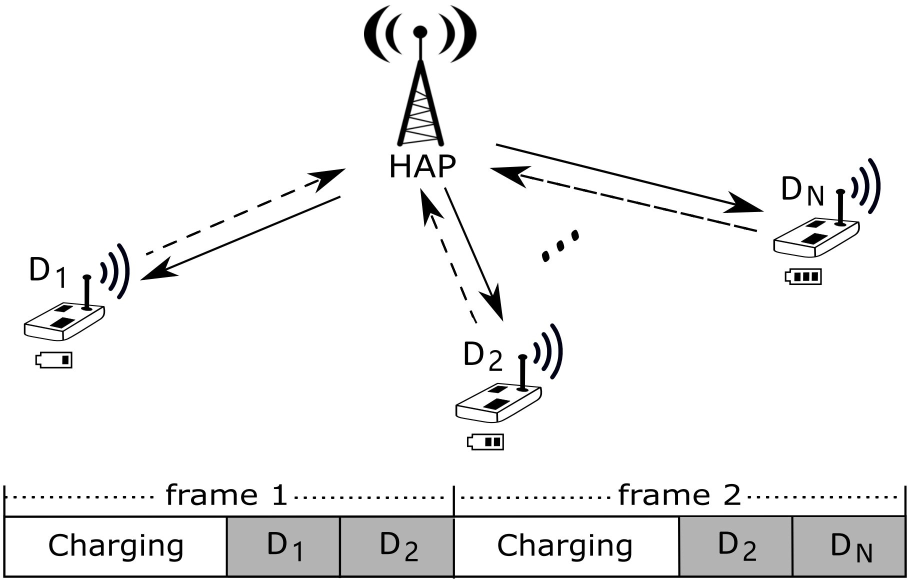

Figure 3.2: An RF charging network and time slots.

the Log-distance model. Thus, the channel gain is calculated as

$$\begin{cases} PL\left(d_i\right)[dB] = PL\left(d_0\right) + 10\beta log_{10}\left(\frac{d_i}{d_0}\right) + \mathcal{X}, \\ g_{i0}^t = 10^{-\frac{PL(d_i)}{10}}, \end{cases} \tag{3.1}$$

where d_i is the distance between the device D_i and the HAP, $PL(d_0)$ is the path loss at a reference distance, β is the path loss exponent. The term $\mathcal{X}$ is a zero mean Gaussian distributed random variable (in dB) with standard deviation μ to reflect shadowing effect.

The HAP has only past CSI and not the instantaneous CSI of devices. This is reasonable because in practice collecting CSI requires the HAP to first charge devices before collecting reply to pilot signals. This becomes a challenge when devices have varying channel gains and may not receive sufficient energy to respond to pilot signals, or when there are many devices in which the HAP has to send pilot signals. Lastly, CSI remains constant within a time slot but varies across time slots.

Devices have an RF energy harvester with a conversion efficiency of $\eta \in [0, 1]$. Note that the RF conversion efficiency is non-linear and it is a function of the received power [127]. Let E_i^t denote the energy level of device D_i at the beginning of time slot t. Device D_i receives $e_i^t = P^t \eta \; g_{0i}^t \tau_C$ amounts of energy in the charging slot of time slot t. Each device has a battery with a capacity of B_{max}. Once a device's battery reaches its capacity, any excess energy is discarded. In addition, the battery of devices leaks at a constant rate of ϱ in each time slot. The battery storage of each device D_i thus evolves as follows:

$$E_i^{t+1} = \begin{cases} (1-\varrho)\left(MIN\left(B_{max}, E_i^t + e_i^t\right)\right), & I_i^t = 0, \\ 0, & I_i^t = 1. \end{cases} \tag{3.2}$$

Without loss of generality, device uses all its stored energy to transmit data if it is selected by the HAP in a given time slot.

Note, in practice, a device may allocate some of its harvested energy for sampling. A selected device then uses the transmit power

$$p_i^t = \frac{E_i^t + e_i^t}{\tau_D}. \tag{3.3}$$

The data rate (in bit/s/Hz) of device D_i in time slot t is

$$r_i^t(\mathcal{S}_z) = \begin{cases} 0, & I_i^t = 0, \\ \tau_D log_2\left(1 + p_i^t g_{i0}^t \frac{1}{\sigma^2}\right), & I_i^t = 1, \end{cases} \tag{3.4}$$

where σ^2 is the ambient noise power. From here onward, define the *reward* as the sum-rate over T time slots as

$$R(\mathcal{S}_z) = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N} r_i^t(\mathcal{S}_z). \tag{3.5}$$

Instead of the average sum-rate, an alternative reward definition is the minimum data rate of devices. Let $r_i(\mathcal{S}_z)$ be the average transmission rate of device D_i over

T time slots, which is given by

$$r_i(\mathcal{S}_z) = \frac{1}{T}\sum_{t=1}^{T} r_i^t(\mathcal{S}_z). \tag{3.6}$$

The new reward that considers the minimum data rate of devices is then defined as

$$R_i(\mathcal{S}_z) = \text{MIN}\left\{r_i(\mathcal{S}_z) \mid i \in \mathcal{D}\right\}, \tag{3.7}$$

where $\mathcal{D}$ is the set of devices. This reward ensures all devices have a non-zero throughput. This is important for sensing applications that require at least a sample from *all* sensor devices. This chapter will use Eq. (3.5) to show the efficacy of the proposed approaches. All notations are summarized in Table 3.1.

Notation	**Description**
T	The number of time slots.
K	The number of data slots in one time slot.
$\mathcal{X}$	A Gaussian distributed random variable.
P^t	The transmission power of the HAP.
B_{max}	Battery capacity.
E_i^t	Energy level of device D_i at the beginning of time slot t.
τ_C	Size of the charging slot.
τ_D	Size of the data slot.
g_{i0}^t	The uplink channel gain between the HAP and device D_i.
g_{0i}^t	The downlink channel gain between the HAP and device D_i.
p_i^t	Transmit power of device D_i at time slot t.
η	Energy conversion efficiency.
β	Path loss exponent.
σ^2	Noise power.
ϱ	Battery leakage rate.
R	The sum-rate.
R_i	Average transmission rate of device i.
$\mathcal{S}_z$	A schedule.
$\mathcal{D}$	The set of devices.

Table 3.1: Key notations used in this chapter.

3.2 The Problem

The aim is to maximize the average sum rate over a planning time horizon T. To do this, the HAP needs to determine a schedule $\mathcal{S}_z$, which selects which set of devices to transmit in each time slot. Formally, the problem is

$$\max_{S \in \widehat{\mathcal{S}}} \quad \mathbb{E}_{\varphi}\left[R(\mathcal{S})\right], \tag{3.8}$$

where we maximize over the joint distribution of channel gains to/from each sensor device in N.

There are two challenges when solving the aforementioned problem. First, the size of $\widehat{\mathcal{S}}$ grows exponentially with higher T and K values. Second, the HAP does not have instantaneous CSI information. This chapter addresses both of these problems using cross-entropy [128] and Gibb sampling [129]. The details of each method are outlined in Section 3.3 and 3.4, respectively.

3.2.1 Analysis

To gain some insights into the problem, this section conducts an analysis on (i) the sum rate over T time slots, (ii) the sum-rate gap between a random devices selection policy and the optimal solution, and (iii) the optimality of the Round Robin (RR) policy. In the analysis to follow, it is assumed that devices receive ϵ worth of energy in each charging slot and each device has the same uplink channel gain g.

Proposition 1. *Given data slot size τ_D, the total number of devices N, the number of selected device K in each slot, and the battery leakage rate ϱ, the sum-rate R over T time slots satisfies,*

$$\begin{aligned} &KT\tau_D log_2\left(1+\frac{g\epsilon}{\tau_D\sigma^2}\right) \leq R \leq \\ &\sum_{t=1}^{T} K\tau_D log_2\left(1+\frac{g\epsilon[1-(1-\varrho)^t]}{\varrho\tau_D\sigma^2}\right). \end{aligned} \tag{3.9}$$

Proof. The analysis first shows the lower bound. The worst case occurs when the same K devices are selected in each time slot. This means that these K devices are only able to accumulate ϵ worth of energy. This implies that the transmit power and transmission rate of each selected device is $\frac{\epsilon}{\tau_D}$ and $\tau_D log_2(1+\frac{g\epsilon}{\tau_D\sigma^2})$ respectively.

Consequently, the lower bound of the sum rate is $KT\tau_D log_2(1+\frac{g\epsilon}{\tau_D\sigma^2})$. For the upper bound, the best case is when the HAP queries K devices that have never been transmitted before in each time slot. This means that in time slot t, each selected device has $\frac{\epsilon[1-(1-\varrho)^t]}{\varrho}$ worth of energy. This means the maximum sum-rate of time slot t is $K\tau_D log_2(1+\frac{g\epsilon[1-(1-\varrho)^t]}{\varrho\tau_D\sigma^2})$. Consequently, the upper bound over T time slots is $\sum_{t=1}^{T} K\tau_D log_2(1+\frac{g\epsilon[1-(1-\varrho)^t]}{\varrho\tau_D\sigma^2})$, as desired. □

Given the previous proposition, a gap can be found between the maximum sum rate and the performance of a random policy whereby the HAP selects K devices randomly.

Corollary 1. *The maximum sum-rate gap between the random policy and the optimal solution is* $\sum_{t=1}^{T} K\tau_D log2(\frac{\varrho\tau_D\sigma^2+g\epsilon[1-(1-\varrho)^t]}{\varrho(\tau_D\sigma^2+g\epsilon)})$.

Proof. In the worst case, the random policy selects the same set of devices in each time slot, and thus obtains the lower bound of Proposition-1. Consequently, the gap in sum rate is simply the difference between the upper and lower bound of Proposition-1. Specifically, in time slot t, the difference between the upper and lower bound of sum rate is $K\tau_D log_2(1+\frac{g\epsilon[1-(1-\varrho)^t]}{\varrho\tau_D\sigma^2}) - K\tau_D log_2(1+\frac{g\epsilon}{\tau_D\sigma^2})$. Consequently, over T time slot, the maximum gap in sum-rate is $\sum_{t=1}^{T} K\tau_D log_2(\frac{\varrho\tau_D\sigma^2+g\epsilon[1-(1-\varrho)^t]}{\varrho(\tau_D\sigma^2+g\epsilon)})$. □

The previous proposition assumes that the HAP is able to query K new devices in each slot. The next proposition relaxes this assumption and shows that the Round Robin (RR) policy, where the HAP ensures each device has equal opportunity to transmit, is optimal.

Proposition 2. *Let* $\Delta = \lceil \frac{N}{K} \rceil$ *be an integer. Define* $\hat{\Delta} = 1 + \frac{g\epsilon[1-(1-\varrho)^{\Delta}]}{\varrho\tau_D\sigma^2}$ *and* $\bar{T} = T - \Delta$*. Then, the Round Robin policy is optimal.*

Proof. This section first proves the sum rate from slot $\Delta + 1$ to T is optimal. After a device is queried, it at most has Δ time slots to accumulate $\frac{\epsilon[1-(1-\varrho)^{\Delta}]}{\varrho}$ worth of energy before it is queried again. This means in each slot the maximum sum-rate is $K\tau_D log(1 + \frac{g\epsilon[1-(1-\varrho)^{\Delta}]}{\varrho\tau_D\sigma^2})$. In particular, the sum-rate over $(T - \Delta)$ time slots, i.e., $\bar{T}K\tau_D log(\hat{\Delta})$ or $\tau_D log(\hat{\Delta}^{\bar{T}K})$, is optimal. To see this, recall the Arithmetic-Mean Geometric-Mean (AM-GM) inequality,

$$\left(\frac{z_1 + z_2 + \cdots + z_n}{n}\right)^n \geq z_1 z_2 \ldots z_n. \tag{3.10}$$

The maximum value on the right hand side is attained when $z_1 = z_2 = \cdots = z_n$. Using the AM-GM inequality and the assumption that all $\hat{\Delta}$ value are equal, it has

$$\hat{\Delta}^{\bar{T}K} = \left(\frac{\hat{\Delta} + \hat{\Delta} + \cdots + \hat{\Delta}}{\bar{T}K}\right)^{\bar{T}K}. \tag{3.11}$$

This implies that the RR policy achieves the highest sum-rate value over $\Delta + 1$ to T time slots. Next, considers the sum rate of the first Δ time slots. In each time slot $t = 1, 2, \ldots, \Delta$, the RR policy selects K devices that have yet to transmit. This means each selected device accumulates $\frac{\epsilon[1-(1-\varrho)^{t}]}{\varrho}$ of energy for data transmission. This produces the sum-rate of $K\tau_D log(1+\frac{g\epsilon[1-(1-\varrho)^{t}]}{\varrho\tau_D\sigma^2})$ in time slot t. By the AM-GM inequality, the terms in the logarithm $K\tau_D log((1 + \frac{g\epsilon[1-(1-\varrho)^{1}]}{\varrho\tau_D\sigma^2}) \times (1 + \frac{g\epsilon[1-(1-\varrho)^{2}]}{\varrho\tau_D\sigma^2}) \times \ldots \times (1 + \frac{g\epsilon[1-(1-\varrho)^{\Delta}]}{\varrho\tau_D\sigma^2}))$ must be similar or equal to yield the highest sum-rate. This means if the HAP uses another policy that delays querying devices to allow them to accumulate more energy, there will be a larger discrepancy between the energy of devices, meaning that the sum rate of such a policy will be lower than that of the RR policy. This concludes the proof. $\square$

3.3 A Cross Entropy (CE) Algorithm

Recall that the problem is to identify a schedule $\mathcal{S}$, that maximizes reward (3.5). To this end, this chapter will use CE to construct and select the best schedule. As we will see later, CE maintains a probability distribution that is then adapted iteratively to determine the transmitting device in each slot.

Briefly, CE was originally proposed to estimate the occurrence probability of rare events in a stochastic network [130]. Then, the authors of [131] adapted CE to solve combinatorial optimization problems. CE operates iteratively. In each iteration, it has two main phases: (i) it generates samples according to an initial Probability Distribution Function (PDF) or Probability Mass Function (PMF), and (ii) it then evaluates these samples and identifies so-called *elite* samples. Then, it updates the parameters of the PDF or PMF using the statistics of these *elite* samples. Specifically, CE first generates J samples according to an initial distribution. Let us denote each sample j as x_j, where $j = 1, 2, \ldots, J$. Each sample has a so-called reward, which is denoted as $\mathbb{S}(x_j)$. Next, CE sorts the J samples according to their reward in non-decreasing order. Then it identifies the samples in the $(1-\rho)$-th percentile, where $\rho \in [0, 1]$. Let the reward of the $(1-\rho)$-th sample be δ. CE then records all samples with a reward that is higher than δ; these are the so-called elite samples, which is denoted by the set X_ρ. Lastly, CE uses the statistics of elite samples to update the parameters of the PDF or PMF. CE repeats the said two phases until convergence.

Algorithm 3.1 details the CE-based approach. The parameter α determines the learning rate of CE. Specifically, it controls how fast the vector $\mathcal{B}^c$ changes in each iteration. The sets X^c and X_ρ^c are respectively used to store the samples and elite samples in the c-th iteration. The vector $\mathcal{B}^c$ is a multivariate Bernoulli distribution that is used to generate sample or schedule x_j in c-th iteration, i.e., $x_j \sim Ber(\mathcal{B}^c)$. The distribution $\mathcal{B}^c$ indicates the success/failure probability of element I_i^t in schedule x_j at iteration c. Initially, the CE algorithm sets all elements in $\mathcal{B}^c$ to 0.5, i.e.,

Algorithm 3.1: A CE-based algorithm for devices selection.

```
Output: 𝓑^c
Initialize: 𝓑^c = (0.5, 0.5, ..., 0.5), c = 1, δ, α, X^c, X_ρ^c
while c > 1 AND not converge do
    X^c = ∅, X_ρ^c = ∅
    for j ← 1 to J do
        Generate x_j ~ Ber(𝓑^c)
        Store x_j in X^c
        Calculate 𝕊(x_j)
    end
    Ŝ = Sort (𝕊(x_1), ..., 𝕊(x_j))
    δ^c = Percentile((1 − ρ), Ŝ)
    for each x_j ∈ X^c do
        if 𝕊(x_j) ≥ δ^c then
            Store x_j in X_ρ^c
        else
            Ignore x_j
        end
    end
    for n ← 1 to |𝓑^c| do
        Calculate 𝓑_n^c as per Eq. 3.12
        𝓑_n^c = α𝓑_n^c + (1 − α)𝓑_n^{c−1}
    end
    c ← c + 1
end
return 𝓑^c
```

$\mathcal{B}^1 = (0.5, 0.5, \ldots, 0.5)$, which means each device has the same probability to be selected, see line 1. Let $\mathcal{B}^c_n$ denote the n-th item of the vector $\mathcal{B}^c$. In line 5 - 7, the algorithm generates J samples, i.e., schedules, and calculates their corresponding reward or sum rate. In line 9, CE sorts the reward of samples in ascending order: $\mathbb{S}(1) \leq \ldots \leq \mathbb{S}(J)$. Let us denote the sorted list as $\hat{\mathbb{S}}$ and define $Percentile((1-\rho), \hat{\mathbb{S}})$ as a function that calculates a threshold to identify elite samples, i.e., samples that belong to the $(1-\rho)$-th percentile value of $\hat{\mathbb{S}}$. In line 10 we see that CE sets the threshold δ^c to $Percentile((1-\rho), \hat{\mathbb{S}})$. Next, CE identifies elite samples according to threshold δ^c and sample rewards. We see that, in line 12-14, CE collects samples with a reward larger than δ^c and stores them in the set X_ρ. After that, as CE seeks to generate better samples in the next iteration, it updates each element of the multivariate Bernoulli distribution $\mathcal{B}^c$ according to X_ρ. The update formula is given as

$$\mathcal{B}^c_n = \frac{\sum_{j=1}^{J} I_{\{\mathbb{S}(x_j) \geq \delta^c\}} I_{\{x_{j,n}=1\}}}{\sum_{j=1}^{J} I_{\{\mathbb{S}(x_j) \geq \delta^c\}}}, \tag{3.12}$$

where $x_{j,n}$ is the n-th element of the sample x_j. The denominator corresponds to the number of elite samples. The numerator is equal to the number of elite samples where the n-th element is equal to one. Instead of updating the value of $\mathcal{B}^c_n$ directly via the solution of (3.12), CE uses a smoothing process, see line 20. Specifically, the value of $\mathcal{B}^c_n$ is equal to the weighted average of the solution of (3.12) and the value of $\mathcal{B}^{c-1}_n$. CE converges when the value of each element of vector $\mathcal{B}^c$ is within a tolerance, i.e., 0.01, away from one or zero.

3.4 A Gibbs Sampling Based Algorithm

The next solution is based on Gibbs sampling, which allows us to efficiently sample possible optimal schedules. Gibbs sampling was originally proposed to generate samples from a joint probability distribution indirectly [132]. Let $p(\Theta)$ be a joint probability distribution, where $\Theta = (\theta_1, \theta_2, \ldots, \theta_\omega)$ is a random vector with ω elements. Gibbs sampling can be used to generate Θ according to the conditional probability $p(\theta_a|\Theta_{-a})$, where Θ_{-a} represents the set of all elements of Θ except element θ_a. The Gibbs sampling process can be viewed as the construction of a Markov chain with multiple states. Each state is a possible vector Θ. The transition probability between each state follows the conditional probability distribution $p(\theta_a|\Theta_{-a})$. The steady-state distribution of the Markov chain is given by $p(\Theta)$. Let $\Theta^{(m)}$ be the m-th sample or state of the Markov chain. Gibbs sampling has two main steps: (i) initially, it randomly generates a vector or sample $\Theta^{(1)}$, and (ii) after that, it updates each element of $\Theta^{(m+1)}$ one by one from $\theta_1^{(m+1)}$ to $\theta_\omega^{(m+1)}$. Let $\theta_a^{(m+1)}$ denote the element to be updated in the sample $\Theta^{(m+1)}$. It has $\Theta_{-a}^{(m+1)} = \{\theta_1^{(m+1)}, \theta_2^{(m+1)}, \ldots, \theta_{a-1}^{(m+1)}, \theta_{a+1}^{(m)}, \ldots, \theta_\omega^{(m)}\}$. That is, the first $a-1$ elements of $\Theta_{-a}^{(m+1)}$ have been updated. The remaining $\omega - a$ elements of $\Theta_{-a}^{(m+1)}$ are those in the vector $\Theta^{(m)}$. In particular, Gibbs sampling produces $\theta_a^{(m+1)}$ according to the conditional probability $p(\theta_a^{(m+1)} \mid \theta_1^{(m+1)}, \theta_2^{(m+1)}, \ldots, \theta_{a-1}^{(m+1)}, \theta_{a+1}^{(m)}, \ldots, \theta_\omega^{(m)})$. Step (ii) is then repeated M times to obtain the sample $\Theta^{(M)}$. As M grows larger, i.e., $M \to \infty$, the sample $\Theta^{(M)}$ converges to the joint probability distribution $p(\Theta)$.

In the second approach, called Gibbs$^+$, let ξ^t be the set of K selected devices in time slot t. Formally, $\xi^t = \{i|i \in N \wedge I_i^t = 1\}$. It, then, has samples or schedules Θ containing T sets of type ξ^t. Each sample Θ can then be denoted as $\Theta = \{\xi^t \mid t = 1, 2, \ldots, T\}$. In the m-th iteration, Gibbs$^+$ updates each ξ^t of $\Theta^{(m)}$ according to the conditional probability $p(\xi^t|\Theta_{-t})$, i.e., $\xi^t \sim p(\xi^t|\Theta_{-t})$. The distribution $p(\xi^t|\Theta_{-t})$ indicates the probability that the HAP chooses ξ^t as the set of selected devices at time slot t when other elements are fixed in the schedule $\Theta^{(m)}$. The m-th iteration ends after Gibbs$^+$ updates the last element ξ^t of $\Theta^{(m)}$.

Now we are ready to illustrate how Gibbs$^+$ updates the set ξ^t in the m-th iteration. As before, the HAP selects K devices in each time slot t. Consequently, there are $N - K$ devices that are not in ξ^t. Let the set N^t contain these $N - K$ devices, where $N^t = \{i \mid i \in N \wedge i \notin \xi^t\}$. In time slot t, Gibbs$^+$ removes the device with the lowest throughput from the set ξ^t. Let $\mathcal{R}$ be the removed device. Gibbs inserts $\mathcal{R}$ into the set N^t. After that, Gibbs$^+$ inserts a device k from the set N^t

into ξ^t. This creates $|N^t|$ different selected device sets which are denoted as $\xi^t_{\mathcal{R},k}$. Gibbs$^+$ uses $|N^t|$ different $\xi^t_{\mathcal{R},k}$ to replace ξ^t in schedule $\Theta^{(m)}$ and obtains $|N^t|$ new schedules. Let $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ be the new schedule in which the t-th element is $\xi^t_{\mathcal{R},k}$. Let $P(\xi^t_{\mathcal{R},k}|\Theta^{(m)}_{-t})$ be the probability that Gibbs$^+$ updates ξ^t as $\xi^t_{\mathcal{R},k}$. Next, Gibbs$^+$ determines $P(\xi^t_{\mathcal{R},k}|\Theta^{(m)}_{-t})$ according to the reward or sum-rate of schedule $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$. Let $R(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ denote the reward of schedule $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$. The formula to calculate $P(\xi^t_{\mathcal{R},k}|\Theta^{(m)}_{-t})$ is given as [129]:

$$P\left(\xi^t_{\mathcal{R},k}|\Theta^{(m)}_{-t}\right) = \frac{exp\left(\gamma R\left(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t}\right)\right)}{\sum_{\forall \xi^t_{\mathcal{R},k}} exp\left(\gamma R\left(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t}\right)\right)}, \tag{3.13}$$

where γ is a fixed parameter that is larger than zero. When γ is large, as per Eq. (3.13), there is a high probability that Gibbs$^+$ replaces ξ^t with $\xi^t_{\mathcal{R},k}$.

Algorithm 3.2: A Gibbs-based algorithm for selecting devices

Output: $\Theta^{(m)}$
Initialize: $\Theta^{(m)} = \{\xi^t | t = 1, 2, \dots, T\}, m = 1, M$
while $m < M$ **do**
  Run schedule $\Theta^{(m)}$ and record r^t_i
  for *each time slot* t **do**
    $\hat{\Theta} = \hat{R} = \emptyset$
    $N^t = \{i \mid i \in N \wedge i \notin \xi^t\}$
    Remove the device with the lowest r^t_i in ξ^t
    Insert $\mathcal{R}$ in N^t
    for *each device* $k \in N^t$ **do**
      $\xi^t_{\mathcal{R},k} = \xi^t \cup k$
      Obtain schedule $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$
      Store $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ in $\hat{\Theta}$
      Run schedule $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ for ζ times
      Store the average $R(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ in $\hat{R}$
      Calculate $P(\xi^t_{\mathcal{R},k}|\Theta^{(m)}_{-t})$ as per Eq.(3.13)
    end
    Update ξ^t as $\xi^t_{\mathcal{R},k}$
  end
  $m \leftarrow m + 1$
end
return $\Theta^{(m)}$

Now we are ready to present Gibbs$^+$ in its entirety; see Algorithm 3.2. Initially, Gibbs$^+$ set m to one and randomly generates a schedule $\Theta^{(m)}$, i.e., Gibbs$^+$ randomly selects K devices in each time slot, see line 1. In line 3, Gibbs$^+$ or the HAP uses schedule $\Theta^{(m)}$ and records the corresponding throughput of each device. For each time slot t, Gibbs$^+$ first sets $\hat{\Theta}$ and $\hat{R}$ to the empty set. These sets will respectively be used to record schedules generated by Gibbs$^+$ and their corresponding reward or sum rate. It then selects devices that have yet to be selected in time slot t to construct the set N^t, see line 5-6. In line 7-8, Gibbs$^+$ removes the device with the lowest throughput in ξ^t and inserts the device into the set N^t. In line 10-15, Gibbs$^+$ selects a device from the set N^t to replace $\mathcal{R}$ in the set ξ^t. This creates an alternative schedule $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$, which is stored in the set $\hat{\Theta}$. Gibbs$^+$ runs $(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ for ζ times, and calculates the average reward $R(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ and stores $R(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ in the set $\hat{R}$. Then, Gibbs$^+$, calculates the probability $P(\xi^t_{\mathcal{R},k}|\Theta^{(m)}_{-t})$ using the reward $R(\xi^t_{\mathcal{R},k}, \Theta^{(m)}_{-t})$ and (3.13). In line 17, Gibbs$^+$ selects $\xi^t_{\mathcal{R},k}$ to replace ξ^t as per the probability computed by (3.13). Gibbs$^+$ repeats line 3 to 19 until $m = M$.

3.5 Evaluation

The proposed algorithms are evaluated in Matlab [128]. The conducted simulations consisted of ten devices and a HAP. These devices are randomly placed 1 to 6 meters from the HAP; this placement ensures devices are within the receiver sensitivity of the Powercast RF-energy harvester. The simulation study the following parameters: (i) smoothing parameter α, (ii) number of samples, (iii) transmission power P^t, (iv) number of selected devices K, (v) battery leakage rate ϱ, and (vi) standard deviation $\mathcal{X}$. The antenna gain of the HAP and devices is set to 1 dBi and 6.1 dBi, respectively as per [133]. The path loss exponent is 2.5. The standard deviation is set to one in case (i) to (v). Assume that the noise power is -80 dBm. The HAP operates in the 915 MHz frequency band. The battery of devices is initially empty and it has a maximum capacity of 1 J. There are 20 time slots. The time slot and charging slot size is set to 1 s and 0.2 s, respectively. The charging efficiency is as per the

Powercast P20110B harvester [127]. For the CE method, its parameter ρ, which controls the number of elite samples, is set to 0.01. As for Gibbs$^+$, set $\gamma = 1$. Table 3.2 summarizes parameter settings.

Table 3.2: Parameter settings in simulation.

Parameter	**Value**
The antenna gain of the HAP	1 dBi
The antenna gain of the devices	6.1 dBi
The path loss exponent β	2.5
Noise power σ^2	-80 dBm
HAP broadcast frequency	915 MHz
Battery capacity B_{max}	1 J
Number of time slots	20
Charging slot size	0.2 second
Time slot duration	1 second
Parameter ρ	0.01
Parameter γ	1

The following rules were used to benchmark against the proposed approaches:

- **Random Pick (RP)**: In each time slot, the HAP randomly selects K out of N devices to transmit data.

- **Round Robin (RR)**: The HAP selects K devices according to a fixed order to ensure each device gets equal number of turns to transmit its data.

- **Perfect Information Selection (PIS)**: The HAP will select the K devices with the highest energy level and best uplink channel to transmit data in each time slot. This means the HAP has perfect information of the energy level of devices and their uplink channel condition. Thus, PIS allows us to benchmark against the theoretical maximum sum rate.

- **Original Gibbs Sampling (OGS)**: In each time slot, OGS will randomly replace a device instead of the device that has the lowest throughput.

3.5.1 Convergence

There are five devices. This experiment studies the convergence behavior of CE and Gibbs$^+$ algorithm for 300 and 100 iterations, respectively. Referring to Figure 3.3 and Figure 3.4, the proposed CE and Gibbs$^+$ algorithms converged and outperform RR and RP after converging. The reason is that, after convergence, CE and Gibbs$^+$ do not select devices that are located far from the HAP until they accumulate sufficient energy to produce a higher throughput than those devices nearer to the HAP. While RR only selects devices in a fixed order and RP simply selects devices randomly.

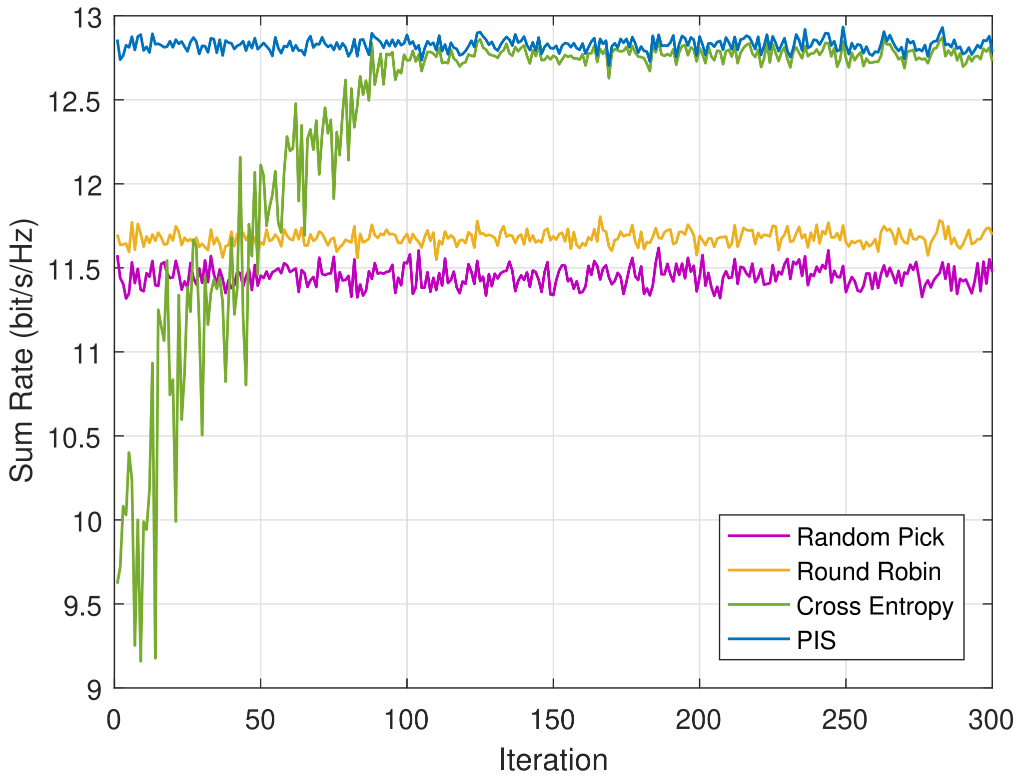

Figure 3.3: Converge curve for CE algorithm.

3.5.2 Smoothing Parameter

Here, the smoothing parameter value α has one of the following values: 0.1, 0.2, . . ., 1.0. The battery leakage rate is set to 20%. The transmit power P^t is 3 W. The

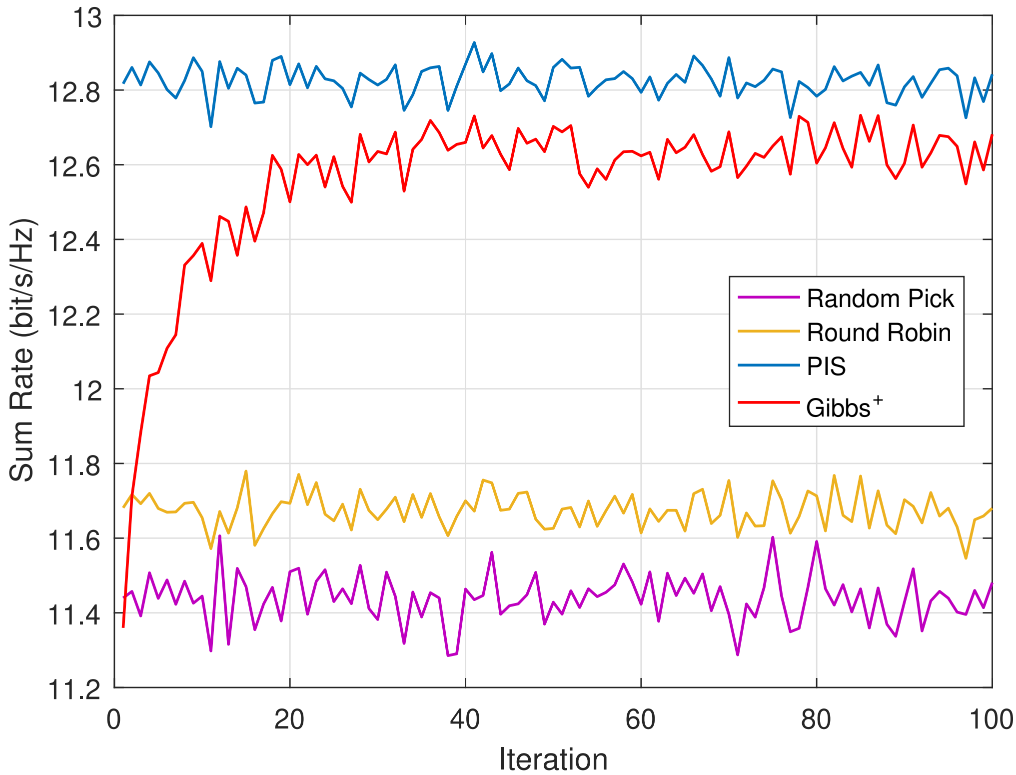

Figure 3.4: Converge curve for Gibbs$^+$ algorithm.

HAP selects $K = 5$ devices at a time.

Figure 3.5 shows the average sum-rate of CE decreases with higher α values. When $\alpha = 0.1$, the sum-rate of sample size 500, 1000, and 1500 is 12.7582 bps/Hz, 12.7539 bps/Hz, and 12.7540 bps/Hz, respectively. When α increases to 1.0 the sum-rate respectively decreases to 12.3909 bps/Hz, 12.6008 bps/Hz, and 12.6815 bps/Hz. Recall that CE generates schedules, i.e., samples, according to a multivariate Bernoulli distribution. A large α value causes some elements to have a probability of one or zero prematurely. This causes CE to return a solution with a lower sum rate.

Figure 3.6 shows that the learning duration of CE decreases with higher α values. We find that when $\alpha = 0.1$, the learning duration of sample size 500, 1000, and 1500 is 497, 1056 and 1681 seconds, respectively. Then, when α increases to 1.0, the learning duration of sample size 500, 1000, and 1500 decreases to 14, 48 and 86 seconds respectively. This is because a small α value, e.g., 0.1, means that the Probability Mass Function (PMF) changes slowly in each iteration. Thus, CE will have a slow convergence rate. To this end, to balance the trade-off between learning duration and sum-rate performance, smoothing parameter α is set to a value greater than equal to 0.6 when the sample size is larger than 1500.

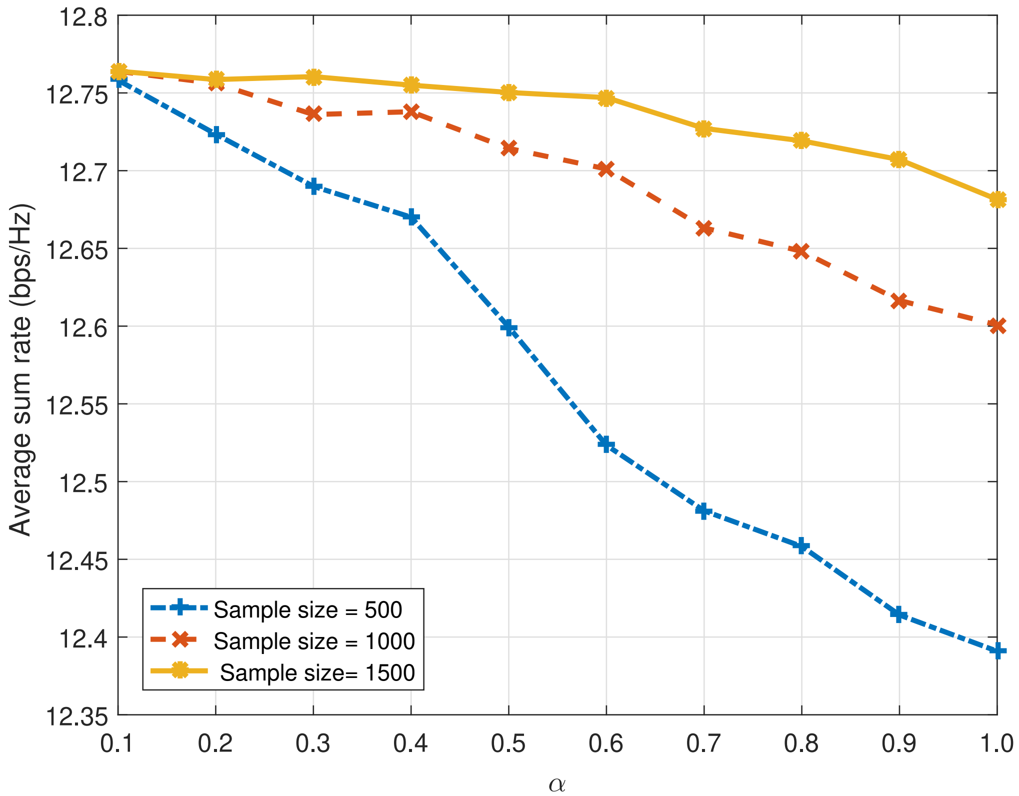

Figure 3.5: Sum-rate of CE versus different α values.

3.5.3 Sample Size

To study sample sizes, this section considers $\{500, 1000, 1500, \ldots, 5000\}$. The battery leakage rate is set to 20%. The transmit power of the HAP is 3 W. In each time slot, the HAP selects $K = 5$ devices to transmit.

Figure 3.7 shows that the average sum rate of CE increases with a higher number of samples. Referring to Figure 3.7, when the sample size is 500, the sum-rate when the smoothing parameter has a value of 0.6, 0.7, 0.8, and 0.9 is 12.519 bps/Hz,

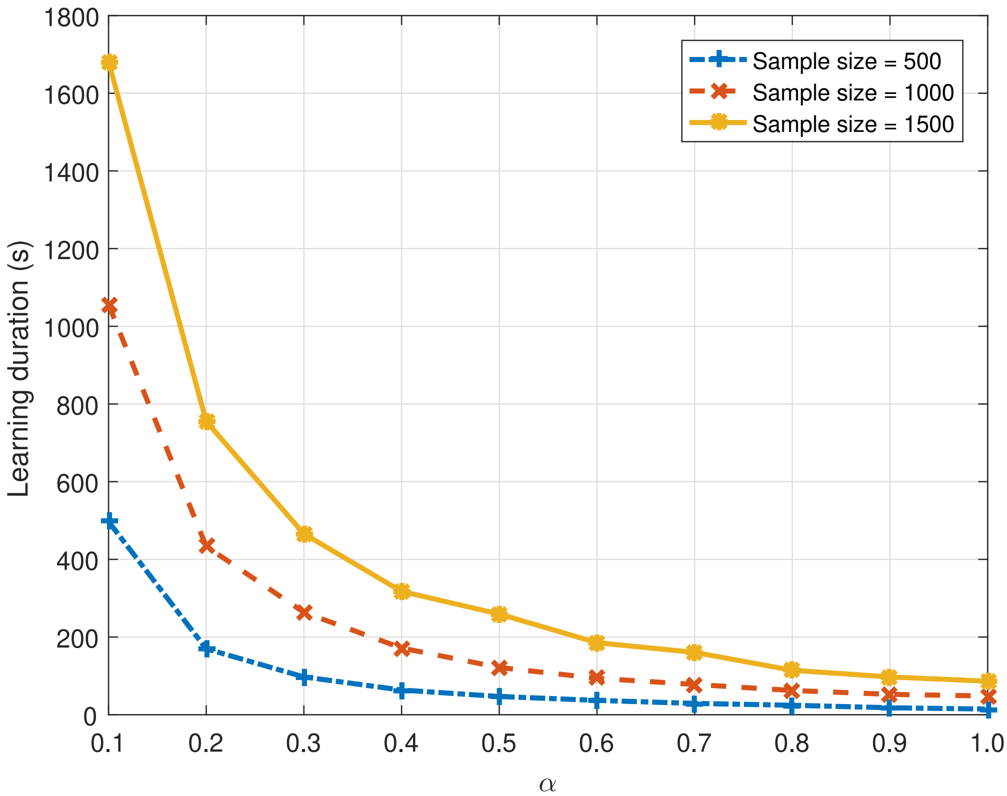

Figure 3.6: Learning duration of CE versus different α values.

12.502 bps/Hz, 12.436 bps/Hz and 12.372 bps/Hz respectively. When the sample size increases to 5000, the sum-rate increases respectively to 12.765 bps/Hz, 12.761 bps/Hz, 12.757 bps/Hz, and 12.751 bps/Hz. This is because a higher sample size, i.e., 5000, covers more potential schedules, meaning there is a higher probability of finding the best solution. This is important because CE updates the PMF according to elite samples. As expected, when $\alpha = 0.6$, CE produces the highest sum rate. Recall that a lower α value, i.e., 0.6, increases the probability that CE converges to a global optimal solution as it allows more exploration of the solution space. From Figure 3.8 we observe that when the sample size is 500, the learning duration of smoothing parameter $\alpha = 0.6$, 0.7, 0.8 and 0.9 is 41, 34, 29 and 22 seconds, respectively. When the sample size increases to 5000 the learning duration of smoothing parameter $\alpha = 0.6$, 0.7, 0.8, and 0.9 respectively increases to 998, 914, 735, and 672 seconds. This is because with more samples, i.e., 5000, CE needs a longer time to collect the reward of samples in each iteration. This slows the learning rate of CE. Thus CE uses a longer time to converge when $\alpha = 0.6$, which is consistent with the result of Section 3.5.2. Referring to Figure 3.7 and Figure 3.8, we find that, when the sample size is larger than 3000, the sum-rate of $\alpha = 0.7$ is close to $\alpha = 0.6$ while yields a shorter learning duration. Therefore, a conclusion is that a sample size of 3000 and a smoothing parameter of 0.7 is the best combination to yield a high sum rate.

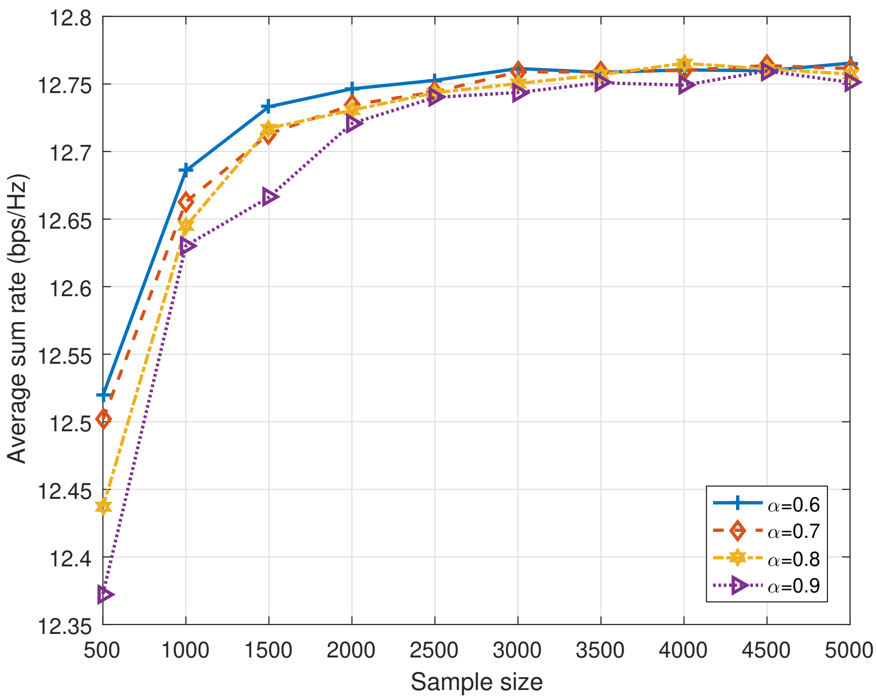

Figure 3.7: Sum-rate of CE versus different number of samples.

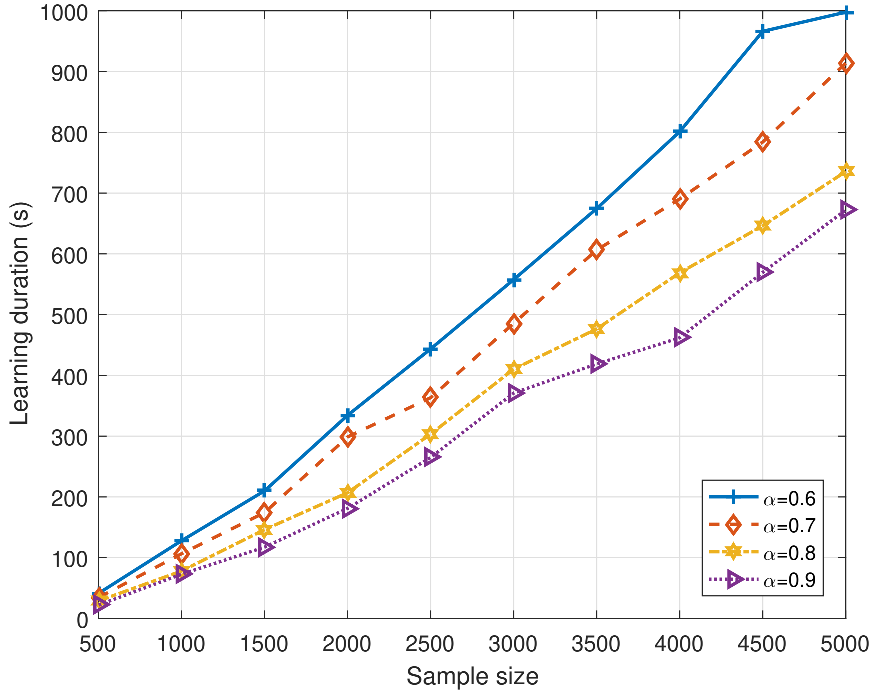

Figure 3.8: Learning duration of CE versus the number of samples.

3.5.4 Charging Power

This section studies the following HAP transmit power values (in Watts): $P^t \in \{1, 2, 3, 4, 5\}$. The battery leakage rate is 20%. In each time slot, the HAP will select $K = 5$ devices to transmit.

Figure 3.9 shows the sum rate of PIS, RR, RP, CE, Gibbs$^+$, and OGS when the HAP uses different transmit power or P^t values. When P^t changes from 1 W to 5 W, the sum rate of CE, Gibbs$^+$, PIS, RR, RP and OGS increases by about 27.15%, 38.22%, 25.72%, 59.74%, 65.25%, and 67.69%, respectively. This is because a higher transmit power P^t results in devices having a higher energy level on average, which helps improve their sum rate. Referring to Figure 3.9, CE and Gibbs$^+$ always outperform RR, RP, and OGS. This is because CE does not select devices located far away from the HAP until they have accumulated sufficient energy to produce a higher throughput than those devices nearer to the HAP. Gibbs$^+$ reduces the selection frequency of devices located far from the HAP and gives those devices located nearer to the HAP more chances to transmit. The RR rule only selects devices according to a fixed order, i.e., device D_1 to D_5 in one time slot, and selects device D_6 to D_{10} in the next time slot before repeating the sequence. The RP rule simply picks five devices randomly in each time slot. An interesting observation is that OGS's performance is close to RP. This is because in each time slot OGS randomly removes and selects one device. This means OGS produces a new schedule randomly. When the transmit power is $P^t = 1$ W, the sum rate attained by CE is 9.7% higher Gibbs$^+$, 35.36% higher than RR, and 42.56% higher than RP. This is because when P^t is 1 W, the received power or energy at devices will be low, especially those devices located far from the HAP, which only received less than 0.05 mW. At such received power, the energy conversion efficiency is less than 1%. Moreover, Gibbs$^+$, RR, and RP select these far away devices more frequently than CE, which explains their low sum rate. The charging efficiency of the RF harvester that is used by devices is higher with higher received signal power [127]. When the transmission power is $P^t = 2$ W, far away devices have a correspondingly higher received power, which leads to a better charging efficiency, i.e., 20%, meaning that their energy level is also higher. Consequently, when the transmission power P^t increases from 1 W to 2 W, the sum rate of Gibbs$^+$, RR, and RP shows a more significant increase as compared to CE. The sum rate of Gibbs$^+$, and the RR and RP rules increases by 24.8%, 38.7%, and 42.96% respectively, while CE only increases by 14.9%. From Figure 3.9, we find that the sum rate of CE is 99% that of PIS. When P^t is larger than 1W, the sum rate of Gibbs$^+$ reaches 99% of CE. Gibbs$^+$ uses less time than CE to find the schedule that produces a high sum rate. The running duration of Gibbs$^+$ is around 136 second which is 60% faster than the running time of CE.

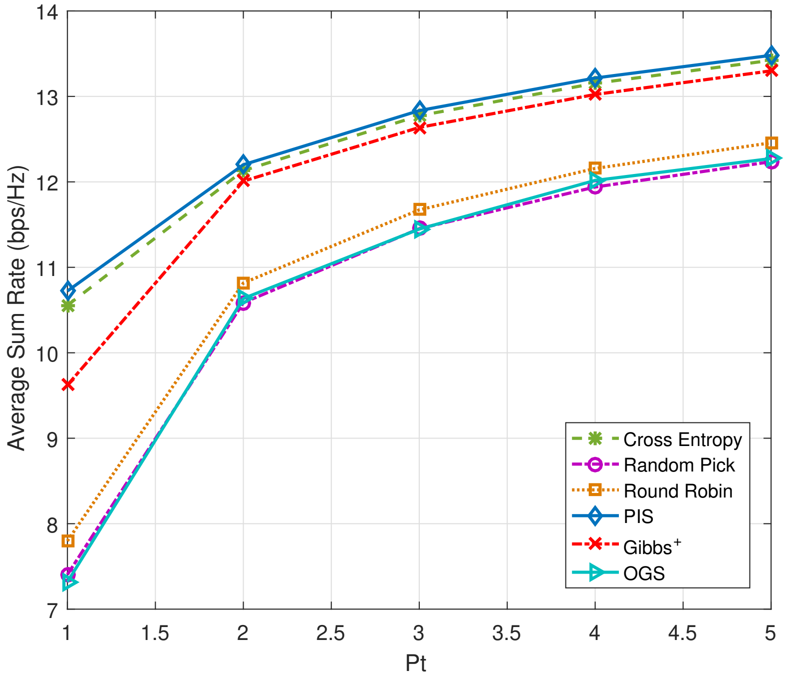

Figure 3.9: Sum-rate of PIS, RR, RP, OGS, CE, and Gibbs$^+$ versus HAP transmit power.

3.5.5 Number of Selected Devices

To study how the number of devices impact the sum-rate at the HAP, the simulation study in this section considers one to nine devices. The transmission power of the HAP is 3 W. The battery leakage rate is 20%.

Figure 3.10 shows the sum rate of PIS, RR, RP, CE, Gibbs$^+$, and OGS when the HAP selects a different number of devices, i.e., K, to transmit. From Figure 3.10, we find that the sum rate of CE and PIS decreases with higher K values. Specifically, the sum rate of these two strategies decreases by 13.1% and 13.2%, respectively. This is because as the HAP needs to select more devices in each time slot, it will have to select devices located further away. These devices tend to have less energy in each charging slot and thereby have a lower throughput than those nearer to the HAP. Referring to Figure 3.10, the sum rate of RR and RP increased by 9.8% and 13.4%. The reason is that when using the RR and RP rule, devices will have more opportunities to transmit, especially devices near the HAP. When the selected

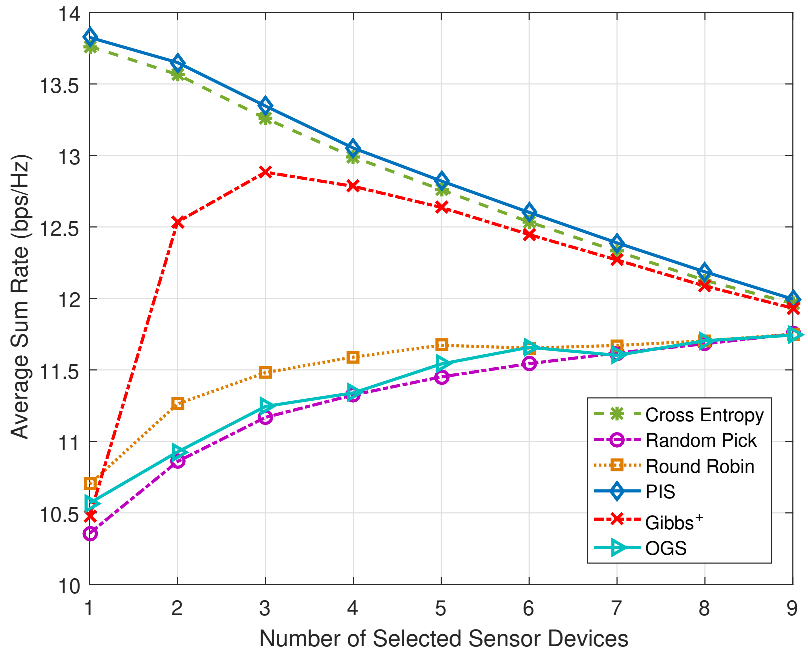

Figure 3.10: Sum-rate of PIS, RR, RP, OGS, CE, and Gibbs$^+$ versus the number of selected devices.

device number increases from one to three, the sum rate of Gibbs$^+$ increases by about 23%. This is because when $K = 1$, Gibbs$^+$ randomly generates new schedules in each iteration. The device replacement strategy ensures that Gibbs$^+$ selects $K-1$ devices that have the highest throughput in each time slot. Consequently, Gibbs$^+$ produces a higher throughput when $K > 1$. The sum rate of Gibbs$^+$ decreases by 7.4% when K increases from three to nine. Recall that as the HAP needs to select more devices in each time slot, it has to select devices that are located further away from itself. This causes a decrease in sum rate when the HAP uses Gibbs$^+$. The results show that the sum rate of RR, RP, PIS, OSG, CE, and Gibbs$^+$ becomes closer to each other with higher K values. This is because by selecting more devices, PIS, RR, RP, OGS, CE, and Gibbs$^+$ are more likely to select the same devices to transmit in each time slot, which produces a similar sum rate.

3.5.6 Battery Leakage Rates

Different battery leakage rate ϱ also influences the sum-rate. To this end, this section investigates how ϱ influences the selection strategy of the CE method. It reports on the following ϱ values: $\{0\%, 20\% \ldots, 100\%\}$. The HAP will select five devices to transmit in each time slot.

Figure 3.11 shows the sum-rate of PIS, RR, RP, OGS, CE, and Gibbs^{+} when ϱ changes from 0% to 100%. As expected, the sum-rate of the RR, RP, and OGS decreases with higher battery leakage rates. The sum-rate of RR, RP, and OGS decreases by 6.9%, 5.1%, and 4.3% respectively. This is because the increased battery leakage rate results in selected devices having a low transmit power. The results show that the sum-rate of CE, Gibbs^{+}, and PIS only decreases by 1.9%, 1.8%, and 2.0% when ϱ increases from 0% to 100%. This is because in each time slot CE, Gibbs^{+}, and PIS select the nearest four devices to transmit. Recall that this chapter assumes the selected devices do not lose energy and will use all stored energy to transmit. That is when using CE, Gibbs^{+}, and PIS, in each time slot the transmission power of the nearest devices does not decrease with higher ϱ values. Consequently, the sum-rate of CE, Gibbs^{+}, and PIS does not decrease significantly.

Figure 3.12 shows the selection strategy of CE when batteries have a different leakage rate ϱ. From Figure 3.12, CE always select device D_1 to D_4 to transmit. This is because they are located near the HAP. Thus, these devices have higher throughput than D_5 to D_{10} in each time slot. When ϱ increases from 0% to 100%, the HAP increases the selection time of D_5 from eight to eighteen times. At the same time, the HAP decreases the selection frequency of D_6,...,D_{10}. This is because for $\varrho = 0\%$ devices that have not transmitted any data will have the opportunity to accumulate energy. This means that within two or three time slots, devices located further away from the HAP will accumulate sufficient energy to produce a higher throughput than those nearer to the HAP; for example, device D_5. The HAP thus selects D_6, ..., D_{10} more frequently when $\varrho = 0\%$. In each time slot, the amount of

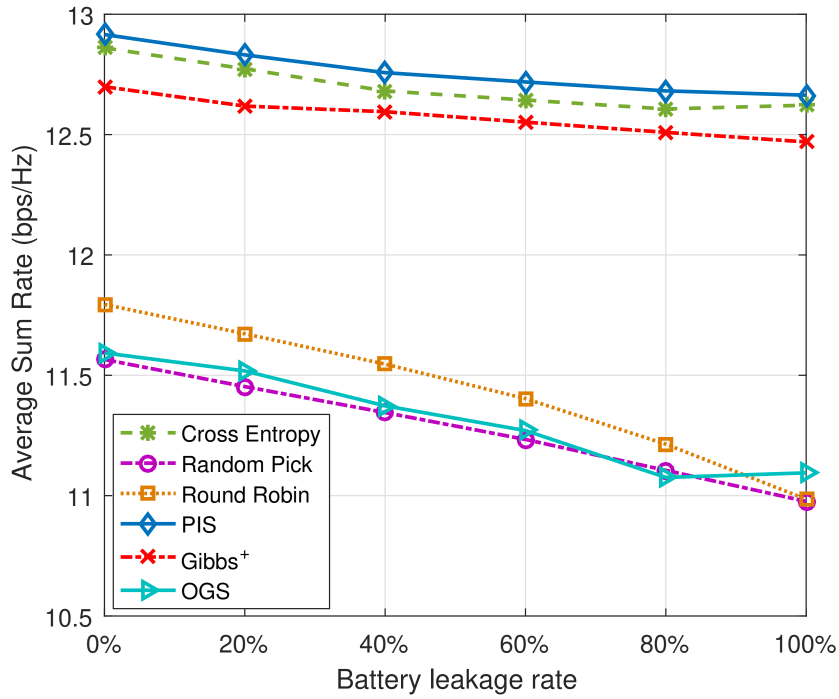

Figure 3.11: Sum-rate of PIS, RR, RP, OGS, CE, and Gibbs$^+$ versus the battery leakage rate of devices.

energy accumulated by devices located far from the HAP will decrease with higher leakage rate ϱ. This means that when the battery leakage rate is larger than 0%, devices located further away from the HAP need a longer time to accumulate energy in order to generate higher throughput than device D_5. The HAP, thus, reduces how often it selects these far away devices.

3.5.7 Impact of Channel Variation

Channel variation also influences CE and Gibbs$^+$. Recall that $\mathcal{X}$ relates to the severity of channel condition. To this end, a simulation is conducted for the following standard deviation μ values: $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$. The transmission power of the HAP is 3 W. The HAP selects five devices in each time slot.

From Figure 3.13, the learning duration of CE increases from 649 to 4571 seconds when μ changes from 0.5 to 3.0. This is because a higher μ value means that the channel gain will vary more drastically between device D_i and the HAP. Recall that

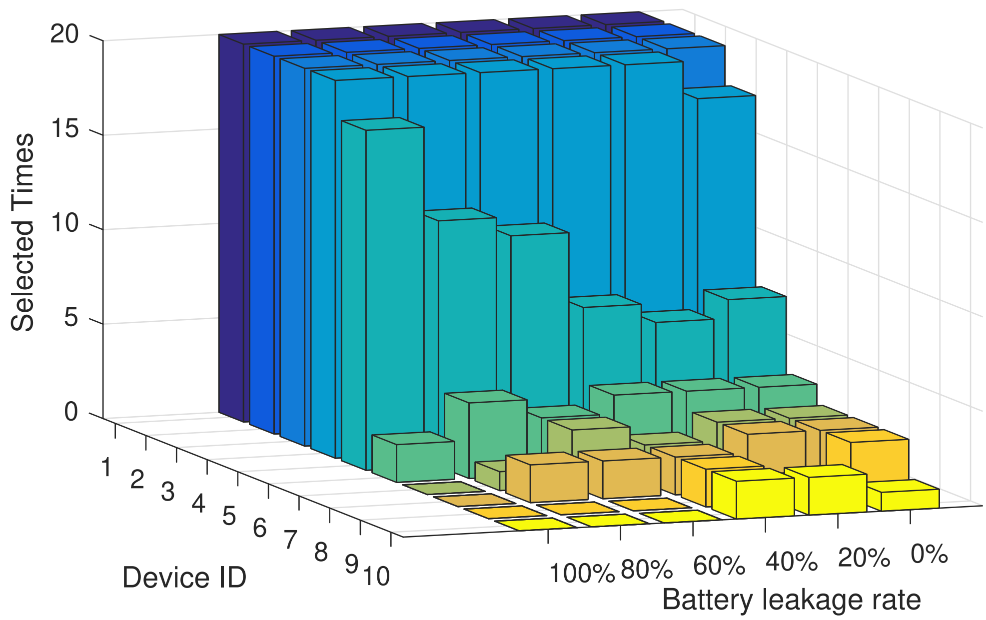

Figure 3.12: Selection strategy of CE versus the battery leakage rate of devices.

the reward of a sample is a function of the channel state. Consequently, devices that make it to the elite samples vary more significantly in each iteration when the channel gain variance is high. This slows the convergence of CE. Referring to Figure 3.13, the running duration of Gibbs$^+$ remains around 133 seconds when μ changes from 0.5 to 3.0. This is because the running duration of Gibbs$^+$ is influenced by: (i) the number of iterations, and (ii) the number of potential schedules in each time slot. Hence, μ impacts on neither (i) nor (ii). Figure 3.14 shows the sum rate of RIP, RR, RP, CE, Gibbs$^+$, and OGS. Referring to Figure 3.14, when μ increases from 0.5 to 3.0, the sum rate of CE remains at around 12.75 bps/Hz. This means the slight increase in μ value does not decrease the sum rate of CE.

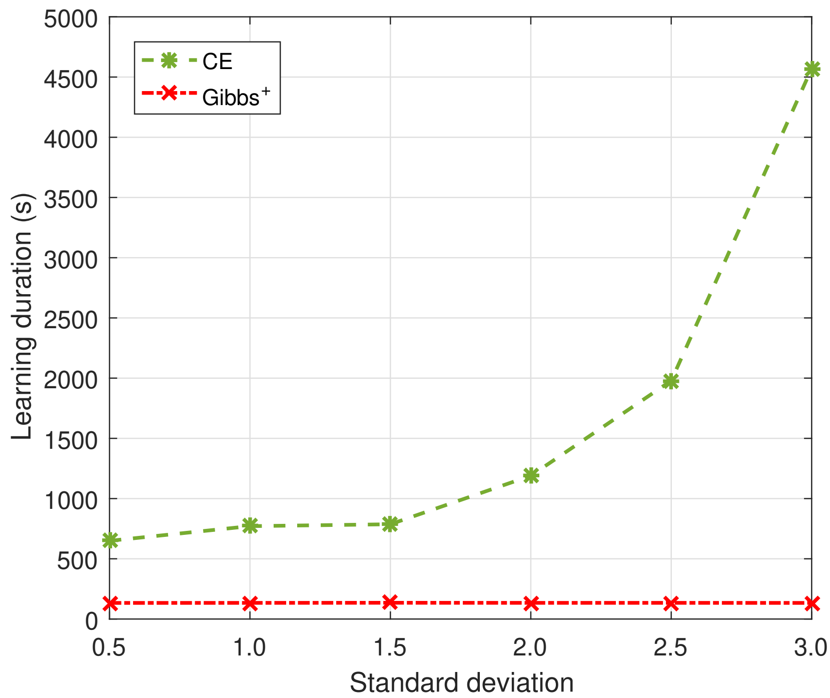

Figure 3.13: Learning duration of CE versus standard deviation μ.

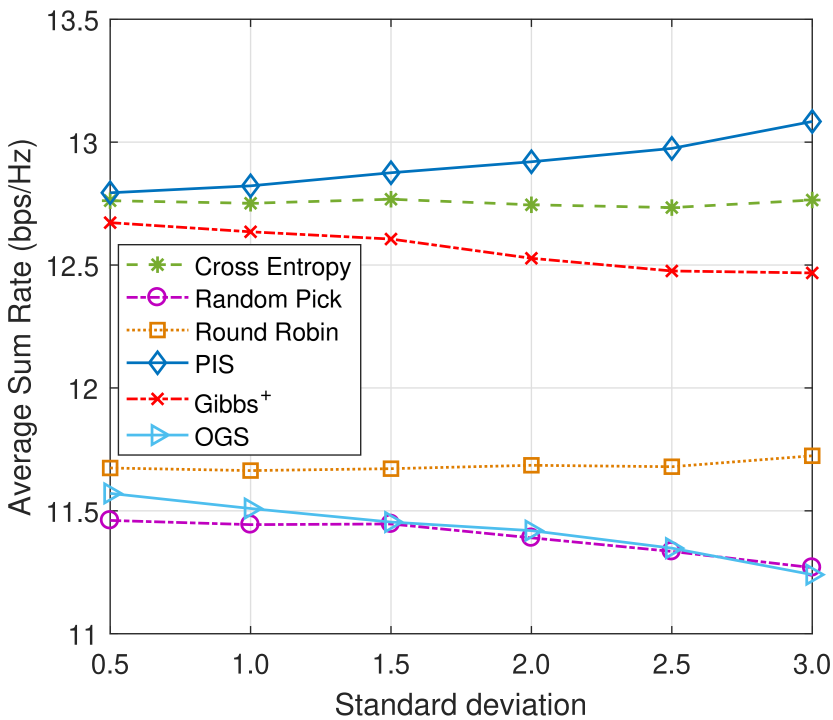

Figure 3.14: Sum-rate of PIS, RR, RP, OGS, CE, and Gibbs$^+$ versus standard deviation μ.

3.6 Conclusion

This chapter has considered a device selection problem in an RF charging network where multiple devices are powered by a HAP. The HAP aims to select the best set of devices to transmit in each time slot in order to maximize the sum rate over a planning horizon. Critically, this chapter considers a challenging aspect whereby the HAP does not have perfect CSI nor battery state information of devices. This is significant in a large-scale RF charging network as it becomes impractical to poll every device for the said information. To address the problem, this chapter proposes a CE-based algorithm and a Gibbs$^+$ algorithm. The simulation results show that the performance of CE and Gibbs$^+$ algorithm respectively achieve 99% and 98% of average sum-rate attained by PIS. The proposed algorithms also achieve a higher sum rate than RR, RP, and OGS. In addition, the Gibbs$^+$ algorithm is around 60% faster than the CE algorithm.

This chapter only focuses on sum-rate maximization in an RF-charging network. It ignores information freshness of samples or data from devices, which quantifies when a sample was taken by a device. Hence, a key topic in the next chapter is to ensure devices are able to update a HAP frequently to minimize AoI.

Chapter 4

Age of Information Minimization in RF-Charging WSNs

The previous chapter considers a sum-rate maximization problem in an RF-charging network. However, it ignores information freshness. Hence, this chapter addresses a challenging problem: transmit device selection, where the HAP does not have the uplink CSI of devices. The main research problem is to determine the set of transmitting devices in each frame so as to minimize the average AoI at the HAP.

To illustrate the research problem, consider the WPCN shown in Figure 4.1. For simplicity, this example assumes the WPCN consists of two devices and a HAP, i.e., $N = 2$, and only consider two frames. In each frame, the HAP first broadcasts energy and then selects one device to generate and transmit its sample. The aim is to find the best transmit device in each frame so as to minimize the average AoI of each frame. Consider Figure 4.1. The AoI of devices increases by one at the start of each frame. If a device transmits its sample successfully, the HAP resets the device's AoI to zero at the end of a frame. Assume D_1 only needs one frame to accumulate energy to transmit its sample to the HAP, while D_2 needs two frames. Also, assume D_2 has a poor channel in the first frame. Otherwise, both devices have a good channel. Now consider the following cases: (i) *Case-1:* The HAP selects D_2 in the first frame and selects D_1 in the next frame. In this case, D_2 fails to transmit as it has insufficient energy and a bad channel. Its AoI in frame-1 and frame-2 is respectively one and two. As for D_1, it is one and zero. The average AoI per frame is calculated as $\frac{1+2+1+0}{2\cdot 2} = 1$, (ii) *Case-2:* the HAP selects D_1 followed by D_2. In this case, both devices will have sufficient energy to transmit when they are selected by the HAP. As shown in Figure 4.1, the AoI of D_1 is respectively zero and one over two frames. As for D_2, it is one and zero. Hence, the average AoI per frame is $\frac{0+1+1+0}{2\cdot 2} = 0.5$.

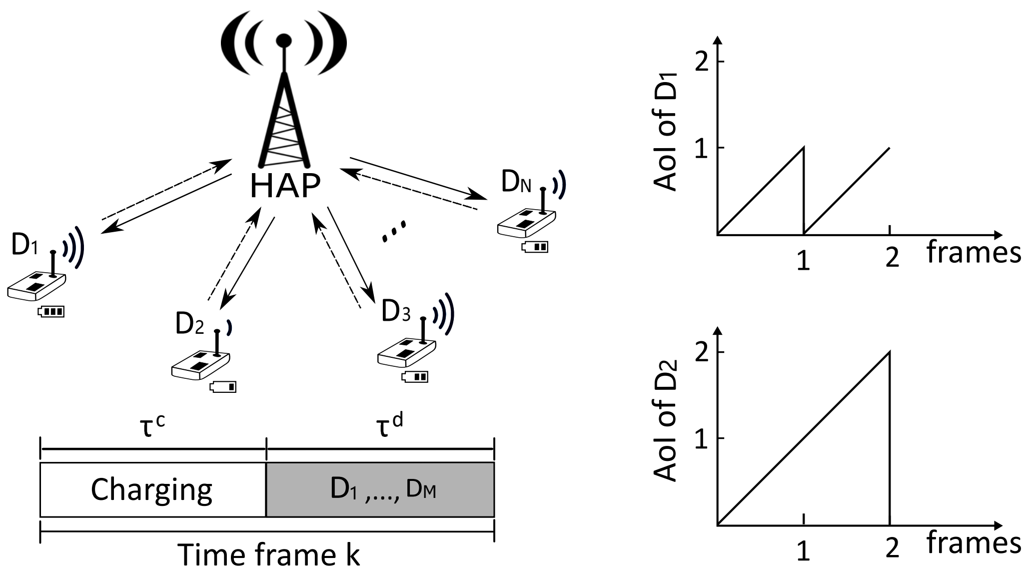

Figure 4.1: An RF-charging network, a frame and AoI evolution in *Case-2*.

From the above toy example, the key problem at the hand is that determine the devices that have sufficient energy to generate a sample and then transmit successfully in order to minimize the average age of information. The key challenge is that the HAP has the select the best devices without knowing their channel state information and energy level.

The rest of this chapter is structured as follows. Section 4.1 outlines the system model and problem. Then Section 4.2 outlines a novel Markov model, which is used to gain insights into the problem. The proposed distributed solution is presented in Section 4.3 followed by the evaluation in Section 4.4. Section 4.5 concludes this chapter.

4.1 System Model and Problem

Let $\mathcal{N} = \{D_1, D_2, \ldots, D_N\}$ be a set of devices. Time is divided into K frames. Each frame has two phases: (a) energy harvesting, and (b) data transfer. Their respective duration is τ^c and τ^d. In frame k, the HAP first broadcasts energy over its omni-directional antenna with power P (in Watt). In each frame k, the HAP selects $M < N$ out of N devices, where it assigns each device an orthogonal channel. Note that, in each frame, there are $\binom{N}{M}$ different device selection choices. Let $x_i^k \in \{0, 1\}$ denote whether the HAP selects device D_i in frame k; namely, it has $x_i^k = 1$ ($x_i^k = 0$) if the HAP selects (does not select) device D_i in frame k. Let $s^k = \{x_1^k, x_2^k, \ldots, x_N^k\}$ be the selection status of N devices. Define a *schedule* as $s_z = \{s^k \mid k = 1, \ldots, K\}$. The collection of schedules is denoted as $\mathcal{S}$.

The uplink channel gain between a device D_i and the HAP in frame k is denoted as g_{i0}^k, whereas for downlink it is g_{0i}^k. The channel gain is as per the Log-distance path loss, which is calculated as per Eq. (3.1).

4.1.1 Sampling and Buffer Model

A selected device consumes e_s amount of energy to generate a sample. It does not generate a new sample if its energy level is lower than e_s. Each device has a data storage capacity that only stores one latest sample [134]. At the end of a frame, device D_i removes a successfully transmitted sample. Otherwise, it retains the sample. Let $B_i^k \in \{0, 1\}$ and $\hat{B}_i^k \in \{0, 1\}$ denote the buffer state of device D_i at the beginning and conclusion of the data transfer phase in frame k, respectively. Specifically, $B_i^k = 1$ means a device has a sample, while $B_i^k = 0$ indicates its buffer is empty.

4.1.2 Energy Model

Devices have an RF energy harvester [135] with a non-linear energy conversion efficiency that is a function of the received power. A device stores any unused energy for future use. The amount of energy received by device D_i in frame k is

$$e_i^k = \frac{\left[\psi_i^k - H_{max}^i \Omega_i\right] \tau^c}{1 - \Omega_i}, \tag{4.1}$$

$$\Omega_i = \frac{1}{1 + \exp(a_i b_i)}, \tag{4.2}$$

$$\psi_i^k = \frac{H_{max}^i}{1 + \exp(-a_j(p^k g_{0i}^k - b_j))}, \tag{4.3}$$

where H_{max}^i is a constant that corresponds to the maximum harvested power at device D_i. Parameter a_i and b_i are constants that relate to the RF-energy harvester hardware of [135]. In addition, Ω_i is a constant that is specific to a given circuit specification and ψ_i^k is the logistic function with respect to the received power of devices D_i in frame k. Each device is equipped with a battery with a capacity of B_{max}, meaning any energy that arrives once it has $E_i^k = B_{max}$ is lost. Formally, the energy storage of device D_i at the end of frame k evolves as

$$E_i^k = \begin{cases} 0, & x_i^k = 1 \wedge B_i^k = 1, \\ \min(B_{max}, E_i^{k-1} + e_i^k), & \text{Otherwise.} \end{cases} \tag{4.4}$$

4.1.3 Transmission Model and AoI

A selected device, say D_i, will use all its energy to transmit a sample. Its transmit power in frame k is

$$p_i^k = \begin{cases} 0, & E_i^{k-1} + e_i^k < e_s \wedge \hat{B}_i^{k-1} = 0, \\ \frac{E_i^{k-1} + e_i^k}{\tau^d}, & E_i^{k-1} + e_i^k < e_s \wedge \hat{B}_i^{k-1} = 1, \\ \frac{E_i^{k-1} + e_i^k - e_s}{\tau^d}, & E_i^{k-1} + e_i^k > e_s. \end{cases} \tag{4.5}$$

A device's transmission fails if the Signal-to-Noise Ratio (SNR) at the HAP falls below the threshold ζ. Let $I_i^k \in [0,1]$ denote whether device D_i transmits data successfully in time frame k, which is given as

$$I_i^k = \begin{cases} 1, & \frac{p_i^k g_{i0}^k}{N_0 W} \geq \zeta, \\ 0, & \frac{p_i^k g_{i0}^k}{N_0 W} < \zeta, \end{cases} \tag{4.6}$$

where N_0 and W denotes noise spectral density and channel bandwidth, respectively.

Recall that, the AoI of a device is defined as the number of frames that have elapsed since the sample stored at the HAP was generated at the device. Let $\hat{k}_i$ be the frame index in which device D_i obtains its sample, and $A_i^{s_z}$ denotes the AoI of device D_i when the HAP uses schedule s_z. At the beginning of a time frame, the AoI of device D_i evolves as

$$A_i^{s_z}(k+1) = \begin{cases} k - \hat{k}_i + 1, & x_i^k = 1 \wedge I_i^k = 1, \\ A_i^{s_z}(k) + 1, & \text{Otherwise.} \end{cases} \tag{4.7}$$

Note that the freshest samples are those from the current frame. i.e., frame k. Therefore, it always has $\hat{k}_i \leq k$. Let $\bar{A}(s_z)$ denote the average AoI over K frames when using schedule s_z. Mathematically, the average AoI per frame is

$$\bar{A}(s_z) = \frac{1}{NK} \sum_{k=1}^{K} \sum_{i=1}^{N} A_i^{s_z}(k). \tag{4.8}$$

4.1.4 The Problem

The aim is to minimize the average AoI over a given duration K. To do this, the HAP needs to determine a schedule s_z that selects which set of devices to sample and transmit in each frame. Formally, the problem is

$$\min_{s_z \in \mathcal{S}} \quad \mathbb{E}_{\varphi}\left[\bar{A}(s_z)\right], \tag{4.9}$$

where φ is the joint distribution over random channel gains to/from each sensor device in $\mathcal{N}$.

4.2 A Markov Model

This section first outlines a novel Markov chain to study how the average AoI is affected when the HAP randomly selects M out of N devices at a time given random channel gains and energy arrivals. To ensure mathematical tractability, the Markov chain model assumes (i) a sample is generated with zero energy cost, (ii) the energy arrival at devices follows a Bernoulli distribution, i.e., one unit of energy arrives with probability $\hat{p}$, (iii) the uplink channel to the HAP is as per the Gilbert–Elliot (GE) channel model. It has two states, where the channel is either 'Good' ($\hat{g}$) or 'Bad' ($\hat{b}$). There is a probability that governs whether the channel remains in a state or changes state. Let $P_{\hat{g}\hat{b}}$ and $P_{\hat{b}\hat{g}}$ denote the transition probability from $\hat{g}$ to $\hat{b}$ and from $\hat{b}$ to $\hat{g}$, respectively. The steady-state probability of state $\hat{g}$ and state $\hat{b}$ is given as $\hat{\pi}_{\hat{g}} = \frac{P_{\hat{b}\hat{g}}}{P_{\hat{b}\hat{g}}+P_{\hat{g}\hat{b}}}$ and $\hat{\pi}_{\hat{b}} = \frac{P_{\hat{g}\hat{b}}}{P_{\hat{b}\hat{g}}+P_{\hat{g}\hat{b}}}$, respectively, and (iv) each device has a battery capacity of one unit of energy. Each device has either zero or one unit of energy. The transition probability from zero to one state and from one to zero state is $\hat{p}$ and $\frac{M(1-\hat{p})}{N}$, respectively. Let $\hat{\pi}^0$ and $\hat{\pi}^1$ be the steady-state probability of zero and one state, respectively. It has $\hat{\pi}^0 = \frac{M(1-\hat{p})}{M(1-\hat{p})+N\hat{p}}$ and $\hat{\pi}^1 = \frac{N\hat{p}}{M(1-\hat{p})+N\hat{p}}$. As an aside, it is also possible to use the channel model in [136] with multiple states. In this case, it can group a set of channel states to be 'Good' or 'Bad', where the corresponding probability $\hat{\pi}_{\hat{g}}$ and $\hat{\pi}_{\hat{b}}$ is equal to the sum of the steady-state probability of states in group 'Good' and 'Bad', respectively.

Referring to Figure 4.2, the Markov chain model has Y states; each state is the AoI of a device. The probability that the AoI increases by one is denoted as λ. The value of λ corresponds to (i) the probability that a device is selected by the HAP, i.e., $\frac{M}{N}$, (ii) the probability that a device has one unit of energy, and (iii) that a device's channel is in the 'Good' state. Formally, it has $\lambda = 1 - \frac{M\hat{\pi}^1\hat{\pi}_{\hat{g}}}{N}$.

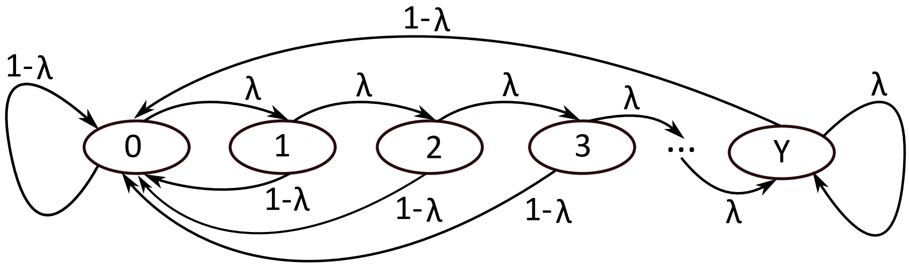

Figure 4.2: A Markov model depicting the AoI evolution of a device.

The steady-state probability for each state f can be shown to be

$$\hat{\pi}_f = \begin{cases} \frac{\lambda^Y}{(1-\lambda)\sum\limits_{y=0}^{Y-1}\lambda^y+\lambda^Y}, & f=Y, \\ \frac{\lambda^f(1-\lambda)}{(1-\lambda)\sum\limits_{y=0}^{Y-1}\lambda^y+\lambda^Y}, & \text{Otherwise.} \end{cases} \tag{4.10}$$

The expected AoI is then $\bar{A} = \hat{\pi}_1 + 2\hat{\pi}_2 + \cdots + Y\hat{\pi}_Y$. Using (4.10), we have

$$\begin{aligned} \bar{A} =& (1-\lambda)\sum_{v=1}^{Y-1}\frac{v\lambda^v}{(1-\lambda)\sum\limits_{y=0}^{Y-1}\lambda^y+\lambda^Y} \\ &+ \frac{Y\lambda^Y}{(1-\lambda)\sum\limits_{y=0}^{Y-1}\lambda^y+\lambda^Y}. \end{aligned} \tag{4.11}$$

Now this section studies a network with ten devices, i.e., $N = 10$. The number of the selected devices, i.e., M, varies from one to seven. The probability that an uplink channel is in the 'Good' and 'Bad' state is 0.5. The maximum AoI, i.e., Y, is set to 20. Referring to Figure 4.3, if the HAP selects more devices in each frame, the average AoI of devices when the energy arrival rate equals 0.5 to 0.9 decreases by around 70%, 75%, 78%, 80%, and 82%, respectively. This is because a higher energy arrival coupled with the HAP selecting more devices per frame leads to a lower average AoI. As expected, the lowest average AoI is attained when the energy arrival rate is high, namely 0.9. Specifically, this means the probability that devices have energy is high when they are selected, which increases the probability of successful transmissions.

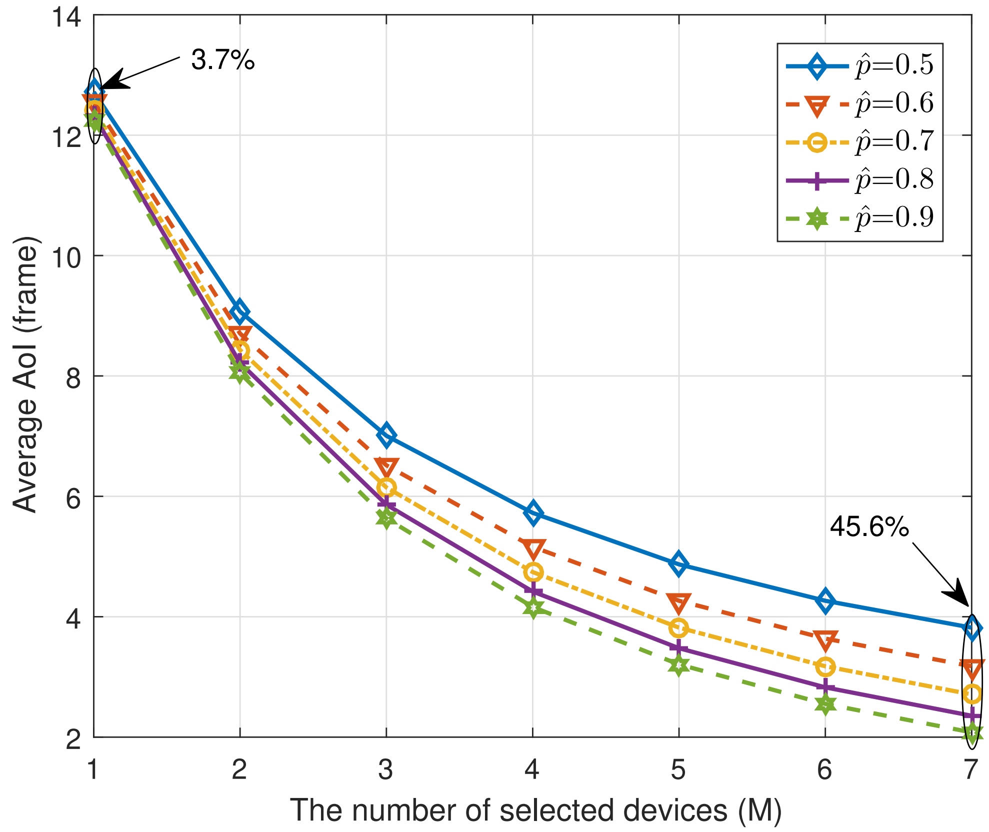

Figure 4.3: Expected AoI versus the number of selected devices.

Next, this section studies channel conditions, where the probability of 'Good' channel, i.e., $\hat{\pi}_{\hat{g}}$, varies from 0.1 to 0.7. It sets $M = 5$. Referring to Figure 4.4, as expected, higher energy arrivals coupled with better channel state results in a lower average AoI. This is because the probability of a successful transmission is proportional to the probability that a selected device has energy and its channel condition is 'Good'.

Note that the GE model is used for its mathematical tractability. Other models which contain multiple channel levels can also be used. Specifically, in this case, this section models the channel state according to the stationary probability of being in each level. For example, there are three states, i.e., 'good', 'mild', and 'bad'. The stationary probability of being in 'good', 'mild', and 'bad' state is π_g, π_m, and

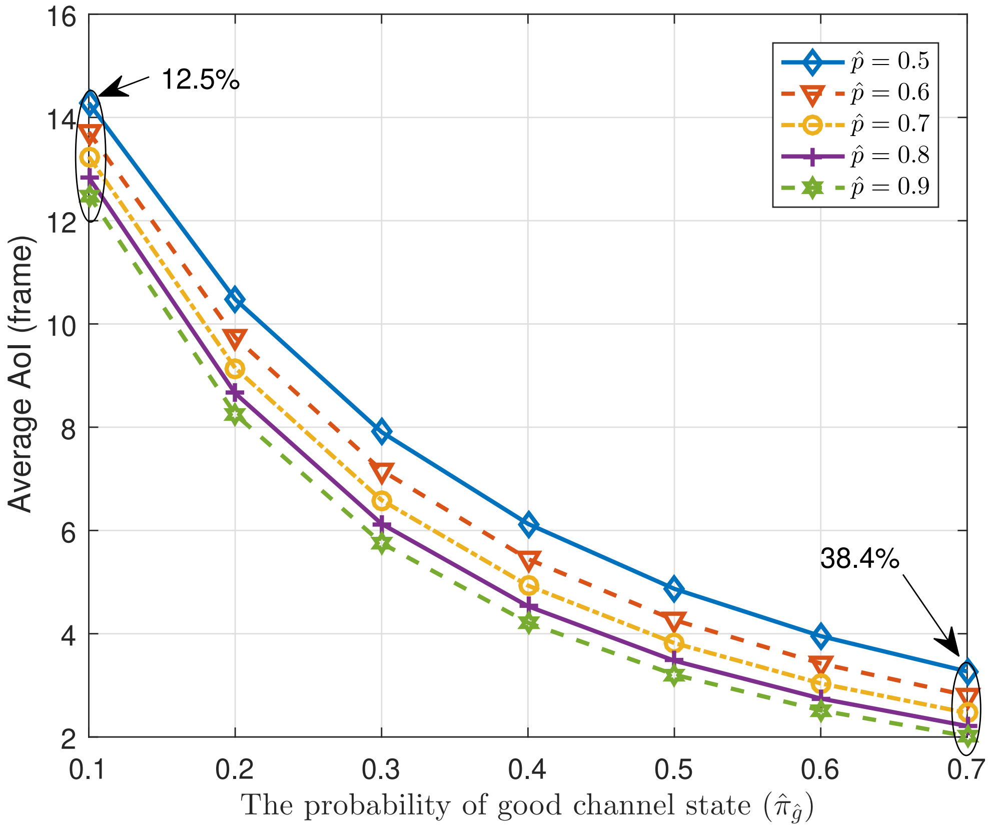

Figure 4.4: Expected AoI versus channel conditions.

π_b, respectively. Consider the example that shown in Figure 4.5. The stationary probability of being in state-1, state-2, and state-3 is π_1, π_2, and π_3, respectively. The said stationary probabilities then allow us to determine whether a channel is in a 'good' or 'bad' state given a level. In the above example, assume the channel state is considered 'good' if it is in state-2 or state-3. Then, the probability of being in 'Bad' and 'Good' channel state is π_1 and π_2+ π_3, respectively.

4.3 A Distributed Q-Learning Algorithm

DQL is a distributed protocol, whereby each device and the HAP act independently. In other words, the HAP is not required to collect channel state or battery level information from all devices nor devices need to communicate with one another. It has the following basic idea. At the beginning of a data slot, devices decide whether

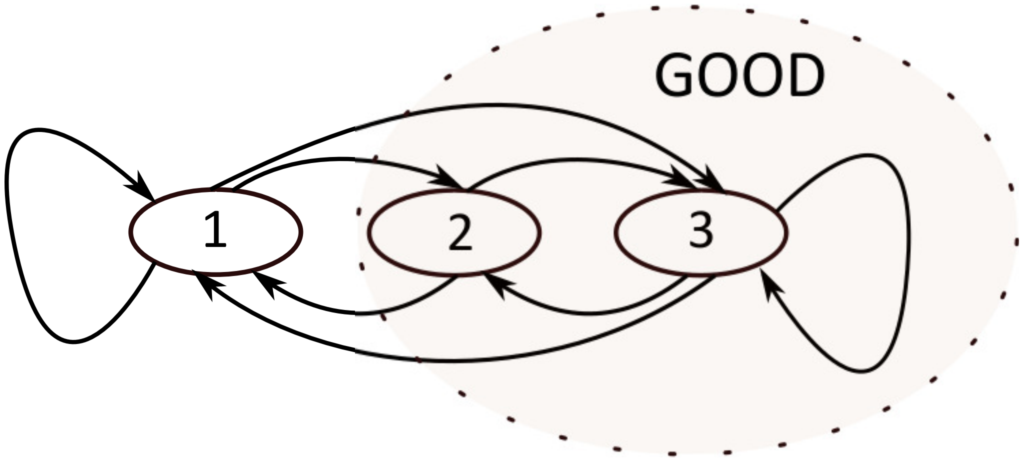

Figure 4.5: An example of a three-state channel model.

to send a request to the HAP based on their policy that is dependent on their energy level, buffer state, and uplink channel state. Upon receiving a request from devices, the HAP then selects devices that have the highest AoI to sample and transmit their packets. Advantageously, the HAP does not need to collect uplink channel state, battery state, and buffer state of devices. This helps devices save their energy.

Next, this section formulates a Markov Decision Process (MDP) and outlines the details of DQL.

4.3.1 A Decision Problem

Each device aims to minimize its AoI. Let $a_i^k \in \{0, 1\}$ denote the action of device D_i, where a value of one means that it will send a request to the HAP. Otherwise, it remains silent ($a_i^k = 0$). Let $A_i^k(a_i^k)$ denote the AoI of device D_i at the end of frame k when D_i selects action a_i^k, which is given as

$$A_i^k(a_i^k) = \begin{cases} k - \hat{k}_i, & a_i^k = 1 \wedge x_i^k = 1 \wedge I_i^k = 1, \\ A_i^{k-1}\left(a_i^{k-1}\right) + 1, & \text{Otherwise.} \end{cases} \tag{4.12}$$

Let R_i^k denote the difference between the AoI of device D_i at the end of frame

$k-1$ and that of frame k. Then, when device D_i takes action a_i^k in frame k, it has

$$R_i^k(a_i^k) = A_i^{k-1}(a_i^{k-1}) - A_i^k(a_i^k). \tag{4.13}$$

Formally, the problem of each device is to find its best action so as to maximize the following long-term reward:

$$R^\pi = \lim_{K\to\infty} \frac{1}{K}\mathbb{E}^\pi\left[\sum_{k=1}^{K} R_i^k(a_i^k)\right], \tag{4.14}$$

where π is a policy used to select a_i^k and the expectation is taken with respect to joint distribution of channel gains between device D_i and the HAP. The optimal policy π^* is

$$\pi^* = \arg\max_{a_i^k\in\{0,1\}} \frac{1}{K}\mathbb{E}^\pi\left[\sum_{k=1}^{K} R_i^k(a_i^k)\right]. \tag{4.15}$$

4.3.2 An MDP Model

Problem (4.14) can be modeled as an MDP $\{\hat{\mathcal{S}}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, where the corresponding state, action space, transition probability, and reward are defined as follows: (i) **State** $\hat{\mathcal{S}}$ consists of the energy level, uplink channel gain and the buffer state of a device. Specifically, letting $\widehat{s_i^k}$ denote the state of device i in frame k, it has $\widehat{s_i^k} = \{E_i^{k-1} + e_i^k, g_{i0}^k, B_i^k\}$, (ii) the **action** space is defined as $\mathcal{A}$. There are two actions, i.e., $a = 0$ and $a = 1$, (iii) the **transition probability** $\mathcal{P}$ is unknown as a model free approach is considered, and (iv) the **reward** $R_i \in \mathcal{R}_i$ of device D_i after taking an action is calculated as per Eq. (4.13).

4.3.3 Q-Learning and DQL Algorithm

Given that the MDP is model free, the chapter will apply the Q-learning algorithm [137] to find the optimal policy. Q-learning learns the optimal policy based on a so-called Q-table. Each Q-table is indexed by a state-action pair $(\widehat{s}^k, a^k)$ that has a corresponding Q-value $Q(\widehat{s}^k, a^k)$. This Q-value represents the expected discounted reward for taking an action in a state [137]. Q-learning aims to calculate $Q(\hat{s}^k, a^k)$ for each action of each state. The update rule of a Q-table is given as [137]:

$$\begin{aligned} Q(\hat{s}^k, a^k) =& (1-\alpha)Q(\hat{s}^k, a^k) + \alpha[r(\hat{s}^k, a^k) \\ &+ \gamma \max Q(\hat{s}^{k+1}, a^{k+1})], \end{aligned} \tag{4.16}$$

where $\alpha \in [0, 1]$ is the learning rate, $\gamma \in [0, 1]$ is the discount fact and $r(\hat{s}^k, a^k)$ is the reward of taking action a^k in state s^k, which is calculated as per Eq. (4.13).

Referring to Algorithm-4.1, a device D_i first initializes its Q-table arbitrarily. In each frame k, devices D_i selects an action as per the ϵ-greedy strategy; as shown from (3) to (8). Next, device D_i first calculates the reward after taking an action as per Eq. (4.13), as shown in line (9) and line (10). It then observes its state in the next frame and finds the maximum Q-value of its observed state. After that, device D_i updates its Q-table as per (4.16). Lastly, in each frame, the run-time complexity of DQL is $\mathcal{O}(|\mathcal{A}|)$. This is because a device needs to retrieve the value of each action for a given state in order to determine an action with the highest Q-value.

Algorithm 4.1: DQL algorithm for devices.

1 **Initialize:** Q-table
2 **for** *each frame* $k \in K$ **do**
3 Observe state $\hat{s}_i^k$ and generate random number $\hat{z} \in [0, 1]$;
4 **if** $\hat{z} \leq \epsilon$ **then**
5 Randomly select an action
6 **else**
7 Select the action with the highest Q-value
8 **end**
9 Calculate reward as per Eq. (4.13)
10 Update the Q-table as per Eq. (4.16)
11 **end**

4.4 Evaluation

All experiments in evaluation were conducted in Matlab. Experiments place $N = 30$ devices randomly between 1 to 6 meters from the HAP. They have a battery capacity of 1 J. The antenna gain of the HAP and devices is set to 1 dBi and 6.1 dBi respectively. Experiments set a_i and b_i of the energy harvester to 0.014 and 150, respectively, and it is capable of harvesting at a maximum of 24 mW [135]. The HAP operates at 915 MHz. The path loss exponent is 2.5. The noise power spectral density N_0 is set as -124 dBm/Hz. The channel bandwidth is 2 MHz. Experiments consider 100 frames. The charging duration and data transfer duration is respectively $\tau_c = \tau_d = 0.5$ s. The energy consumption of a sensor node to generate a sample is 0.26 mJ [138]. This section studies (i) convergence of DQL algorithm (ii) the number of devices, i.e., N, (iii) number of channels, i.e., M, and (iv) SNR threshold ζ. In case (i), (ii), and (iii), the SNR threshold is fixed and set as 3 dB. As for (i), (ii) and (iv), there are five channels.

DQL is compared against (i) **Random Pick (RP)**: in each frame, the HAP randomly selects at most M out of N devices, (ii) **Round Robin (RR)**: the HAP selects devices in turn, (iii) **AoI-Greedy (AG)**: the HAP selects devices that have the highest AoI to transmit with probability $1 - \hat{\epsilon}$; otherwise, it selects devices randomly. Note that the parameter $\hat{\epsilon} = 0.7$ is fixed, and (iv) **Perfect Information Selection (PIS)**: the HAP knows when devices generate their sample and has the perfect information of the battery state of devices and their uplink channel state. In each frame, the HAP selects M devices with the freshest sample and are capable of transmitting a packet. Hence, this rule yields the optimal result.

The first experiment studies the convergence of the proposed DQL algorithm. The simulation runs for 400 iterations. Referring to Figure 4.6, the proposed DQL algorithm converged. After converging, DQL outperforms RP, RR, and AG. This is because, after converging, DQL selects devices that have sufficient energy frequently to generate and update samples. This ensures the HAP receives more packets in each frame and attains a lower average AoI.

The second experiment varies the HAP transmit power from 3 W to 5 W. Referring to Figure 4.7, the average AoI of DQL decreases by around 14%. This is because devices accumulate energy quicker to generate and transmit their sample

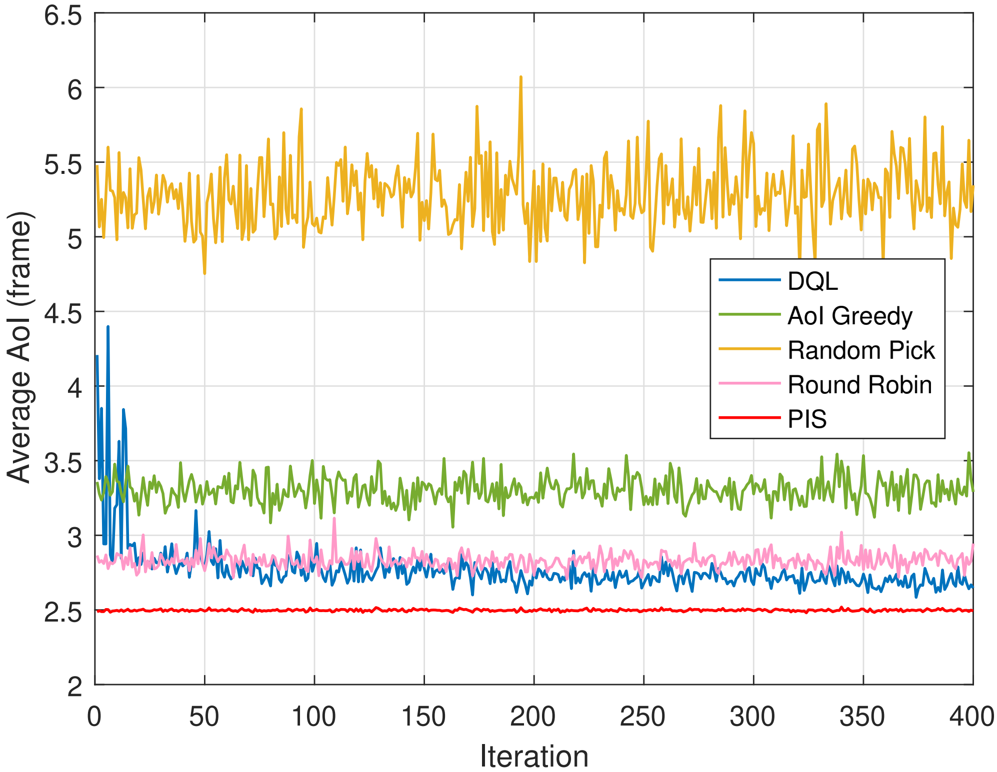

Figure 4.6: Convergence curve of DQL algorithm.

when the HAP has higher transmit power, i.e., 5 W. Their average AoI thus reduces.

Figure 4.8 shows the impact of device numbers. The average AoI of DQL is at most 11% lower than RR when there are fewer than 40 devices. When there are more than 40 devices, the average AoI of DQL is around 4.5% higher than RR. Note that with more devices, the action space of the HAP increases exponentially. Hence, there is an increased probability that DQL does not converge to the optimal Q-table, which affects the resulting average AoI.

The number of uplink channels has an impact on performance. This is because with more channels, the HAP receives more packets in each frame, which helps lower the average AoI. This is confirmed in Figure 4.9, where the average AoI of AG, RP, RR, DQL, and PIS decreases by around 35%, 45%, 33%, 39%, and 45% respectively.

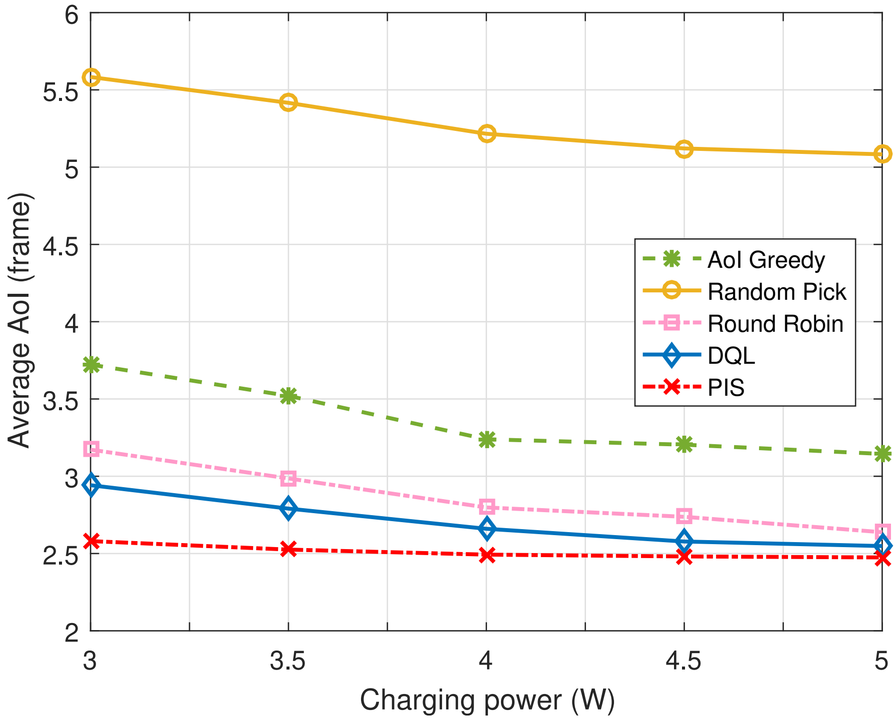

Figure 4.7: Average AoI versus HAP transmission power.

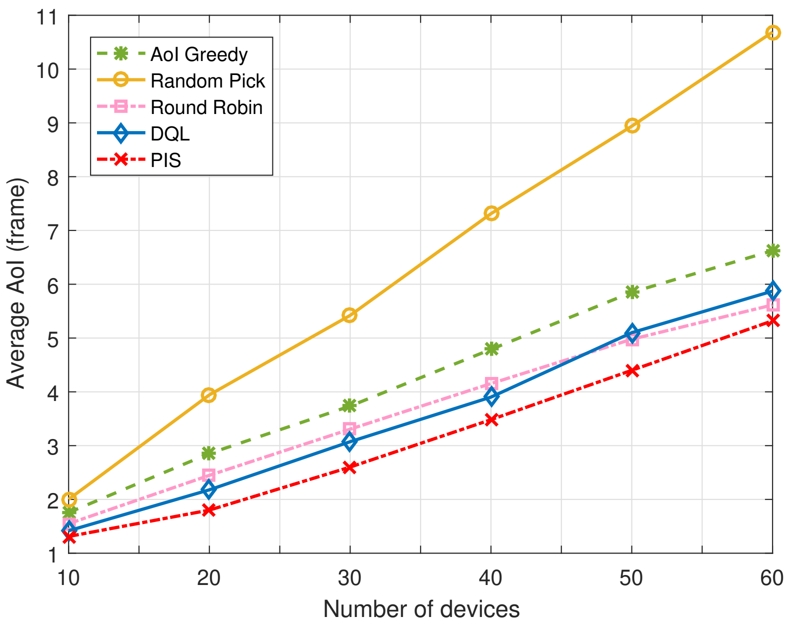

Figure 4.8: Average AoI versus the number of devices.

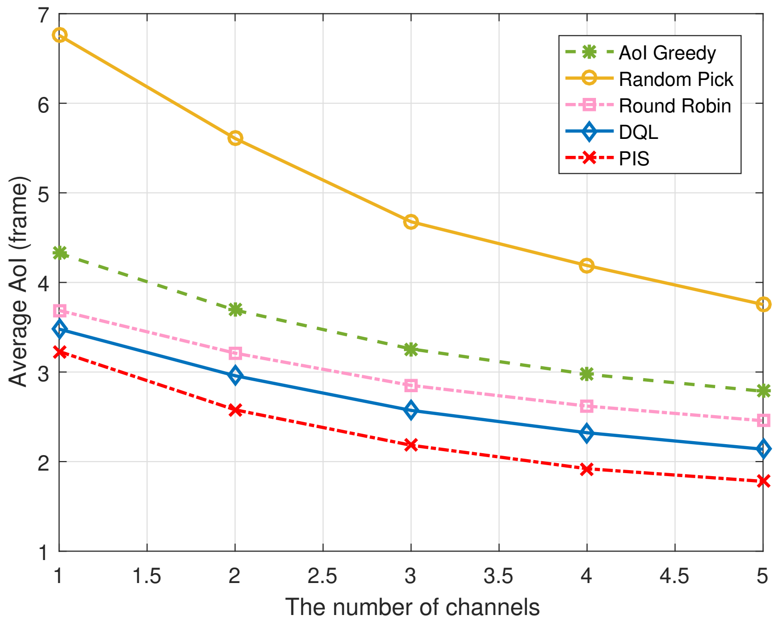

Figure 4.9: Average AoI versus the number of channels.

The last experiment studies varying SNR thresholds. Referring to Figure 4.10, when $\zeta = 3$ dB, the average AoI of DQL is 7%, 49%, and 19% lower than RR, RP, and AG, respectively. At 15 dB, the average AoI of DQL is 48%, 57%, and 61% lower than RR, RP, and AG, respectively. The gap between RR, AG, and DQL increases when the SNR threshold increases from 3 dB to 15 dB. This is because DQL does not select devices that are likely to experience a transmission failure, which helps devices conserve energy. This allows the HAP to receive updates from devices located further away more frequently. Therefore, DQL has a much lower average AoI than RR, RP, and AG.

4.5 Conclusion

This chapter has outlined a decentralized learning-based algorithm called DQL that allows a HAP to select devices without knowing their battery level and channel state. The results show that the average AoI decreases with higher HAP transmit power, number of channels, and higher SNR threshold. They also show that the average

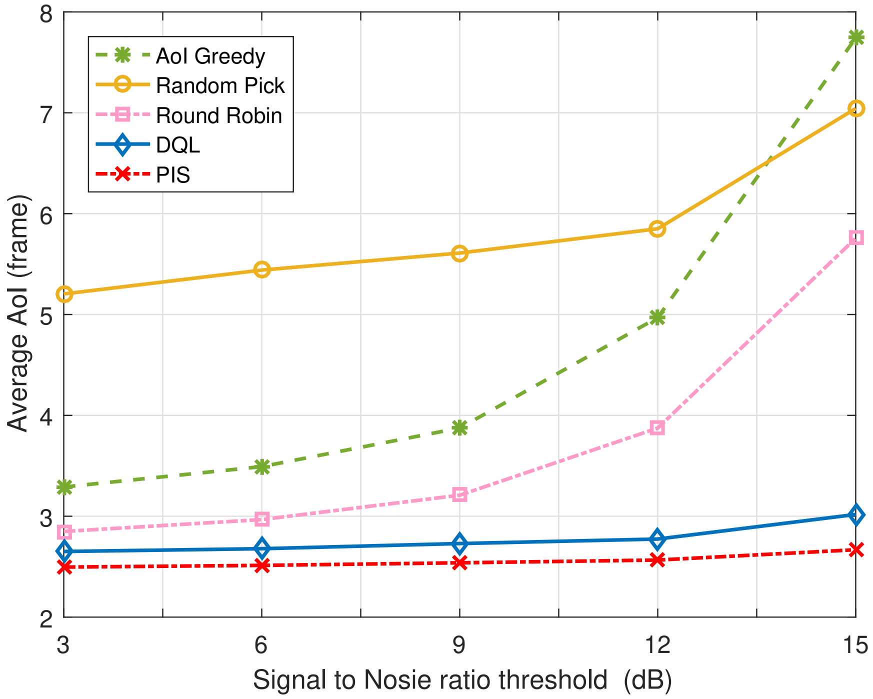

Figure 4.10: Average AoII versus SNR threshold.

AoI of DQL is 48%, 57%, and 61% lower than RR, RP, and AG, respectively.

A key assumption of DQL is that devices do not monitor multiple targets, especially when these targets have time-varying states. To this end, a research direction is to consider how devices monitor these targets to ensure they capture the state of targets. In this regard, it is necessary to consider the following cases: (i) one device monitors multiple targets, and (ii) multiple devices that cooperate to monitor one target. To this end, the next chapter addresses cases (i) and (ii) and outlines two approaches that minimize the AoII of targets.

Chapter 5

Minimizing Age of Incorrect Information in RF-Charging WSNs

Age of Incorrect Information (AoII) is a new performance metric that addresses the shortcomings of the AoI metric and error penalty functions that are used to quantify status updates [139]. To this end, this chapter contributes to the growing body of literature on AoII. It considers an RF-energy harvesting wireless sensor network that monitors multiple targets. The aim is to compute the optimal sensor activation schedule in order to minimize the average AoII of targets. Further, it presents two reinforcement learning-based methods to determine the said activation schedule.

To illustrate the research problem, consider the RF-charging network in Figure 5.1. There are three targets, two devices, and a HAP. Target T_1 and T_3 are being monitored by device D_1 and D_2, respectively. On the other hand, target T_2 is monitored by device D_1 and D_2. For simplicity, this example assumes that each target has two states, i.e., 'on' and 'off'. As shown in Figure 5.1, this example assumes that, in the first frame, all targets are in the 'on' state. In the second frame, target T_1 remains in the 'on' state, while target T_2 and T_3 transition to the 'off' state.

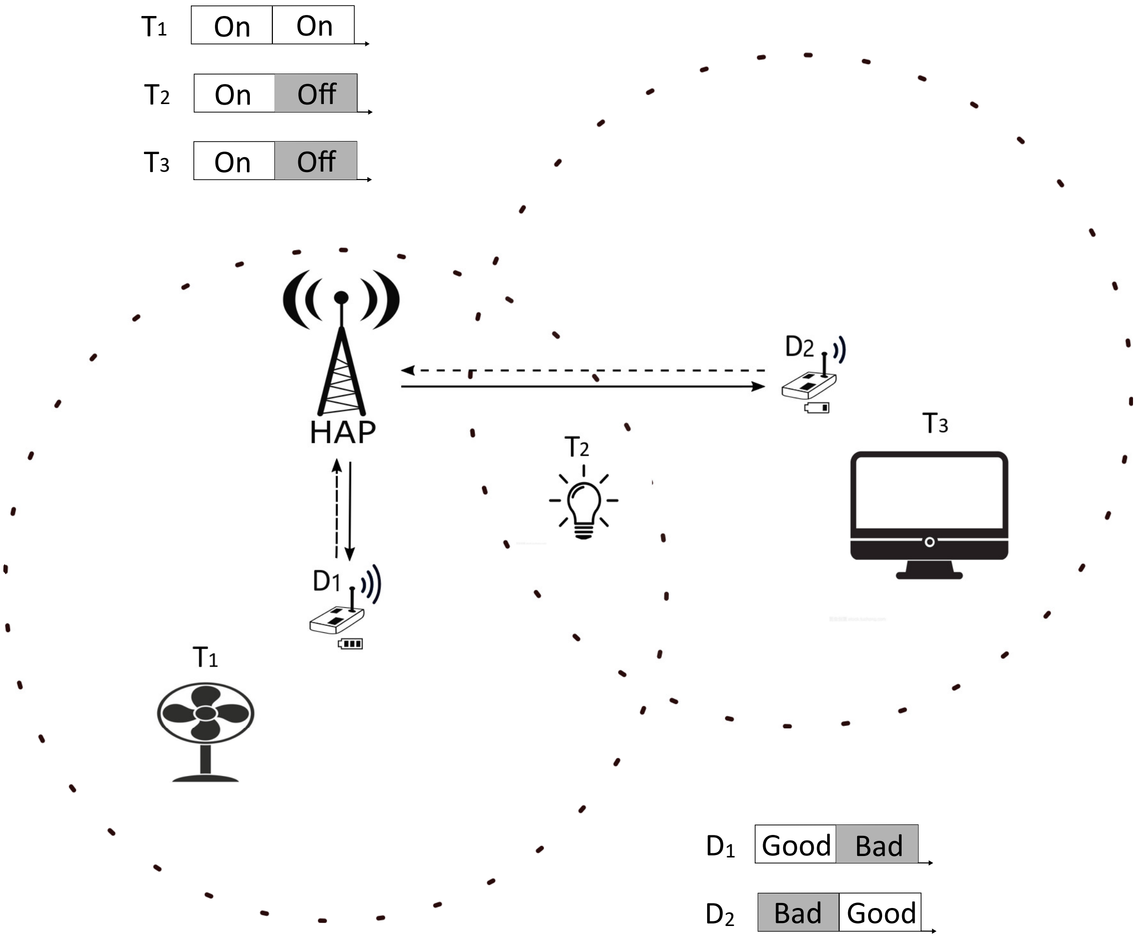

Figure 5.1: An example RF-charging network.

For devices, assume that device D_1 only needs one frame to accumulate energy to generate a sample, while device D_2 needs two frames. This example considers two uplink channel states, i.e., 'Good' and 'Bad'. Specifically, an active device will successfully transmit data to the HAP if the channel state is 'Good'. Otherwise, data transmission fails. As shown in Figure 5.1, in the first frame, the channel state of D_1 and D_2 is 'Good' and 'Bad', respectively. In the second frame, device D_1 has a 'Bad' channel while D_2 has a 'Good' channel. In this example, initially, the HAP does not store any state, and the HAP will update its stored state when it receives data from devices. The AoII of targets increases by one at the start of each frame. If the state stored at the HAP is consistent with a target, the HAP resets the AoII of the target to zero at the end of a frame. Now consider the following cases: (i)

Case-1: both devices D_1 and D_2 are inactive in the first frame and decide to be active in the second frame. In this case, there is no device that transmits data successfully in the first frame. In the second frame, device D_2 transmits successfully. Therefore, as shown in Figure 5.2, the state of target T_1 that is stored at the HAP is always different from the state at T_1. In the second frame, the state of target T_2 and T_3 that is stored at the HAP is consistent with the actual state at target T_2 and T_3. The average AoII of targets per frame is calculated as $\frac{1+2+1+0+1+0}{2 \cdot 3} = \frac{5}{6}$, (ii) *Case-2*: Device D_1 decides to be active in the first frame and be inactive in the second frame. Device D_2 decides to be inactive and active in the first and in the second frame, respectively. In this case, device D_1 transmits successfully in the first frame, and device D_2 transmits successfully in the next frame. Therefore, as shown in Figure 5.2. In the first frame, the state of T_3 at the HAP does not match the actual state at target T_3. Hence, the average AoII of targets per frame is calculated as $\frac{0+0+0+0+1+0}{2 \cdot 3} = \frac{1}{6}$.

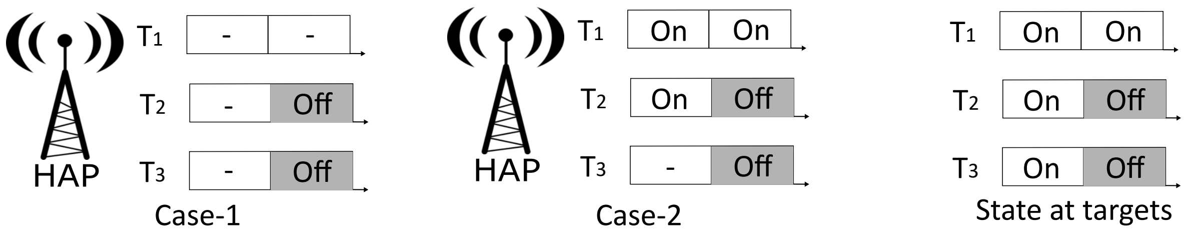

Figure 5.2: State evolution at three targets, and the state evolution at the HAP for Case-1, which results in the highest average AoII, and for Case-2, which results in the lowest AoII. In Case-1, the HAP receives an update from D_2 in the second frame. In Case-2, the HAP receives an update from D_1 and D_2 in the first and second frame, respectively.

From the above example, the problem at the hand is to determine the best active device set in each frame so as to minimize the average AoII. The main challenge is that the battery state and channel state are unknown. Another challenge is that both the HAP/scheduler and devices do not know the cooperation relation between devices, i.e., they do not know two devices monitoring the same target.

The rest of this chapter is structured as follows. Section 5.1 shows the system model and formulates the problem. The proposed Decentralized Q-Learning (DQL) algorithm is presented in Section 5.2 followed by State Space Free Learning (SSFL) algorithm in Section 5.3. After that, Section 5.4 discusses results followed by the conclusion in Section 5.5.

5.1 System Model

Let $\mathcal{D} = \{D_1, D_2, \ldots, D_M\}$ be a set of devices; each device is responsible for monitoring one or more targets. A HAP is responsible for charging these M devices via RF and collecting samples from them. Time is slotted and each frame has index t. There are T frames; each frame has a duration of one second, which means the term power and energy are interchangeable.

In frame t, the HAP charges devices in $\mathcal{D}$ with transmit power P (in Watt) and receives samples from devices via Orthogonal Frequency Division Multiplexing (OFDM). Specifically, each device has a distinct channel for data upload.

In each frame, device D_i selects one of the following actions: (i) sleep, or (ii) active. Let a_i^t be a binary variable, whereby it is set to $a_i^t = 1$ if a device is active; otherwise, $a_i^t = 0$. Let $s^t = \{a_1^t, a_2^t, \ldots, a_M^t\}$ be a vector that records the action of devices in frame t. Let the z-th schedule be defined as $s_z = \{s^t | t = 1, 2, \ldots, T\}$. Let $\mathcal{S}$ denote the collection of schedules.

This chapter assumes block fading. The uplink and downlink channel gain between the HAP and device D_i in frame t is defined as g_{i0}^t and g_{0i}^t, respectively. Specifically, the channel gain is as per the Log-distance path loss , which is calculated as per Eq (3.1). Note that this chapter assumes that devices are only aware of their historical uplink CSI. The historical uplink CSI of device D_i is denoted as g_{i0}.

5.1.1 Targets

There are J targets; each target is indexed by j, and modeled by an N-state Markov chain [140] $\{X_t\}_{t \in T}$ as shown in Figure 5.3; each state is indexed by n. Let the probability of staying in the same state be p, and the probability of transition from state n to $\hat{n}$ is $p_{n,\hat{n}}$. State transitions occur at the beginning of each frame. Let $X_j(t)$ denote the state of target j in frame t.

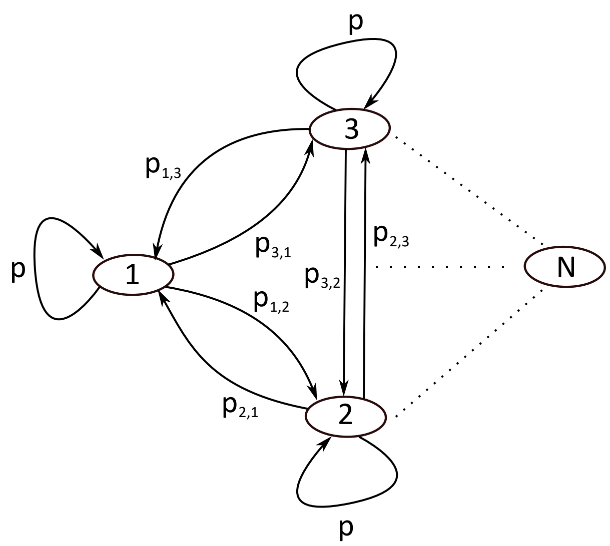

Figure 5.3: An N-state Markov chain model.

5.1.2 Sampling and Buffer

An active device consumes e_s amount of energy to generate a sample before transmission. It does not generate a new sample if its energy storage is lower than e_s. Each device has a buffer that only stores the latest sample. At the end of one frame, a device removes a successfully transmitted sample. Otherwise, it retains the sample. Let $B_i^t \in \{0, 1\}$ denote the buffer state of device D_i. Specifically, $B_i^t = 1$ if device D_i has a sample. Otherwise, $B_i^t = 0$.

5.1.3 Energy Model

Each device has an RF energy harvester [135] with a non-linear energy conversion efficiency. Let e_i^t be the amount of energy received by device D_i in frame t, which is calculated as per Eq (4.1), Eq (4.2), and Eq (4.3). Note that the received energy of device D_i in frame t, i.e., e_i^t, is only available in frame $t + 1$.

When a device, say D_i, is active in frame t, i.e., $a_i^t = 1$, and there is a sample in its buffer, i.e.,$B_i^t = 1$, it consumes all its available energy to transmit a sample. Device D_i saves its energy for future use if it decides to sleep or it does not have a sample. Let E_i^t be the energy storage of device D_i at the end of frame t, which is given as

$$E_i^t = \begin{cases} e_i^t, & a_i^t = 1 \wedge B_i^t = 1, \\ \min(B_{max}, E_i^{t-1} + e_i^t), & a_i^t = 1 \wedge B_i^t = 0 \vee a_i^t = 0, \end{cases} \tag{5.1}$$

where B_{max} is the battery capacity.

In frame t, the energy consumption of device D_i is zero if it decides to sleep or its buffer is empty, i.e., $B_i^t = 0$. Note that $B_i^t = 0$ implies that device D_i does not generate a new sample in frame t. The energy consumption of D_i equals its energy level at the beginning of frame t, i.e., E_i^{t-1}, if D_i is active ($a_i^t = 1$) and its buffer is not empty ($B_i^t = 1$). Let $\hat{e}_i^t(s_z)$ be the energy consumption of device D_i in frame t, which is defined as

$$\hat{e}_i^t(s_z) = \begin{cases} E_i^{t-1}, & a_i^t = 1 \wedge B_i^t = 1, \\ 0, & a_i^t = 1 \wedge B_i^t = 0 \vee a_i^t = 0. \end{cases} \tag{5.2}$$

5.1.4 Transmission Model

When a device, say D_i, decides to be active ($a_i^t = 1$), it consumes e_s amount of energy to generate a sample. It then uses its remaining energy to transmit its sample to the HAP. If the energy level of D_i is lower than e_s and there is a sample in its buffer, it directly transmits the sample. The transmit power of D_i in frame t is given as

$$p_i^t = \begin{cases} E_i^{t-1} - e_s, & E_i^{t-1} > e_s, \\ E_i^{t-1}, & E_i^{t-1} < e_s \wedge B_i^t = 1, \\ 0, & E_i^{t-1} < e_s \wedge B_i^t = 0 \vee E_i^{t-1} = e_s. \end{cases} \tag{5.3}$$

A device's transmission fails if its Signal-to-Noise Ratio (SNR) at the HAP falls below the threshold ζ. Let $I_i^t \in \{0, 1\}$ denote whether device D_i transmits data successfully in time frame t. We have

$$I_i^t = \begin{cases} 1, & \frac{p_i^t g_{i0}^t}{N_0 W} \geq \zeta, \\ 0, & \frac{p_i^t g_{i0}^t}{N_0 W} < \zeta, \end{cases} \tag{5.4}$$

where N_0 and W denotes noise spectral density and channel bandwidth, respectively.

5.1.5 Targets Coverage

Each device is able to monitor one or more targets. Let m_i^j be a binary variable, whereby it is set to $m_i^j = 1$ if device D_i monitors target j; otherwise, $m_i^j = 0$. Let $m_i = \{m_i^1, m_i^2, \ldots, m_i^J\}$ be a vector that records the targets that is monitored by device D_i. Let $X_j^i(t)$ denote the state of target j as recorded by device D_i at time t. Let $\hat{X}_j^i(t)$ denote the state information of target J that is transmitted to the HAP by device D_i at time t, which is defined as

$$\hat{X}_j^i(t) = I_i^t m_i^j X_j^i(t). \tag{5.5}$$

Note that $\hat{X}_j^i(t) = 0$ implies that device D_i does not monitor target j or device D_i does not transmit a packet to the HAP successfully.

5.1.6 State

The HAP only records the newest state information. For example, in the second frame, the HAP receives two state information of target j. One from device D_1 which is obtained in frame one and another from device D_2 which is obtained in frame two. The HAP then records the state information from device D_2. Let $\hat{X}_j(t)$ be the state of target j at the HAP at the end of frame t, and let τ_t^i be the generation time of the packet updated by device D_i in frame t. We have

$$\hat{X}_j(t) = \begin{cases} \hat{X}_j(t-1), & \sum_{i=1}^{M} I_i^t = 0, \\ \hat{X}_j^1(t), & \tau_t^1 = \max(\tau_t^1, \tau_t^2, \ldots, \tau_t^M) \wedge I_1^t m_1^j \neq 0, \\ \hat{X}_j^2(t), & \tau_t^2 = \max(\tau_t^1, \tau_t^2, \ldots, \tau_t^M) \wedge I_2^t m_2^j \neq 0, \\ & \vdots \\ \hat{X}_j^M(t), & \tau_t^M = \max(\tau_t^M, \tau_t^2, \ldots, \tau_t^M) \wedge I_M^t m_M^j \neq 0. \end{cases} \tag{5.6}$$

An assumption is that a state *mismatch* occurs when the state at the HAP is not the same as the state at a target, i.e., when $\hat{X}_j(t) \neq X_j(t)$. On the other hand, the HAP has a *consistent* state of target j.

5.1.7 Age of Incorrect Information

The AoII of a target is defined as the number of frames that have elapsed since a state *mismatch* exists between the HAP and a target [139]. Let $\Delta_j(t)$ denote the AoII of target j at the end of frame t. Let $V_j(t)$ denote the index of the last frame in which the HAP has a *consistent* state of target j. Specifically, $\Delta_j(t) = (t - V_j(t))$ when a state mismatch exists at the end of frame t, i.e., when $\hat{X}_j(t) \neq X_j(t)$. Otherwise, $\Delta_j(t) = 0$. Formally, the term $\Delta_j(t)$ denotes the AoII of target j at the end of frame t is given as

$$\Delta_j(t) = (t - V_j(t))\mathbb{I}\{X_j(t) \neq \hat{X}_j(t)\}, \tag{5.7}$$

where $\mathbb{I}\{X_j(t) \neq \hat{X}_j(t)\}$ is an indicator function that returns one and zero when $\hat{X}_j(t) \neq X_j(t)$ and $\hat{X}_j(t) = X_j(t)$, respectively

Let $\Delta_j^{s_z}(t)$ denote the AoII of target j at the end of frame t when devices use schedule s_z. Let $\bar{\Delta}(s_z)$ be the average AoII of J targets over T frames, which is given as

$$\bar{\Delta}(s_z) = \frac{1}{JT} \sum_{j=1}^{J} \sum_{t=1}^{T} \Delta_j^{s_z}(t). \tag{5.8}$$

5.1.8 The Problem

The problem is to find a schedule in $\mathcal{S}$ to minimize the average AoII of J targets over T frames. Mathematically,

$$\min_{s_z \in \mathcal{S}} \quad \mathbb{E}_{\varphi}\left[\bar{\Delta}(s_z)\right], \tag{5.9}$$

where φ is the joint distribution over random channel gains to/from each device. The main challenge of the problem is that the HAP is incapable to collect the CSI and battery state information of each device when there are many devices that exist in a sensor network since the HAP has to send pilot signals. To address the aforementioned challenge, this chapter proposes two distributed algorithms. Specifically, the next section outlines a distributed Q-Learning algorithm followed by a state space free learning algorithm.

5.2 Distributed Q-Learning Algorithm

The first solution is a Q-Learning (QL) based algorithm. It is a distributed protocol, whereby each device acts independently. The basic idea is that, at the beginning of a frame, each device decides its active probability according to its energy level, buffer state, historical uplink channel state, and AoII of targets that are under its sensing range. This means devices are not required to report their uplink channel state, energy level, and buffer state to the HAP nor devices need to communicate with one another. This helps devices save energy.

Next, Section 5.2.1 outlines a decision problem for devices. Specifically, each device aims to select a probability to be active in each frame so as to minimize the sum of AoII of targets. After that, Section 5.2.2 formulates the decision problem as a Markov Decision Process (MDP).

5.2.1 A Decision Problem

Each device aims to minimize the sum of AoII of its targets. Let the action of device D_i be $\hat{a}_i^t \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, where each value represents a probability to be active. Let $\hat{\Delta}_i^t(\hat{a}_i^t)$ be the sum AoII of targets that under the coverage of device D_i at the end of frame t when device D_i selects $\hat{a}_i^t$. Let $\Delta_j^t(\hat{a}_i^t)$ denote the AoII of target j at the end of frame t when device D_i selects action $\hat{a}_i^t$, which is defined as

$$\hat{\Delta}_i^t(\hat{a}_i^t) = \sum_{j=1}^{J} m_i^j \Delta_j^t(\hat{a}_i^t). \tag{5.10}$$

Let R_i^t be the difference between the AoII of targets of D_i at the end of frame $t-1$ and frame t, which is given as

$$R_i^t(\hat{a}_i^t) = \hat{\Delta}_i^{t-1}(\hat{a}_i^{t-1}) - \hat{\Delta}_i^t(\hat{a}_i^t). \tag{5.11}$$

Formally, the problem of each device is to find the optimal action so as to maximize the following long-term reward:

$$R = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}[\sum_{t=1}^{T} R_i^t(\hat{a}_i^t)], \tag{5.12}$$

where the expectation is taken with respect to joint distribution of channel gains between device D_i and the HAP.

5.2.2 An MDP Model and DQL algorithm

An MDP is defined as a four tuple, i.e., $\{\hat{\mathcal{S}}, \hat{\mathcal{A}}, \mathcal{T}, \mathcal{R}\}$. Specifically, the notation $\hat{\mathcal{S}}$ and $\hat{\mathcal{A}}$ represents the state space and action space, respectively. The reward of taking action $\hat{a}^t$ in state $\hat{s}^t$ is defined as $\mathcal{R}(\hat{s}^{t+1}|\hat{s}^t, \hat{a}^t)$. The transition probability between state $\hat{s}^{t+1}$ and $\hat{s}^t$ after taking action $\hat{a}^t$ is defined as $\mathcal{T}(\hat{s}^{t+1}|\hat{s}^t, \hat{a}^t)$. A policy $\pi(\hat{s})$, where $s \in \mathcal{S}$, maps the state space to the action space. Then the problem (5.12) can be modeled as an MDP as follows:

1. State $\hat{\mathcal{S}}$: The state of a device consists of its energy level, its historical uplink channel state information, its buffer state and summation AoII of targets that under its monitor. Let s_i^t denote the state of device D_i in frame t, which is expressed as $\hat{s}_i^t = \{E_i^{t-1}, g_{i0}, B_i^t, \sum_{j=1}^{J} m_i^j \Delta_j(t)\}$.

2. Action $\hat{\mathcal{A}}$: The action space is defined as $\hat{\mathcal{A}} = [0, 0.2, 0.4, 0.6, 0.8, 1]$, where each element represents a certain probability to be active.

3. Transition probability $\mathcal{T}$: The transition probability is unknown since this chapter considers a model free approach.

4. Reward $\mathcal{R}$: The reward of a device after taking an action in frame t is calculated as per 5.11.

Given above MDP model, this chapter applies the same DQL algorithm as chapter 4 to find the optimal policy, see details in Section 4.3.3.

5.3 State Space Free Learning (SSFL) Algorithm

The second solution is a reinforcement learning-based algorithm, called SSFL. Different from DQL, SSFL does not have a Q-table. Moreover, SSFL does not require uplink channel state, energy level, and buffer state information; this saves on signaling cost.

Let the action of device D_i be $\tilde{a}_i \in [0, \frac{1}{|\tilde{a}_i|}, \ldots, 1]$ where each value represents a probability to be active in each frame over a given duration, i.e., K frames. Then, define for device D_i an action probability vector $\hat{P}_i = \{\hat{P}_{i,0}, \hat{P}_{i,\tilde{a}_i}, \ldots, \hat{P}_{i,1}\}$ where $\hat{P}_{i,\tilde{a}_i}$ denotes the probability that D_i selects action $\tilde{a}_i$.

In SSFL, a device D_i selects its action according to $\hat{P}_i$. Further, SSFL contains Y learning iterations, and each learning iteration contains K frames. At the end of each iteration, the action probability vector of a device will be updated following a reinforcement learning model.

The next section shows the reward formulation of SSFL and the update rule of the so-called action probability vector followed by the detail of the SSFL algorithm.

5.3.1 Reward and Update Rule

Each device D_i computes a reward as follows. Let $\tilde{\Delta}_i^t(\tilde{a}_i^y)$ be the sum of AoII of targets that are under coverage of D_i at the end of frame t when device D_i selects action $\tilde{a}_i$ in the y-th learning iteration. Let $\Delta_j^t(\tilde{a}_i^y)$ denote the AoII of target j at the end of frame t when D_i selects active probability $\tilde{a}_i$ in the y-th learning loop, which is defined as

$$\tilde{\Delta}_i^t(\tilde{a}_i^y) = \sum_{j=1}^{J} m_i^j \Delta_j^t(\tilde{a}_i^y). \tag{5.13}$$

Let $\tilde{\Delta}_i^y(\tilde{a}_i^y)$ denote the sum of AoII of targets that is monitored by device D_i during the y-th learning loop, which is defined as

$$\tilde{\Delta}_i^k(\tilde{a}_i^y) = \sum_{t=yK}^{(y+1)K} \tilde{\Delta}_i^t(\tilde{a}_i^y), \tag{5.14}$$

where the reward constructed by D_i is the difference between the sum of AoII of targets being monitored by device D_i in the $(y-1)$-th iteration and that of the y-th learning loop. The reward is given as

$$\tilde{R}_i^y(\tilde{a}_i^y) = (\tilde{\Delta}_i^{y-1}(\tilde{a}_i^{y-1}) - \tilde{\Delta}_i^y(\tilde{a}_i^y)). \tag{5.15}$$

The probability of selecting action $\tilde{a}_i$ is updated as per the learning automata model in [141]. Specifically, letting $\tilde{a}_i^{'}$ be the action that is not $\tilde{a}_i$, i.e., $\tilde{a}_i^{'} \neq \tilde{a}_i$ ($\tilde{a}_i^{'} \in [0, \frac{1}{|\tilde{a}_i|}, \ldots, 1]$), the update rule of action probability vector $\hat{P}_i$ is given as [142]

$$\hat{P}_{i,\tilde{a}_i} = \hat{P}_{i,\tilde{a}_i} + b \cdot \tilde{R}_i^y(\tilde{a}_i) \cdot (1 - \hat{P}_{i,\tilde{a}_i}), \tag{5.16}$$

$$\hat{P}_{i,\tilde{a}_i^{'}} = \hat{P}_{i,\tilde{a}_i^{'}} - b \cdot \tilde{R}_i^y(\tilde{a}_i) \cdot \hat{P}_{i,\tilde{a}_i^{'}}, \tag{5.17}$$

where the parameter $b \in [0, 1]$ is the learning rate.

5.3.2 Algorithm Details

Algorithm-5.1 shows the steps of SSFL. Initially, a device D_i initializes its action probability vectors, see line 1. Specifically, at the beginning, each action has an equal probability to be selected. In every K frames device D_i selects its active probability as per the ϵ-greedy strategy. In other words, with probability $1 - \tilde{\epsilon}$ device D_i selects $\tilde{a}_i$ with probability $\hat{P}_{i,\tilde{a}_i}$ or it randomly selects its action with probability $\tilde{\epsilon}$, as shown from line 3 to line 8. Next, device D_i calculates the reward of selecting an active probability as per 5.15, and, then, device D_i updates its action probability vector as per 5.16 and 5.17, as shown in line 9 and 10.

Algorithm 5.1: SSFL algorithm for devices.

1 **Initialize:** action probability vector $\hat{P}_i$
2 **for** *every K frames* **do**
3   Generate random number $\tilde{z} \in [0, 1]$;
4   **if** $\tilde{z} \leq \tilde{\epsilon}$ **then**
5   Randomly select an active probability
6   **else**
7   Select $\tilde{a}_i$ with probability $\hat{P}_{i,\tilde{a}_i}$
8   **end**
9   Calculate reward as per Eq. (5.15)
10   Update the action probability vector as per Eq. (5.16) and (5.17)
11 **end**

5.4 Evaluation

The proposed DQL and SSFL algorithm are evaluated using Matlab. Experiments are run over 1000 frames. There are 10 devices, 10 targets, and a HAP. Devices are randomly placed 1 to 6 meters from the HAP. Each target is monitored by one or more devices. The experiments study the following parameters: (i) transmission power P, (ii) SNR threshold ζ, (iii) probability that a target stays in the same state, i.e., p, and (iv) standard deviation μ, which governs the channel condition between the HAP and devices. Sensor nodes consume 0.26 mJ to obtain one sample [143]. The noise power spectral density N_0 is -124 dBm/Hz [144]. The bandwidth W is 2 MHz. The operating frequency band of the HAP is 915 MHz as per [127]. The antenna gain of the HAP and devices is set to 1 dBi and 6.1 dBi respectively [127]. The path loss exponent is 2.5. The battery of devices is initially empty and has a capacity of 1 J. The maximum energy harvesting rate of devices is 24 mW [135]. The parameters a_i and b_i are set as per [135] to 0.014 and 150, respectively. The proposed algorithms are compared against the following rules:

- **Random Pick (RP)**: In each frame, each device randomly determines whether it should be asleep or be active.
- **Round Robin (RR)**: Devices become active in turn or in a round-robin manner. This ensures each device has an equal number of turns to be active.
- **ϵ-Greedy**: A device selects to be active with probability ϵ in the following cases: (i) it has sufficient energy to sample, and (ii) its buffer is not empty; otherwise, it becomes active or enters sleep mode randomly.
- **Perfect Information Selection (PIS)**: This rule yields the optimal result. This is because each device knows the following information: (i) whether the state information recorded in its buffer is consistent with the current state of its monitored targets, and (ii) its battery state and uplink channel state. In one frame, a device only selects to be active in the following cases: (i) it has sufficient energy to sample and then transmit successfully, or (ii) the state information recorded in its buffer is consistent with its monitored targets and it has sufficient energy to transmit successfully. PIS does not experience energy waste caused by transmission failures and ensures that the HAP obtains the freshest state information of all targets at the end of each frame.

5.4.1 Convergence

This experiment studies the convergence of DQL and SSFL algorithms. The simulation of DQL and SSFL runs 400 and 1000, respectively. Referring to Figure 5.4 and Figure 5.5, the proposed DQL and SSFL algorithm achieve lower average AoII than RR, RP, ϵ-Greedy after converging. This is because when using DQL and SSFL, devices that have sufficient energy to sample and transmit will be active with a higher probability, which helps the HAP obtain more state information updates in each frame and results in a lower average AoII.

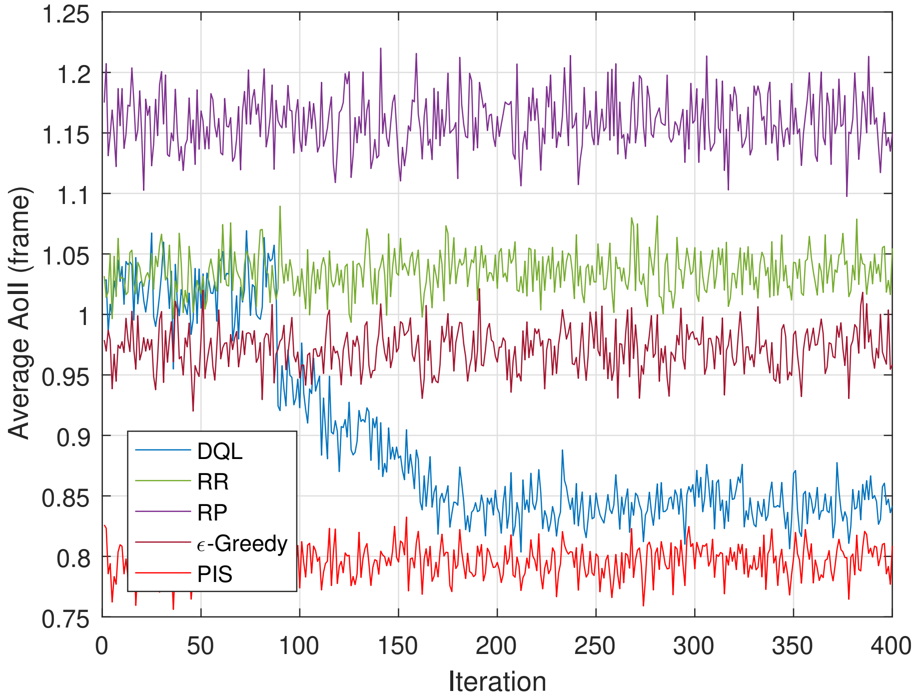

Figure 5.4: Converge curve for DQL algorithm.

5.4.2 Charging Power

This experiment investigates the impact of charging power, i.e., P, on the average AoII. Specifically, this experiment studies the following transmit power values (in Watts): $P \in \{1, 2, 3, 4, 5\}$.

Figure 5.6, shows the average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL

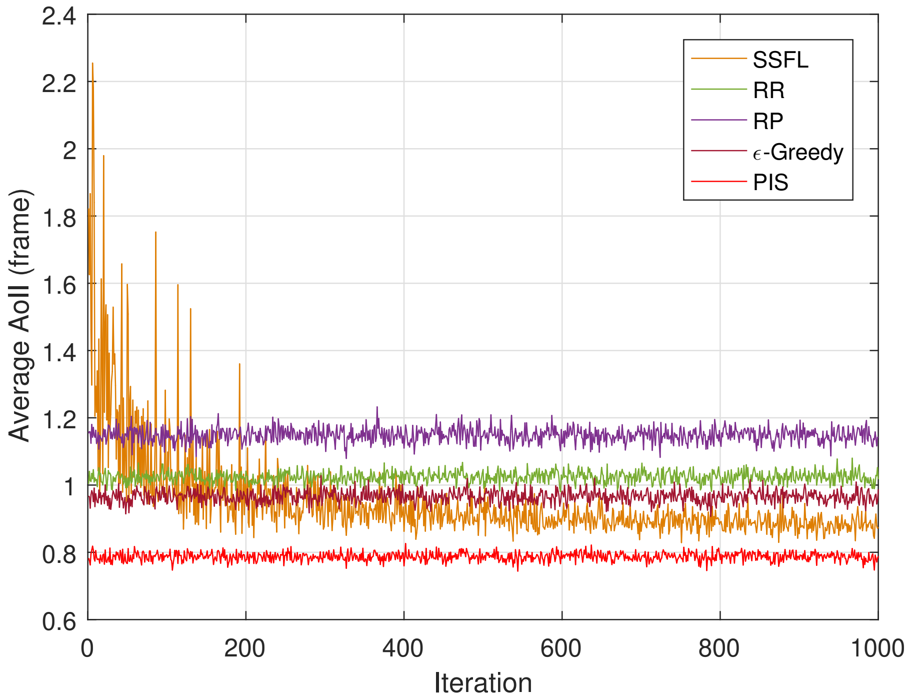

Figure 5.5: Converge curve for SSFL algorithm.

when the HAP uses a different charging power. According to Figure 5.6, when the transmission power of the HAP increases from 1 W to 5 W, the average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL decreases by around 68.8%, 74.1%, 73.1%, 80.4%, 80.3%, and 80.7%, respectively. The reason is that a higher charging power P, i.e., 5 W, results in devices receiving more energy in each frame, which helps devices accumulate sufficient energy to sample and transmit faster than when using a low charging power, i.e., 1 W. This allows devices to update the state information of their monitored targets to the HAP. According to Figure 5.6, the average AoII of DQL and SSFL is around 4.6% and 10% higher than that of PIS, respectively. This is because the DQL and SSFL algorithms do not have perfect uplink channel state information of devices. Consequently, DQL and SSFL algorithms are unable to reduce transmission failures to zero and have a higher average AoII.

Figures 5.7 and 5.8 show the numbers of successful packet transmissions achieved by SSFL and DQL respectively when the charging power varies from 1 W to 5 W. Referring to Figures 5.7 and 5.8, the number of successful packet transmissions

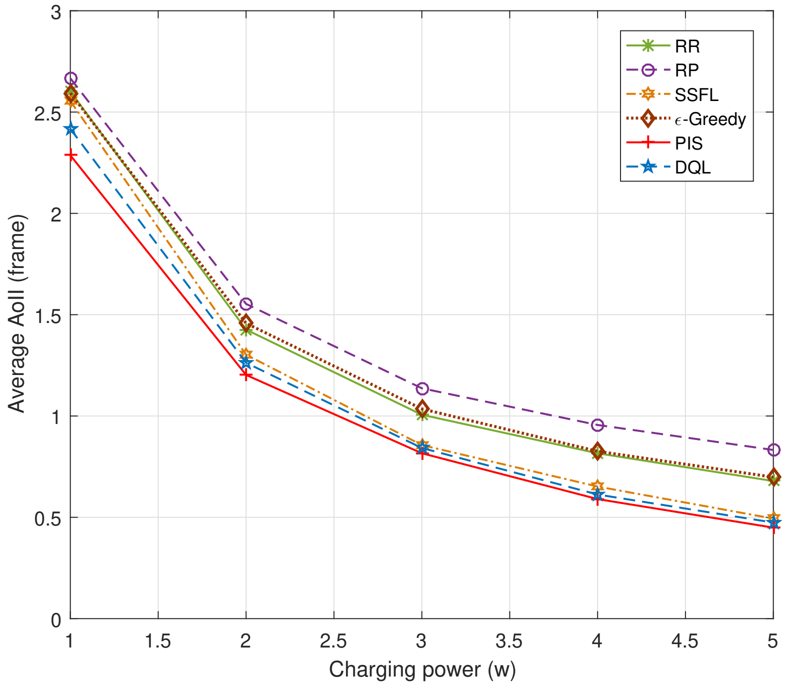

Figure 5.6: Average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL versus HAP transmit power.

attained by DQL is around 4.2%, 4.0%, 5.6%, 3.0%, 4.8% higher than that of SSFL when the charging power P increases from 1 W to 5 W. This is because for DQL, devices select their probability of becoming active according to their energy level, buffer state, and historical uplink channel state, which allows them to more accurately judge when to be active. In contrast, SSFL does not consider such information. Moreover, according to Figures 5.7 and 5.8, devices that are located close to the HAP, i.e., $\{D_1, D_2, \ldots, D_5\}$, have a higher number of successful packet transmissions than devices located farther away from the HAP, i.e., $\{D_6, D_7, \ldots, D_1\}$. This is because these devices have a higher received power, which results in higher charging efficiency, meaning that they can obtain sufficient energy to sample and transmit faster than devices located farther away. Moreover, devices that have a high probability to have sufficient energy to sample and update the HAP will be active when using DQL and SSFL. The faster rate of energy accumulation coupled with DQL and SSFL helps devices that are close to the HAP to have a higher number of successful transmissions.

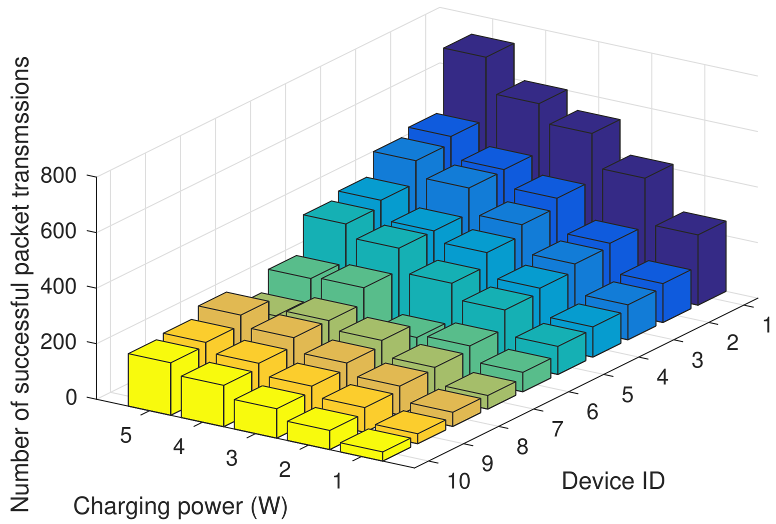

Figure 5.7: Number of successful packet transmissions versus HAP transmit power for the SSFL algorithm.

5.4.3 Signal-to-Noise Ratio Threshold

To study SNR threshold (in dB), i.e., ζ, on the average AoII, it is set to one of the following values: $\zeta \in \{3, 6, 9, 12, 15\}$.

Figure 5.9 shows the average AoII of RP, RR, ϵ-Greedy, PIS, and the proposed two algorithms when considering different SNR thresholds. Referring to Figure 5.9, when the SNR threshold varies from 3 dB to 15 dB, the average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL increases by around 53.6%, 54.8%, 77.8%, 11.4%, 23.4%, and 68.1%, respectively. The reason is that a higher SNR threshold, i.e., 15 dB, results in devices needing more frames to accumulate sufficient energy to transmit a packet to the HAP. This requires the HAP to have more frames to obtain

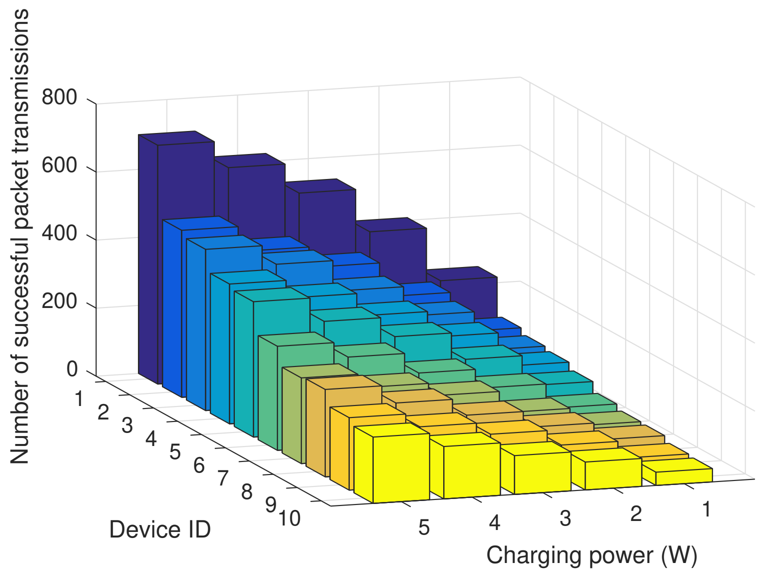

Figure 5.8: Number of successful packet transmissions versus HAP transmit power when using DQL.

a state information that is consistent with its monitored targets, which results in a higher average AoII.

Referring to Figure 5.9, when $\zeta = 3$ dB, the average AoII of SSFL is around 4.6% higher than that of DQL. At $\zeta = 15$ dB, the average AoII of SSFL is about 42.2% higher than that of the DQL. The gap between DQL and SSFL increases when the SNR threshold varies from 3 dB to 15 dB. This is because SSFL decides the active probability of devices over a specific number of frames, i.e., 1000 frames, instead of deciding the active probability in each frame according to the energy level of devices and historical uplink channel state information. SSFL is thus incapable of judging if a device has sufficient energy to transmit in one frame, which causes more energy waste over the entire time horizon, i.e., 1000 frames, for $\zeta = 15$ dB. For devices located far from the HAP, their higher energy consumption requires them to have more frames to successfully transmit a packet to the HAP. The result is that the number of successful packet transmissions from devices located further from the HAP decreases significantly when the SNR threshold increases from 3 dB to 15 dB. This is confirmed in Figures 5.10 and 5.11. Specifically, when using SSFL and a

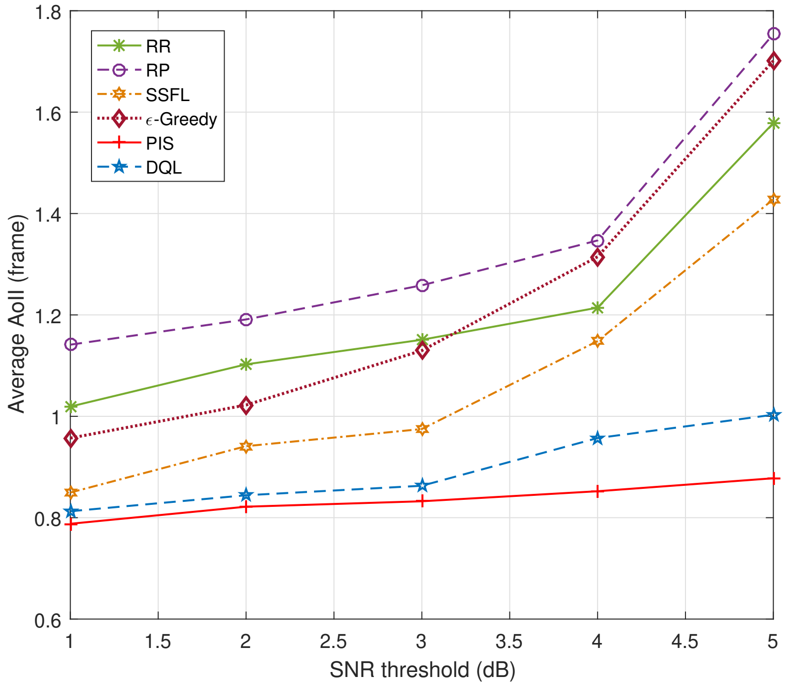

Figure 5.9: Average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL versus SNR threshold.

SNR value of 3 dB, the numbers of successful packet transmissions from devices D_6, D_7, D_8, D_9, and D_{10} are around 10.5%, 5.9%, 4.9%, 5.8%, and 8.1% lower than that of DQL, respectively. When the SNR threshold increases to 15 dB and using SSFL, the numbers of successful packet transmissions from devices D_6, D_7, D_8, D_9, and D_{10} are 24.1%, 30.1%, 39.6%, 25%, and 37% lower than that of DQL, respectively.

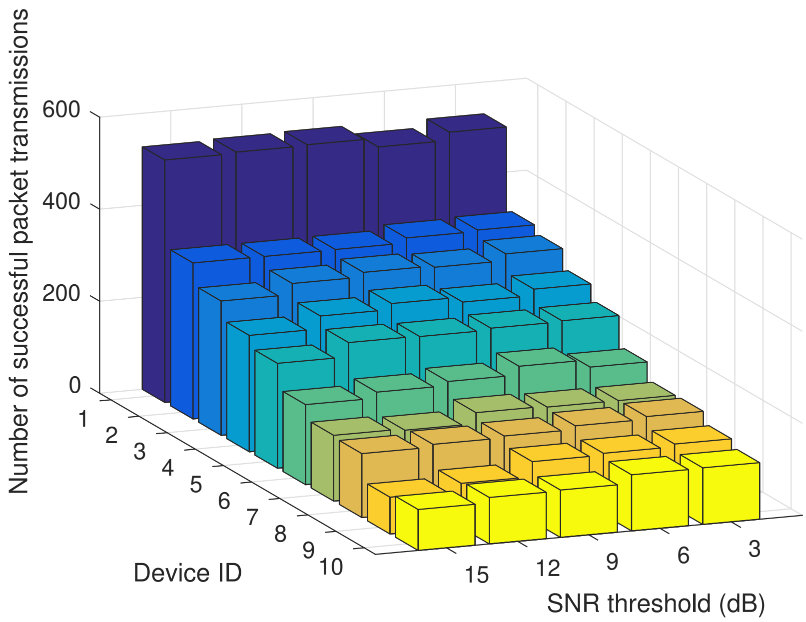

Figure 5.10: Number of successful packet transmissions versus SNR threshold when using DQL.

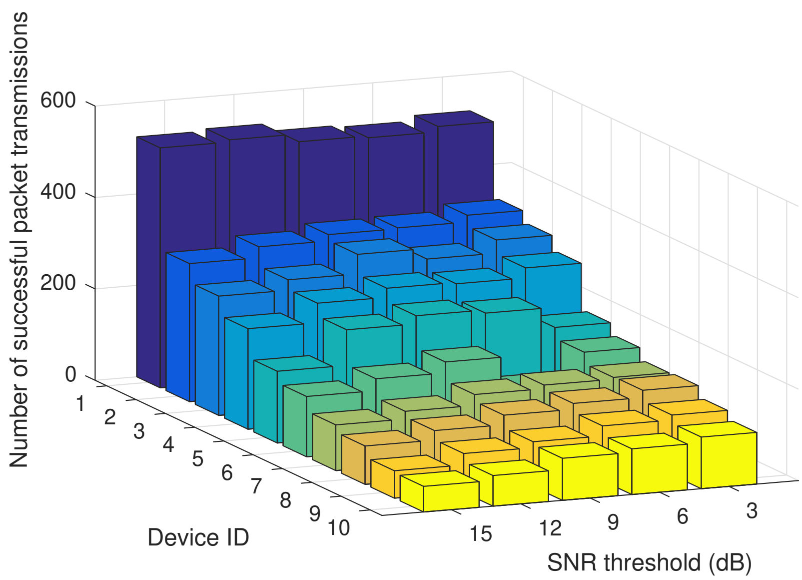

Figure 5.11: Number of successful packet transmissions versus SNR threshold when using SSFL.

5.4.4 State Transition Probability

This section studies the state transition probability of targets, i.e., parameter p. Specifically, the experiment uses following p values: $p \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. Figure 5.12 shows the average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL when considering the different values of parameter p. Referring to Figure 5.12, when the value of p varies from 0 to 0.6, the average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL decreases by about 35.6%, 34.6%, 39.6%, 40.7%, 40.5%, and 35.1%, respectively. This is because a high p value, i.e., 0.6, means that in one frame, targets will stay in the same state as the last frame with a high probability, i.e., 60%. This means the AoII of a target has a high probability to stay at zero in case no devices transmit successfully. This results in the AoII of targets under the coverage devices that are located further away from the HAP having a low value. This helps reduce the average AoII. In contrast, a low p value, i.e., zero, means that in each frame targets will jump to a different state with a high probability, i.e., 100%. This means that the state information that is recorded at the HAP has a high probability to be different from the actual target state. This leads to a higher average AoII.

Referring to Figures 5.13 and 5.14, the number of successful packet transmissions is almost unaffected by the varying value of p. This is because the parameter p does not influence the received energy of devices and the channel state between devices and the HAP. This means that the energy accumulation and data transmission process are not influenced by the varying value of p. This results in a nearly unchanged number of successful packet transmissions for PIS and DQL.

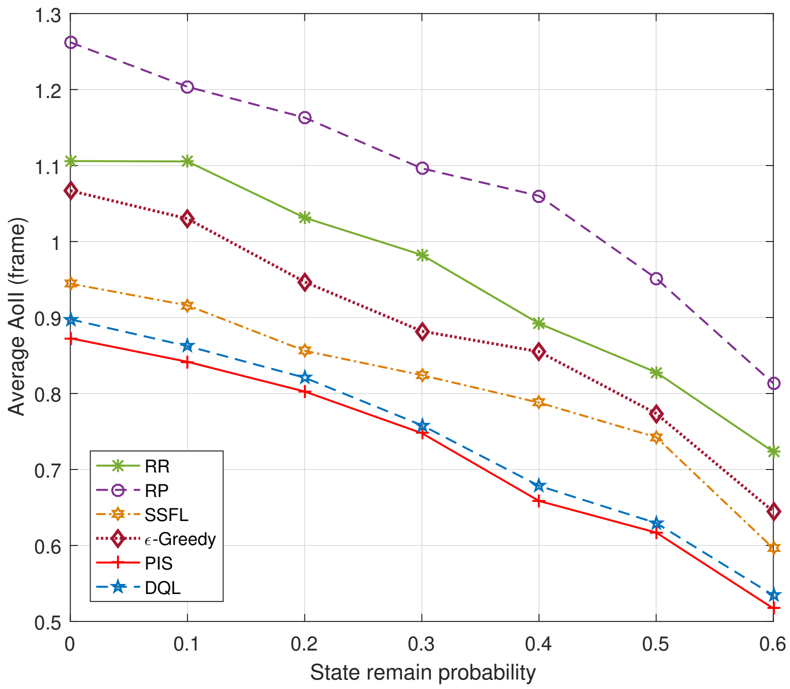

Figure 5.12: Average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL versus the probability that a target stays in the same state.

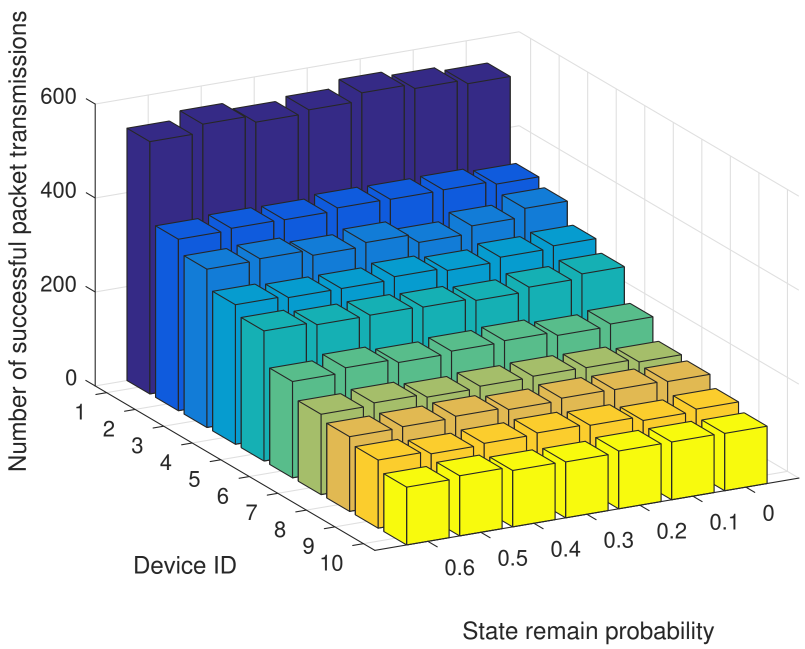

Figure 5.13: Number of successful packet transmissions versus probability of state in the same state when using PIS algorithm.

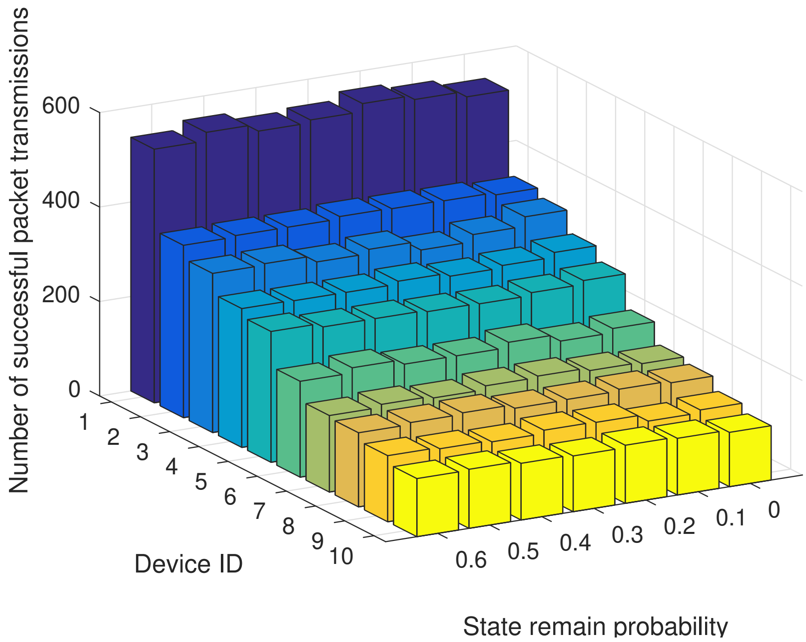

Figure 5.14: Number of successful packet transmissions versus the probability that a target stays in the same state when using DQL.

5.4.5 Channel Variation

The last experiment studies the influence of channel variation on the average AoII. Specifically, this experiment studies the influence of variable μ, where μ value has a value in the set $\{1, 2, 3, 4, 5\}$. Recall that the variable μ is the standard deviation of variable $\mathcal{X}$ which relates to the severity of channel condition.

As shown in Figure 5.15, when the standard deviation μ increases from one to five, the average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL decreases by about 10.2%, 21.0%, 13.4%, 13.6%, 9.6%, and 13.6%, respectively. This is because $\mu = 5$ results in a higher channel gain on average. This can be seen in Figure 5.16, where the occurrence of large channel gains, i.e., greater than 0.00008, is much higher than when $\mu = 1$. This leads to an increase in charging power at devices, see Figure 5.17. Consequently, devices are able to report the state of targets to the HAP more frequently, which leads to a lower average AoII. According to Figure 5.15, the gap between DQL and PIS increases from about 1.6% to 6.0% when the standard deviation changes from one to five. This is because a higher value corresponds to severe channel conditions. In this case, the lack of perfect channel state information causes a device using DQL to make incorrect decisions, i.e., selecting to be active when its channel to the HAP is poor. This results in a higher average AoII.

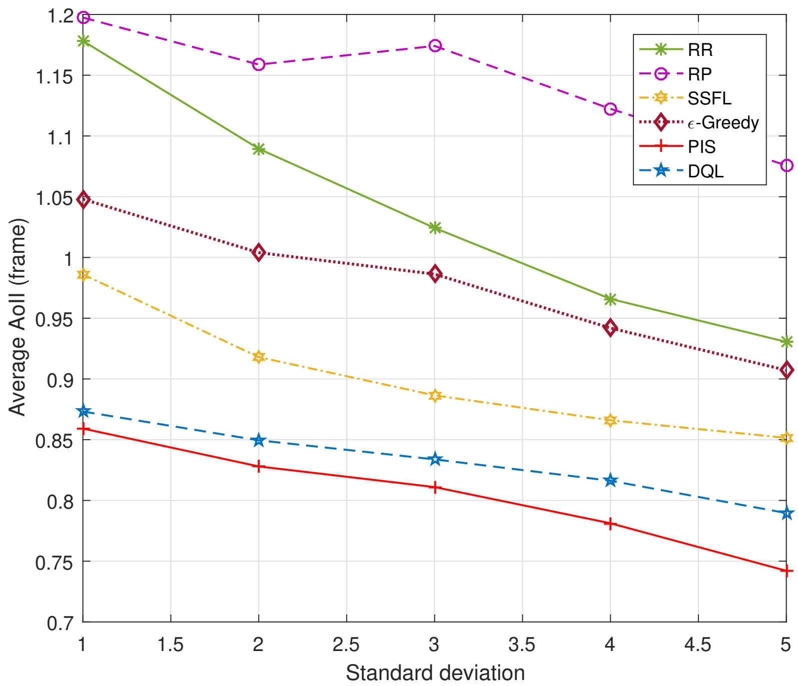

Figure 5.15: Average AoII of RP, RR, ϵ-Greedy, PIS, DQL, and SSFL versus standard deviation μ.

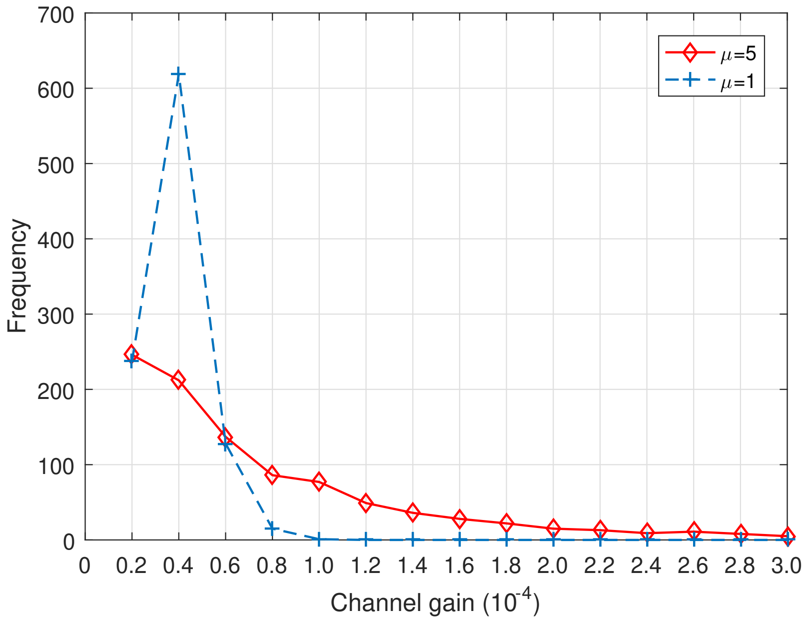

Figure 5.16: Frequency distribution of channel gains of device D_{10}.

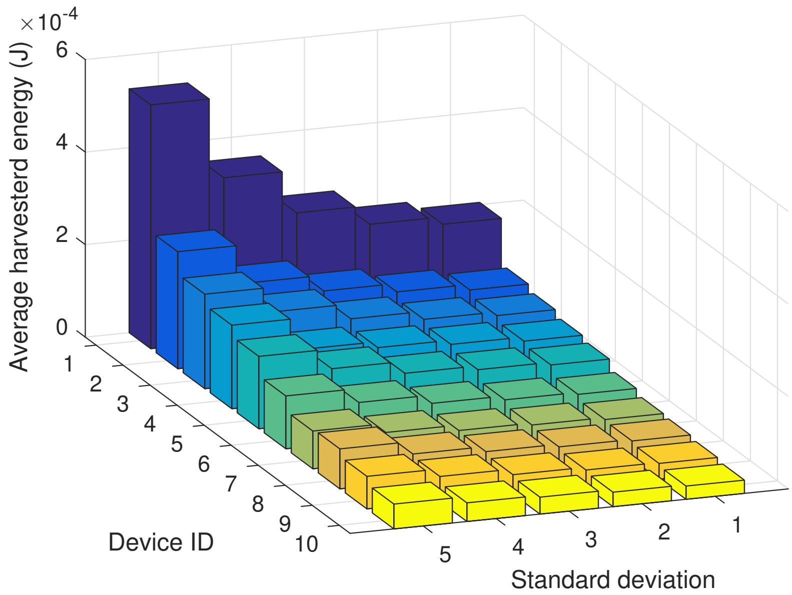

Figure 5.17: Average harvested energy versus standard deviation μ.

5.5 Conclusion

This chapter has considered an active device set selection problem in an RF-charging network where multiple devices are powered by a HAP and are responsible for monitoring multiple targets. Its aim is to find the best set of active devices in each frame so as to minimize the average AoII of targets. Further, it considers a challenging issue whereby the HAP is unaware of the CSI and battery state of devices. To this end, this chapter outlines two distributed algorithms, namely DQL and SSFL, to determine the active device set in each frame. The simulation results show that by activating devices when they have sufficient energy to sample and transmit DQL and SSFL always achieve a lower average AoII than RP, RR, ϵ-Greedy. Moreover, DQL achieves a lower average AoII than SSFL in all experiments since DQL makes decisions according to the battery state and historical CSI of a device while SSFL does not consider such information.

Chapter 6

Conclusion

This thesis has addressed a number of challenging device selection problems in energy-harvesting IoT networks. Specifically, as channel resources are limited, a hybrid access point (HAP) can only select a subset of devices to transmit. The main challenge, however, is that the search space or subsets of devices grow exponentially with the number of devices. Another key challenge is that the HAP does not have channel and energy level information when it selects devices to transmit in each frame. To date, many solutions have been proposed to select devices in order to optimize metrics such as sum rate. However, these solutions consider devices powered by an ambient energy source such as wind and solar energy. In contrast, this thesis considers devices with a dedicated RF energy source. In this respect, the works that consider one or more dedicated energy sources assume that a scheduler or HAP that has perfect channel gain or energy information of devices. However, obtaining both information is impractical in large-scale networks because it requires the HAP/scheduler to poll devices, which incurs high signaling and energy cost.

Motivated by the above gaps, this thesis aims to design centralized and decentralized algorithms to select the best set of devices to sample and transmit in order to maximize one of the following objectives: sum rate, age of information (AoI) or age of incorrect information (AoII). This thesis first investigates device selection in a wireless powered communication network (WPCN) with imperfect channel and energy state information. To this end, Chapter 3 outlines two centralized algorithms, namely cross-entropy algorithm and Gibbs$^+$. The results show that both algorithms are capable of selecting a good set of devices/sensors to transmit even though the scheduler/hybrid access point has imperfect channel state information. Advantageously, the proposed cross-entropy and Gibbs$^+$ algorithms respectively achieve 99% and 98% of the theoretical maximum throughput as computed by PIS, which has perfect information.

The next studied device selection problem concerns AoI, which quantifies the information freshness of samples collected by devices. In this respect, this thesis considers the problem of optimizing information freshness in a network with RF-energy harvesting wireless devices. A HAP charges these devices and instructs a subset of devices to sample targets and transmit their samples to a HAP. Unlike prior works, this thesis has considered a HAP without channel state information of devices. To address this challenge, this thesis has outlined the first decentralized reinforcement learning algorithms for the problem at hand. Specifically, it has outlined a Distributed Q-Learning (DQL) algorithm that enables a HAP to select devices without knowing their uplink channel state information and battery state. DQL achieves at most 48%, 57%, and 61% lower average AoI than round robin, random pick, and AoI-Greedy policy, respectively. The average AoI of DQL is only around 7% higher than the optimal selection strategy that requires channel state information and the energy level of devices.

Lastly, this thesis has studied minimizing AoII in a WPCN that consists of a HAP and multiple energy harvesting devices. Devices are responsible for sensing targets and transmitting the state of targets to the HAP. The HAP's objective is to minimize the AoII of targets. Unlike prior works, sensors/devices monitor multiple targets, and the HAP does not have channel state information and energy level of sensors/devices. To select devices without the said information, this thesis has proposed a Distributed Q-Learning (DQL) algorithm and a State Space Free

Learning (SSFL) algorithm. The results show that DQL and SSFL always achieve a lower average AoII than round Robin, random pick, and ϵ-Greedy. The average AoII of DQL and SSFL are only around 4.6% and 10% higher than the optimal selection strategy, respectively.

There are many possible research directions. As discussed in Chapter 3, and Chapter 4, when considering WPCNs, sensors/devices located far from a HAP require more time to accumulate sufficient energy to sample and transmit than devices located close to the HAP. In this case, devices located close to the HAP will be selected more frequently since they are more likely to have higher energy levels and better channel states, which leads to a fairness problem. In this respect, a possible solution is to employ a mobile HAP. A key problem is to maximize sum-rate by jointly optimizing the trajectory of mobile hybrid access points and set of transmitting devices. Another future work is to consider mobile targets such as vehicles. The key research problem is to forecast the trajectories of mobile targets and select the best set of devices to sample and transmit in order to minimize AoI or AoII of targets.

Bibliography

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Networks*, vol. 54, pp. 2787–2805, Oct. 2010.

[2] A. Whitmore, A. Agarwal, and L. Da Xu, "The internet of things—a survey of topics and trends," *nf. Syst. Front.*, vol. 17, pp. 261–274, Mar. 2015.

[3] D. Niyato, E. Hossain, and S. Camorlinga, "Remote patient monitoring service using heterogeneous wireless access networks: architecture and optimization," *IEEE J. Sel. Areas Commun.*, vol. 27, pp. 412–423, May 2009.

[4] M. N. Bhuiyan, M. M. Rahman, M. M. Billah, and D. Saha, "Internet of things (IoT): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities," *IEEE Internet Things J.*, vol. 8, pp. 10474–10498, July 2021.

[5] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, pp. 16–32, Jan. 2019.

[6] J. I. Vazquez and D. Lopez-de Ipina, "Social devices: autonomous artifacts that communicate on the internet," in *The Internet of Things: First International Conference*, (Berlin), pp. 308–324, Mar. 2008.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surv. Tutorials*, vol. 17, pp. 2347–2376, June 2015.

[8] S. Karpischek, F. Michahelles, F. Resatsch, and E. Fleisch, "Mobile sales assistant-an NFC-based product information system for retailers," in *First International Workshop on Near Field Communication*, (Hagenberg, Austria), pp. 20–23, Feb. 2009.

[9] F. Delmastro, "Pervasive communications in healthcare," *Comput. Commun.*, vol. 35, no. 11, pp. 1284–1295, 2012.

[10] C. Chen, D. J. Cook, and A. S. Crandall, "The user side of sustainability: Modeling behavior and energy usage in the home," *Pervasive Mob. Comput.*, vol. 9, pp. 161–175, Feb. 2013.

[11] J. Ma, X. Zhou, S. Li, and Z. Li, "Connecting agriculture to the internet of things through sensor networks," in *International conference on internet of things and 4th international conference on cyber, physical and social computing*, (Dalian, China), pp. 184–187, Oct. 2011.

[12] D. Yan-e, "Design of intelligent agriculture management information system based on IoT," in *Fourth International Conference on Intelligent Computation Technology and Automation*, vol. 1, (Shenzhen, China), pp. 1045–1049, Mar. 2011.

[13] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "Opportunistic IoT: Exploring the social side of the Internet of Things," in *IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, (Wuhan, China), pp. 925–929, May 2012.

[14] A. Ilic, T. Staake, and E. Fleisch, "Using sensor information to reduce the

carbon footprint of perishable goods," *IEEE Pervasive Comput.*, vol. 8, pp. 22–29, Dec. 2008.

[15] A. Dada and F. Thiesse, "Sensor applications in the supply chain: the example of quality-based issuing of perishables," in *The Internet of Things: First International Conference*, (Zurich, Switzerland), pp. 140–154, Mar. 2008.

[16] A. Dias, L. Gorzelniak, R. A. Jörres, R. Fischer, G. Hartvigsen, and A. Horsch, "Assessing physical activity in the daily life of cystic fibrosis patients," *Pervasive Mob. Comput.*, vol. 8, no. 6, pp. 837–844, 2012.

[17] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The smart thermostat: using occupancy sensors to save energy in homes," in *Proceedings of the 8th ACM conference on embedded networked sensor systems*, (Zurich, Switzerland), pp. 211–224, Nov. 2010.

[18] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Des. Test Comput.*, vol. 18, pp. 62–74, Apr. 2001.

[19] J. Singh, R. Kaur, and D. Singh, "A survey and taxonomy on energy management schemes in wireless sensor networks," *J. Syst. Archit.*, vol. 111, p. 101782, Dec. 2020.

[20] Shinkoh, "KP1430 data sheet." https://www.shinkoh-elecs.jp/wp-content/uploads/2015/07/C_KP1430_20A.pdf/, 2015.

[21] T. Connectivity, "TE connectivity HTU31 sensors data sheet." https://www.tti.com/content/dam/ttiinc/manufacturers/te-connectivity/PDF/HTU31_Sensors_Datasheet.pdf/, 2022.

[22] Sensirion, "Sensirion SFM3003 series data sheet." https://www.mouser.com/datasheet/2/682/Sensirion_Mass_Flow_Meters_SFM3003_Datasheet-2492120.pdf/, 2022.

[23] Shinkoh, "KP1650 data sheet." https://www.shinkoh-elecs.jp/wp-content/uploads/2015/07/C_KP1650_1651_20A.pdf/, 2015.

[24] T. INSTRUMENTS, "TMAG5328 data sheet." https://www.ti.com/lit/ds/symlink/tmag5328.pdf?ts=1677661053933/, 2022.

[25] Onsemi, "AR0130CS data sheet." https://www.onsemi.com/download/data-sheet/pdf/ar0130cs-d.pdf/, 2016.

[26] Kavlico, "Data Sheet P1A Pressure Sensor." https://www.cdiweb.com/datasheets/kavlico/p1a_data_sheet_letter.pdf/, 2016.

[27] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surv. Tutorials*, vol. 13, pp. 443–461, July 2010.

[28] D. Ramasur and G. Hancke, "A wind energy harvester for low power wireless sensor networks," in *IEEE International Instrumentation and Measurement Technology Conference Proceedings*, (Graz, Austria), pp. 2623–2627, July 2012.

[29] R. V. Prasad, S. Devasenapathy, V. S. Rao, and J. Vazifehdan, "Reincarnation in the ambiance: Devices and networks with energy harvesting," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 195–213, 2013.

[30] J. A. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive Comput.*, vol. 4, pp. 18–27, Mar. 2005.

[31] L. Xie and M. Cai, "Human motion: Sustainable power for wearable electronics," *IEEE Pervasive Comput.*, vol. 13, pp. 42–49, Oct. 2014.

[32] M.-L. Ku, W. Li, Y. Chen, and K. R. Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Commun. Surv. Tutorials*, vol. 18, pp. 1384–1412, Oct. 2015.

[33] W. K. Seah, Z. A. Eu, and H.-P. Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-survey and challenges," in *International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, (Aalborg, Denmark), pp. 1–5, July 2009.

[34] D. Mishra, S. De, S. Jana, S. Basagni, K. Chowdhury, and W. Heinzelman, "Smart RF energy harvesting communications: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 53, pp. 70–78, Apr. 2015.

[35] V. Leonov, "Thermoelectric energy harvesting of human body heat for wearable sensors," *IEEE Sens. J.*, vol. 13, pp. 2284–2291, June 2013.

[36] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, "Energy harvesting from human and machine motion for wireless electronic devices," *Proc. IEEE*, vol. 96, pp. 1457–1486, Sept. 2008.

[37] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. Leung, and Y. L. Guan, "Wireless energy harvesting for the internet of things," *IEEE Commun. Mag.*, vol. 53, pp. 102–108, June 2015.

[38] H. Jabbar, Y. S. Song, and T. T. Jeong, "RF energy harvesting system and circuits for charging of mobile devices," *IEEE Trans. Consum. Electron.*, vol. 56, pp. 247–253, Jan. 2010.

[39] Z. Wang, W. Zhang, D. Jin, H. Xie, and X. Lv, "A full-wave RF energy harvester based on new configurable diode connected MOSFETs," in *IEEE International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, vol. 1, (Beijing, China), pp. 117–119, Dec. 2016.

[40] A. K. Moghaddam, J. H. Chuah, H. Ramiah, J. Ahmadian, P.-I. Mak, and R. P. Martins, "A 73.9%-efficiency CMOS rectifier using a lower DC feeding

(LDCF) self-body-biasing technique for far-field RF energy-harvesting systems," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 64, pp. 992–1002, Apr. 2017.

[41] M. Stoopman, S. Keyrouz, H. J. Visser, K. Philips, and W. A. Serdijn, "Co-design of a CMOS rectifier and small loop antenna for highly sensitive RF energy harvesters," *IEEE J. Solid-State Circuits*, vol. 49, pp. 622–634, Feb. 2014.

[42] M. A. Abouzied and E. Sánchez-Sinencio, "Low-input power-level CMOS RF energy-harvesting front end," *IEEE Trans. Microwave Theory Tech.*, vol. 63, pp. 3794–3805, Nov. 2015.

[43] P.-H. Hsieh, C.-H. Chou, and T. Chiang, "An RF energy harvester with 44.1% PCE at input available power of -12 dBm," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 62, pp. 1528–1537, June 2015.

[44] Z. Hameed and K. Moez, "A 3.2 v–15 dBm adaptive threshold-voltage compensated RF energy harvester in 130 nm CMOS," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 62, pp. 948–956, Apr. 2015.

[45] R. Bergès, L. Fadel, L. Oyhenart, V. Vigneras, and T. Taris, "A dual band 915 MHz/2.44 GHz RF energy harvester," in *European Microwave Conference (EuMC)*, pp. 307–310, Dec. 2015.

[46] V. Kuhn, C. Lahuec, F. Seguin, and C. Person, "A multi-band stacked RF energy harvester with RF-to-DC efficiency up to 84%," *IEEE Trans. Microwave Theory Tech.*, vol. 63, pp. 1768–1778, May 2015.

[47] D. Michelon, E. Bergeret, A. Di Giacomo, and P. Pannier, "RF energy harvester with sub-threshold step-up converter," in *IEEE International Conference on RFID (RFID)*, (Orlando, FL, USA), pp. 1–8, June 2016.

[48] Y. Lu, H. Dai, M. Huang, M.-K. Law, S.-W. Sin, U. Seng-Pan, and R. P. Martins, “A wide input range dual-path CMOS rectifier for RF energy harvesting,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 64, pp. 166–170, Feb. 2016.

[49] J. C. Kwan and A. O. Fapojuwo, “Sum-throughput maximization in wireless sensor networks with radio frequency energy harvesting and backscatter communication,” *IEEE Sens. J.*, vol. 18, pp. 7325–7339, July 2018.

[50] Z. Hadzi-Velkov, I. Nikoloska, G. K. Karagiannidis, and T. Q. Duong, “Wireless networks with energy harvesting and power transfer: Joint power and time allocation,” *IEEE Signal Process Lett.*, vol. 23, pp. 50–54, Nov. 2015.

[51] S. Kim, R. Vyas, J. Bito, K. Niotaki, A. Collado, A. Georgiadis, and M. M. Tentzeris, “Ambient RF energy-harvesting technologies for self-sustainable standalone wireless sensor platforms,” *Proc. IEEE*, vol. 102, pp. 1649–1666, Nov. 2014.

[52] R. Vyas, H. Nishimoto, M. Tentzeris, Y. Kawahara, and T. Asami, “A batteryless, energy harvesting device for long range scavenging of wireless power from terrestrial TV broadcasts,” in *IEEE/MTT-S International Microwave Symposium Digest*, pp. 1–3, June 2012.

[53] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. R. Smith, “Powering the next billion devices with Wi-Fi,” in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, (Heidelberg, Germany), pp. 1–13, Dec. 2015.

[54] L. R. Varshney, “Transporting information and energy simultaneously,” in *IEEE international symposium on information theory*, (Toronto, Canada), pp. 1612–1616, Aug. 2008.

[55] H. Lee, Y. Kim, J. H. Ahn, M. Y. Chung, and T.-J. Lee, “Wi-Fi and wireless

power transfer live together," *IEEE Commun. Lett.*, vol. 22, pp. 518–521, May 2017.

[56] H. Ju and R. Zhang, "Throughput maximization in wireless powered communication networks," *IEEE Trans. Wireless Commun.*, vol. 13, pp. 418–428, Jan. 2014.

[57] E. I. George and R. E. McCulloch, "Variable selection via Gibbs sampling," *J. Am. Stat. Assoc.*, vol. 88, no. 423, pp. 881–889, 1993.

[58] F. Iannello, O. Simeone, and U. Spagnolini, "Optimality of myopic scheduling and whittle indexability for energy harvesting sensors," in *46th Annual CISS*, (Princeton, NJ, USA), pp. 1–6, Mar. 2012.

[59] P. Blasco, D. Gündüz, and M. Dohler, "Low-complexity scheduling policies for energy harvesting communication networks," in *IEEE ISIT*, (Istanbul, Turkey), pp. 1601–1605, July 2013.

[60] P. Blasco and D. Gündüz, "Multi-access communications with energy harvesting: A multi-armed bandit model and the optimality of the myopic policy," *IEEE J. Sel. Areas Commun.*, vol. 33, pp. 585–597, Mar. 2015.

[61] O. M. Gul and M. Demirekler, "Average throughput performance of myopic policy in energy harvesting wireless sensor networks," *Sensors*, vol. 17, p. 2206, Sept. 2017.

[62] O. M. Gul and E. Uysal-Biyikoglu, "A randomized scheduling algorithm for energy harvesting wireless sensor networks achieving nearly 100% throughput," in *IEEE WCNC*, (Istanbul, Turkey), pp. 2456–2461, Apr. 2014.

[63] O. M. Gul and E. Uysal-Biyikoglu, "Achieving nearly 100% throughput without feedback in energy harvesting wireless networks," in *IEEE ISIT*, (Honolulu, HI, USA), pp. 1171–1175, June 2014.

[64] O. M. Gul, "Asymptotically optimal scheduling for energy harvesting wireless sensor networks," in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, (Montreal, QC, Canada), pp. 1–7, Oct. 2017.

[65] O. M. Gul and M. Demirekler, "Asymptotically throughput optimal scheduling for energy harvesting wireless sensor networks," *IEEE Access*, vol. 6, pp. 45004–45020, July 2018.

[66] J. Yang and J. Wu, "Online throughput maximization in an energy harvesting multiple access channel with fading," in *IEEE ISIT*, (Hong Kong, China), pp. 2727–2731, Oct. 2015.

[67] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet Things J.*, vol. 6, pp. 2009–2020, Sept. 2018.

[68] S. Luo, H. Zhang, Q. Li, and K. Wu, "Knowledge-assisted DRL for energy harvesting based multi-access wireless communications," in *IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, (Haikou, Hainan, China), pp. 869–876, Dec. 2021.

[69] D. Zhai, H. Chen, Z. Lin, Y. Li, and B. Vucetic, "Accumulate then transmit: Multiuser scheduling in full-duplex wireless-powered IoT systems," *IEEE Internet Things J.*, vol. 5, pp. 2753–2767, Mar. 2018.

[70] S.-M. Park, D.-Y. Kim, K.-W. Kim, and J.-W. Lee, "Joint antenna and device scheduling in full-duplex MIMO wireless powered communication networks," *IEEE Internet Things J.*, Oct. 2022.

[71] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *IEEE INFOCOM*, (Orlando, FL, USA), pp. 2731–2735, May 2012.

[72] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, "A reinforcement learning framework for optimizing age of information in RF-powered communication systems," *IEEE Trans. Commun.*, vol. 68, pp. 4747–4760, May 2020.

[73] S. Leng and A. Yener, "Learning to transmit fresh information in energy harvesting networks using supervised learning," in *55th Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA, USA), pp. 737–741, Oct. 2021.

[74] M. Hatami, M. Leinonen, Z. Chen, N. Pappas, and M. Codreanu, "Asymptotically optimal on-demand AoI minimization in energy harvesting IoT networks," in *IEEE International Symposium on Information Theory (ISIT)*, (Espoo, Finland), pp. 922–927, July 2022.

[75] M. Hatami, M. Leinonen, and M. Codreanu, "Minimizing average on-demand AoI in an IoT network with energy harvesting sensors," in *IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, (Lucca, Italy), pp. 1–5, Sept. 2021.

[76] M. Hatami, M. Leinonen, and M. Codreanu, "AoI minimization in status update control with energy harvesting sensors," *IEEE Trans. Commun.*, vol. 69, pp. 8335–8351, Sept. 2021.

[77] J. Feng, W. Mai, and X. Chen, "Simultaneous multi-sensor scheduling based on double deep Q-learning under multi-constraint," in *IEEE/CIC International Conference on Communications in China (ICCC)*, (Xiamen, China), pp. 224–229, July 2021.

[78] M. Hatami, M. Jahandideh, M. Leinonen, and M. Codreanu, "Age-aware status update control for energy harvesting IoT sensors via reinforcement learning," in *IEEE 31st Annual International Symposium on Personal, Indoor, and Mobile Radio Communications*, (London, UK), pp. 1–6, Aug. 2020.

[79] N. Zhao, C. Xu, S. Zhang, Y. Xie, X. Wang, and H. Sun, "Status update for correlated energy harvesting sensors: A deep reinforcement learning approach," in *IEEE WCSP*, (Nanjing, China), pp. 170–175, Dec. 2020.

[80] L. Liu, X. Qin, X. Xu, H. Li, F. R. Yu, and P. Zhang, "Optimizing information freshness in MEC-assisted status update systems with heterogeneous energy harvesting devices," *IEEE Internet Things J.*, vol. 8, pp. 17057–17070, Dec. 2021.

[81] S. Leng and A. Yener, "An actor-critic reinforcement learning approach to minimum age of information scheduling in energy harvesting networks," in *IEEE ICASSP*, (Toronto, Canada), pp. 8128–8132, June 2021.

[82] L. Cui, Y. Long, D. T. Hoang, and S. Gong, "Hierarchical learning approach for age-of-information minimization in wireless sensor networks," in *IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, (Belfast, United Kingdom), pp. 130–136, June 2022.

[83] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, Mar. 2016.

[84] Q. He, G. Dán, and V. Fodor, "Joint assignment and scheduling for minimizing age of correlated information," *IEEE/ACM Trans. Networking*, vol. 27, pp. 1887–1900, Oct. 2019.

[85] W. Jin, L. Huang, and K. Chi, "Age of information minimization in wireless powered NOMA communication networks," in *IEEE 23rd International*

Conference on High Performance Switching and Routing (HPSR), (Taicang, Jiangsu, China), pp. 201–205, June 2022.

[86] M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "On the role of age of information in the internet of things," *IEEE Commun. Mag.*, vol. 57, pp. 72–77, Dec. 2019.

[87] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[88] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network: Joint active and passive beamforming design," in *IEEE Global Communications Conference (GLOBECOM)*, (Abu Dhabi, United Arab), pp. 1–6, Dec. 2018.

[89] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Commun. Mag.*, vol. 58, pp. 106–112, Jan. 2020.

[90] M. Calvo-Fullana, J. Matamoros, and C. Antón-Haro, "Decentralized sparsity-promoting sensor selection in energy harvesting wireless sensor networks," in *24th European Signal Processing Conference (EUSIPCO)*, (Budapest, Hungary), pp. 582–586, IEEE, Aug. 2016.

[91] P. Du, Q. Yang, Z. Shen, and K. S. Kwak, "Distortion minimization in wireless sensor networks with energy harvesting," *IEEE Commun. Lett.*, vol. 21, pp. 1393–1396, Feb. 2017.

[92] M. Calvo-Fullana, J. Matamoros, and C. Antón-Haro, "Sensor selection and power allocation strategies for energy harvesting wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 34, pp. 3685–3695, Sept. 2016.

[93] M. Calvo-Fullana, J. Matamoros, C. Antón-Haro, and S. M. Fosson, “Sparsity-promoting sensor selection with energy harvesting constraints,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Shanghai, China), pp. 3766–3770, Mar. 2016.

[94] M. Calvo-Fullana, J. Matamoros, and C. Antón-Haro, “Sensor selection in energy harvesting wireless sensor networks,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, (Orlando, FL, USA), pp. 43–47, Dec. 2015.

[95] H. Uzawa, “Iterative methods for concave programming,” *Studies in linear and nonlinear programming*, vol. 6, pp. 154–165, 1958.

[96] E. J. Candes, M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization,” *J.Fourier Anal. Appl.*, vol. 14, pp. 877–905, Oct. 2008.

[97] Y.-B. Chen, I. Nevat, P. Zhang, S. G. Nagarajan, and H.-Y. Wei, “Query-based sensors selection for collaborative wireless sensor networks with stochastic energy harvesting,” *IEEE Internet Things J.*, vol. 6, pp. 3031–3043, Oct. 2018.

[98] A. Hentati, E. Driouch, J.-F. Frigon, and W. Ajib, “Fair and low complexity node selection in energy harvesting wireless sensor networks,” *IEEE Syst. J.*, vol. 12, pp. 3796–3806, Dec. 2017.

[99] A. Hentati, E. Driouch, J.-F. Frigon, and W. Ajib, “Low complexity node selection algorithms in MU-MIMO energy harvesting WSNs,” in *IEEE 84th Vehicular Technology Conference (VTC-Fall)*, (Montreal, QC, Canada), pp. 1–5, Mar. 2016.

[100] P. Zhang, I. Nevat, G. W. Peters, F. Septier, and M. A. Osborne, “Spatial field reconstruction and sensor selection in heterogeneous sensor networks with stochastic energy harvesting,” *IEEE Trans. Signal Process.*, vol. 66, pp. 2245–2257, Feb. 2018.

[101] D. Guo, L. Tang, and X. Zhang, "Joint energy allocation and multiuser scheduling in SWIPT systems with energy harvesting," *IET Communications*, vol. 14, pp. 956–966, Apr. 2020.

[102] R. Y. Rubinstein, "Combinatorial optimization, cross-entropy, ants and rare events," in *Stochastic optimization: algorithms and applications*, pp. 303–363, Springer, 2001.

[103] P. Grover and A. Sahai, "Shannon meets tesla: Wireless information and power transfer," in *IEEE international symposium on information theory*, (Austin, USA), pp. 2363–2367, July 2010.

[104] E. Boshkovska, R. Morsi, D. W. K. Ng, and R. Schober, "Power allocation and scheduling for SWIPT systems with non-linear energy harvesting model," in *IEEE International Conference on Communications (ICC)*, (Kuala Lumpur, Malaysia), pp. 1–6, May 2016.

[105] D. Guo, L. Tang, and X. Zhang, "Optimal energy allocation and multiuser scheduling in SWIPT systems with hybrid power supply," in *IEEE Globecom Workshops (GC Wkshps)*, (Waikoloa, HI, USA), pp. 1–6, Dec. 2019.

[106] R. Morsi, D. S. Michalopoulos, and R. Schober, "Multiuser scheduling schemes for simultaneous wireless information and power transfer over fading channels," *IEEE Trans. Wireless Commun.*, vol. 14, pp. 1967–1982, Dec. 2014.

[107] R. Morsi, D. S. Michalopoulos, and R. Schober, "Multi-user scheduling schemes for simultaneous wireless information and power transfer," in *IEEE International Conference on Communications (ICC)*, (Sydney, NSW, Australia), pp. 4994–4999, June 2014.

[108] M. Chynonova, R. Morsi, D. W. K. Ng, and R. Schober, "Optimal multiuser scheduling schemes for simultaneous wireless information and power transfer,"

in *23rd European Signal Processing Conference (EUSIPCO)*, (Nice, France), pp. 1989–1993, Dec. 2015.

[109] I. Bang, S. M. Kim, and D. K. Sung, "Adaptive multiuser scheduling for simultaneous wireless information and power transfer in a multicell environment," *IEEE Trans. Wireless Commun.*, vol. 16, pp. 7460–7474, Nov. 2017.

[110] Y. Kim, B. C. Jung, I. Bang, and Y. Han, "Adaptive proportional fairness scheduling for swipt-enabled multicell downlink networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, (Marrakesh, Morocco), pp. 1–6, Apr. 2019.

[111] N. Zhao, F. R. Yu, and V. C. Leung, "Opportunistic communications in interference alignment networks with wireless power transfer," *IEEE Wireless Commun.*, vol. 22, pp. 88–95, Feb. 2015.

[112] V. Gupta and S. De, "Adaptive multi-sensing in EH-WSN for smart environment," in *IEEE Global Communications Conference (GLOBECOM)*, (Waikoloa, HI, USA), pp. 1–6, Dec. 2019.

[113] V. Gupta and S. De, "Collaborative multi-sensing in energy harvesting wireless sensor networks," *IEEE Trans. Signal Inf. Process. Networks*, vol. 6, pp. 426–441, June 2020.

[114] R. Song, Q. Wei, and W. Xiao, "ADP-based optimal sensor scheduling for target tracking in energy harvesting wireless sensor networks," *Neural Comput. Appl.*, vol. 27, pp. 1543–1551, June 2016.

[115] F. Liu, C. Jiang, S. Chen, and W. Xiao, "Multi-sensor scheduling for target tracking based on constrained ADP in energy harvesting wsn," in *13th IEEE conference on industrial electronics and applications (ICIEA)*, (Wuhan, China), pp. 1579–1584, May 2018.

[116] F. Liu, W. Xiao, S. Chen, and C. Jiang, "Adaptive dynamic programming-based multi-sensor scheduling for collaborative target tracking in energy harvesting wireless sensor networks," *Sensors*, vol. 18, p. 4090, Nov. 2018.

[117] F. Liu, C. Jiang, and W. Xiao, "Multistep prediction-based adaptive dynamic programming sensor scheduling approach for collaborative target tracking in energy harvesting wireless sensor networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, pp. 693–704, July 2020.

[118] C. Jiang, F. Liu, S. Chen, and W. Xiao, "Finite-horizon adaptive dynamic programming for collaborative target tracking in energy harvesting wireless sensor networks," in *Chinese Control And Decision Conference (CCDC)*, (Nanjing, China), pp. 4731–4736, June 2019.

[119] P. J. Werbos, W. Miller, and R. Sutton, "A menu of designs for reinforcement learning over time," in *Neural networks for control*, vol. 3, pp. 67–95, MIT press Cambridge, MA, 1990.

[120] H. Tabassum, E. Hossain, M. J. Hossain, and D. I. Kim, "On the spectral efficiency of multiuser scheduling in RF-powered uplink cellular networks," *IEEE Trans. Wireless Commun.*, vol. 14, pp. 3586–3600, Mar. 2015.

[121] M. Dimitropoulou, C. Psomas, and I. Krikidis, "k-th best device selection for scheduling in wireless powered communication networks," in *IEEE International Conference on Communications (ICC)*, (Dublin, Ireland), pp. 1–6, June 2020.

[122] M. Dimitropoulou, C. Psomas, and I. Krikidis, "Generalized selection in wireless powered networks with non-linear energy harvesting," *IEEE Trans. Commun.*, vol. 69, pp. 5634–5648, May 2021.

[123] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in multiuser OFDM systems with wireless information and power transfer," in

IEEE Wireless communications and networking conference (WCNC), (Shanghai, China), pp. 3823–3828, July 2013.

[124] H. Ko, H. Lee, T. Kim, and S. Pack, "Information freshness-guaranteed and energy-efficient data generation control system in energy harvesting internet of things," *IEEE Access*, vol. 8, pp. 168711–168720, Sept. 2020.

[125] J. Yang, X. Wu, and J. Wu, "Optimal scheduling of collaborative sensing in energy harvesting sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 33, pp. 512–523, Mar. 2015.

[126] Y. Li and K.-W. Chin, "Random channel access protocols for SIC enabled energy harvesting IoTs networks," *IEEE Systems*, vol. 15, pp. 2269–2280, June 2020.

[127] Powercast, "P2110B 915 MHz RF powerharvester® receiver." https://www.powercastco.com/wp-content/uploads/2016/12/P2110B-Datasheet-Rev-3.pdf/, 2016. [Online].

[128] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Ann. Oper. Res.*, vol. 134, pp. 19–67, Feb. 2005.

[129] X. Li, X. Tang, C.-C. Wang, and X. Lin, "Gibbs-sampling-based optimization for the deployment of small cells in 3G heterogeneous networks," in *IEEE WiOpt*, (Tsukuba, Japan), pp. 444–451, May 2013.

[130] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *Eur. J. Oper. Res.*, vol. 99, pp. 89–112, May 1997.

[131] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodol. Comput. Appl. Probab.*, vol. 1, pp. 127–190, Sept. 1999.

[132] Y. Qian, W. B. Haskell, A. X. Jiang, and M. Tambe, "Online planning for optimal protector strategies in resource conservation games," in *AAMAS*, (Paris, France), pp. 733–740, May 2014.

[133] Powercast, "P2110-EVB evaluation board for P2110 powerharvester® receiver." https://www.powercastco.com/wp-content/uploads/2016/11/p2110-evb1.pdf/, 2015. [Online].

[134] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Trans. Commun.*, vol. 67, pp. 7468–7482, July 2019.

[135] E. Boshkovska, D. W. K. Ng, N. Zlatanov, A. Koelpin, and R. Schober, "Robust resource allocation for MIMO wireless powered communication networks based on a non-linear EH model," *IEEE Trans. Commun.*, vol. 65, pp. 1984–1999, Feb. 2017.

[136] P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams, "Finite state Markov modeling of fading channels-a survey of principles and applications," *IEEE Sig. Proc. Mag.*, vol. 25, pp. 57–80, Sept. 2008.

[137] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, May 1992.

[138] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, and G. Andrieux, "Energy consumption modeling for communicating sensors using LoRa technology," in *IEEE Conference on Antenna Measurements & Applications (CAMA)*, (Sweden), pp. 1–4, Sept. 2018.

[139] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Trans. Networking*, vol. 28, pp. 2215–2228, Oct. 2020.

[140] J. R. Norris, *Markov chains*. Cambridge university press, 1998.

[141] K. S. Narendra and M. A. Thathachar, “Learning automata-a survey,” *IEEE Trans. Syst. Man Cybern.*, pp. 323–334, July 1974.

[142] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, “Energy-efficient orchestration in wireless powered internet of things infrastructures,” *IEEE Trans. Green Commun. Networking*, vol. 3, pp. 317–328, June 2018.

[143] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, and G. Andrieux, “Energy consumption modeling for communicating sensors using LoRa technology,” in *IEEE CAMA*, pp. 1–4, Sept. 2018.

[144] K. W. Choi and D. I. Kim, “Stochastic optimal control for wireless powered communication networks,” *IEEE Trans. Wireless Commun.*, vol. 15, pp. 686–698, Sept. 2015.