Spectrum Efficiency Enhancement in Wireless Communication Networks

by

Ziling Wei

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Communications

Department of Electrical and Computer Engineering
University of Alberta

Abstract

With the fast growth of mobile data traffic, spectrum scarcity has become a serious problem to the development of wireless networks. Due to the limited available spectrum resources, it is critical to improve the spectrum efficiency. Cognitive radio, opportunistic scheduling, and non-orthogonal multiple access (NOMA) are promising techniques which can largely improve the spectrum efficiency in wireless communication networks. However, some challenges exist in deploying them in practical wireless networks. In this thesis, we aim at quality-of-service provisioning of networks by solving these challenges, with four research components.

The first research component focuses on the optimal slot length configuration in cognitive radio networks. A slot length configuration scheme with imperfect spectrum sensing is proposed in this research. In the proposed scheme, the spectrum sensing result is considered when configuring the slot length. Then, an optimization problem to find out the optimal slot length configuration is formulated and analyzed. And an algorithm is proposed to solve the problem.

Then, the opportunistic scheduling in wireless networks is considered in the next two research components. First, considering the limitations of existing centralized opportunistic scheduling schemes, the opportunistic scheduling problem is modeled as a semi-Markov decision process (SMDP) which reduces the implementation complexity. Then, a model-based scheduling method and a model-free scheduling method are proposed to derive the optimal scheduling policy for fully explored networks and partially explored networks, respectively. Second, the problem of distributed opportunistic channel access with energy-harvesting relays is investigated. A distributed opportunistic scheduling (DOS) scheme is proposed. To maximize the average throughput of the network, an optimal stopping strategy with threshold-based structure is derived in this scheme. To obtain the threshold, a low complexity algorithm is proposed to derive the stationary probability distribution of the energy level of each relay, and then, the threshold can be calculated off-line by a proposed iterative algorithm.

Last but not least, NOMA power allocation is investigated for an Internet of Things (IoT) device to offload its computation tasks to a fog computing system. An optimization problem to maximize the long-term average system utility is formulated by optimizing the IoT device's power allocation and task allocation. An algorithm with polynomial time complexity is proposed to solve the problem.

Preface

This thesis is an original work by Ziling Wei. Parts of the thesis have been published or submitted to journals, which are indicated below.

The work of Chapter 3 is submitted to IEEE Access in October 2018 as "Z. Wei and H. Jiang, 'Optimal Slot Length Configuration in Cognitive Radio Networks,' ".

The work of Chapter 6 is published as "Z. Wei and H. Jiang, 'Optimal Offloading in Fog Computing Systems With Non-Orthogonal Multiple Access,' in *IEEE Access*, vol. 6, pp. 49767-49778, September 2018".

Acknowledgements

First I would like to express my deepest gratitude to my supervisor, Dr. Hai Jiang, for his invaluable insight on my research and his continuous support during my Ph.D. journey. His inspiration and rigorous scholarship encouraged me on the road of academic research. He provided me a lot of guidance both the topic selection and the topic research. Many ideas came out during our discussions. It is truly a fortune for me to have Dr. Hai Jiang as my supervisor, and his research attitude will influence me in my future career life.

I would like to sincerely thank Dr. Hao Liang, Dr. Majid Khabbazian, Dr. Zhijun Qiu, and Dr. Lin Cai for their time and efforts in reviewing my thesis and providing their inspiring suggestions to help me improve the quality of my thesis. I am also grateful to Dr. Pedram Mousavi for his insightful suggestions during my candidacy exam.

Also, I wish to thank all professors who delivered courses during my Ph.D. program.

I am especially thankful to current and former colleagues of Advanced Wireless Communications and Signal Processing Lab W5-077 for their technical discussions and continual help. I also want to thank all my friends at the University of Alberta.

Then, I sincerely thank my parents, grandparents, my sister, and especially my fiancee for their endless love and unconditional support. They helped me a lot in various parts of my life.

Last but not the least, I appreciate the financial support from the China Scholarship Council during my Ph.D. program.

Contents

List of Tables

List of Figures

List of Symbols

Symbols in Chapter 3

$f_{\mathrm{i}}(x)$	Probability density function of the sojourn time of idle state
$f_{\mathrm{b}}(x)$	Probability density function of the sojourn time of busy state
P_{d}	Probability of detection
P_{f}	Probability of false alarm
γ_{p}	Received SNR of primary signals
T_{s}	Spectrum sensing time
f_{s}	Sampling frequency of spectrum sensing
T_{i}	Slot length of idle state
T_{b}	Slot length of busy state
η_{c}	Collision ratio
η_{s}	Sensing ratio
T_{slot}	Expected duration of a slot
$T_{\mathrm{b-in-slot}}$	Expected channel busy duration in a slot
$T_{\mathrm{i-in-slot}}$	Expected channel idle duration in a slot
$T_{\mathrm{c-in-slot}}$	Expected duration of collision in a slot
$T_{\mathrm{w-in-slot}}$	Expected duration of wasting in a slot
R_{slot}	Throughput of secondary transmissions

Symbols in Chapter 4

N	Number of users
K	Number of channels
i	Index of users
D_i	Size of user i's task buffer

b_i	Priority index of user i's task
d_i	Backlog of user i's task buffer
k_i	Number of channels occupied by user i
e	Type of an event
l	Index of the user with an event
t	Decision epoch
s_n	State at nth decision epoch
a_n	Applied action at nth decision epoch
π	Scheduling policy
$p(s'\|s,a)$	Transition probability
$r(s,a)$	Reward function of state-action pair (s,a)

Symbols in Chapter 5

I	Number of source-destination pairs
p_{c}	Probability of channel access contention
L	Number of levels for a battery
$B_{\max}$	Capacity of each relay's battery
P_{s}	Transmission power of sources
h_{ij}	Channel gain between source i and relay j
g_{ji}	Channel gain between relay j and destination i
τ_{tx}	Channel coherence time
τ_{c}	Duration of an RTS/CTS packet's transmission
t	Index of cycles
$l_i(t)$	Residual energy level of relay i at the beginning of the tth cycle
σ	Amount of energy for each battery level
N	Stopping rule
P_i^{r}	Transmission power of relay i
β	Energy conversion efficiency
λ	Average network throughput

Symbols in Chapter 6

N	Number of fog nodes
T	Duration of a slot
t	Index of a time slot
i	Index of fog nodes
$D_{\max}(t)$	Amount of data generated by the IoT device at the tth time slot
$D(t)$	Portion of the data to be offloaded to fog nodes at the tth time slot
$Q(t)$	Amount of data in the IoT device's task buffer at the tth time slot
$R_i(t)$	Amount of data that are delivered to fog node i at the tth time slot
$C_i(t)$	Occupancy of fog node i's task buffer at the tth time slot
$L_i(t)$	Amount of data that can be computed by fog node i at the tth time slot
$r_i(t)$	Transmission rate between the IoT device and fog node i at the tth time slot
$f_i(t)$	Service rate of fog node i at the tth time slot
$p_i(t)$	Power allocated to the data offloading to fog node i at the tth time slot

Glossary of Terms

Acronyms	**Definition**
QoS	quality of service
OMA	orthogonal multiple access
FDMA	frequency-division multiple access
TDMA	time-division multiple access
OFDMA	orthogonal frequency-division multiple access
NOMA	non-orthogonal multiple access
SNR	signal-to-noise ratio
DOS	distributed opportunistic scheduling
CSI	channel state information
AF	amplify and forward
DF	decode and forward
SIC	successive interference cancellation
AWGN	additive white Gaussian noise
MDP	Markov decision process
SMDP	semi-Markov decision process
CTMDP	continuous time Markov decision process
DTMDP	discrete time Markov decision process
i.i.d.	independent and identically distributed
PDF	probability distribution function
CDF	cumulative distribution function
IoT	internet of things
RF	radio frequency

Chapter 1

Introduction

1.1 Spectrum Efficiency

With the rapid development of wireless communication technology in the past decades, the wireless data traffic has experienced a dramatic growth. As shown in Fig. 1.1, it is predicted that the mobile traffic data volume in 2021 would be 49 exabytes per month [1]. This growth is major driven by increasing mobile devices (e.g., smart phones, tablets, laptops, and so on). The dramatic increasing of data traffic has largely promoted to establish the *smart city* [3], but also resulted in shortage of communication resources (e.g., spectrum resources). Accordingly, the available radio spectrum is becoming scarce. Spectrum scarcity is now a serious problem to the development of future wireless networks [4]. To meet the quality of service (QoS) for the ever-increasing mobile data traffic, one possible solution is to explore new spectrum. However, almost all appropriate spectrum (i.e., the spectrum under 6 GHz) has been allocated to various wireless services[1]. Accordingly, to alleviate the spectrum scarcity problem, improving the spectrum efficiency[2] is a key and promising solution [5]. It has attracted a lot of research effort, e.g., more than 100 projects on increasing spectrum efficiency are funded by the Enhancing Access to the Radio Spectrum (EARS) program [6]. In wireless communication systems, there are two main challenges that affect spectrum efficiency.

Firstly, since most of the appropriate radio spectrum has been allocated to var-

[1]Although millimeter wave (mmWave) frequencies (30-300GHz) are still available, their coverage may be limited, because 1) the propagation loss in mmWave is high, 2) a mmWave link is highly sensitive to blockage, and 3) mmWave does not work well in a mobile environment.

[2]In a communication system, spectrum efficiency is used to describe the data transmission rate over a given bandwidth channel.

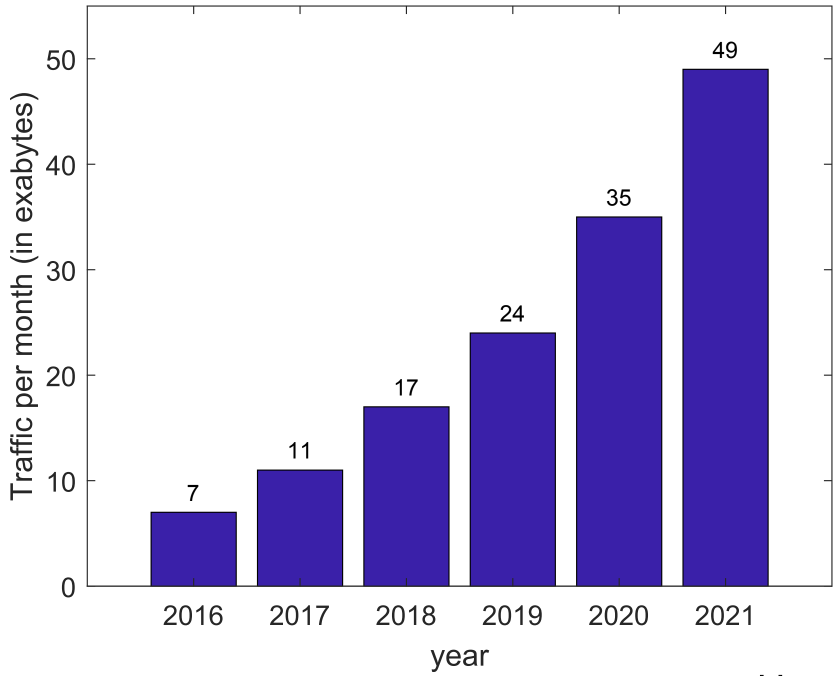

Figure 1.1: Global mobile data traffic from 2016 to 2021 [1].

ious wireless services, only licensed users have the permission to access the specific spectrum bands (also called channels). It has been shown that the licensed channels experience low utilization, e.g., a large portion of allocated spectrum is actually not utilized by the licensed users at any particular time [7]. Therefore, the licensed spectrum under-utilization is one of the significant factors which leads to the low spectrum efficiency problem.

Secondly, a wireless communication system usually needs to support a number of users. If multiple users in a small area start their transmissions simultaneously over the same available channel, large interference between them may be caused. In this case, the performance of each user is largely degraded. Hence, orthogonal multiple access (OMA) technologies over wireless channels, such as frequency-division multiple access (FDMA), time-division multiple access (TDMA) [8], and orthogonal frequency-division multiple access (OFDMA) [9], have been proposed. In OMA technologies, each wireless resource block (i.e., a frequency band in FDMA, a time slot in TDMA, or some subcarriers in OFDMA) is assigned to only one user. A main issue with OMA techniques is that the spectrum efficiency may be low due to the fluctuation in channel conditions for different users, i.e., some resource blocks are allocated to users with poor channel conditions.

To improve the spectrum efficiency, some techniques have been proposed (e.g., cognitive radio [10], opportunistic scheduling [11], non-orthogonal multiple access (NOMA) [12], Massive MIMO [13], ultra wideband techniques [14], millimeter wave [15], etc.). In this thesis, we focus on the techniques, including cognitive radio, opportunistic scheduling, and NOMA. Cognitive radio has been introduced to solve the first challenge, while opportunistic scheduling and NOMA are two promising solutions to the second challenge.

1.2 Cognitive Radio

To address the licensed spectrum under-utilization problem, cognitive radio is introduced to wireless networks (referred to as cognitive radio networks) [16]. In cognitive radio networks, the licensed users of a licensed channel are called primary users, while the users that do not have the license to access the channel are called secondary users. Since the licensed channels experience low utilization, spectrum holes (the durations within which the licensed channel is not utilized by primary users) occur frequently, which is described in Fig. 1.2. Accordingly, spectrum efficiency is largely enhanced by letting secondary users access the spectrum holes. Note that the primary users have priority to access the licensed channel in cognitive radio networks. It means that secondary users are required not to affect the transmission of primary users. Accordingly, two methods are proposed for secondary users to share the licensed channels, namely underlay method and overlay method [10].

In the underlay method, secondary users can access the licensed channel at any time (even if the primary users are active). To meet the QoS requirement of primary users, the interference to the primary receiver should be below a tolerable threshold. Accordingly, the transmission power of secondary users should be limited during the transmission. On the contrary, in the overlay method, a secondary user can access a licensed channel only when primary users of the channel are not transmitting [17]. In the overlay method, the QoS requirement of primary users can be fully satisfied. Further, the above transmission power constraint, which is necessary for the underlay method, is not needed in the overlay method. Hence, the overlay method has attracted

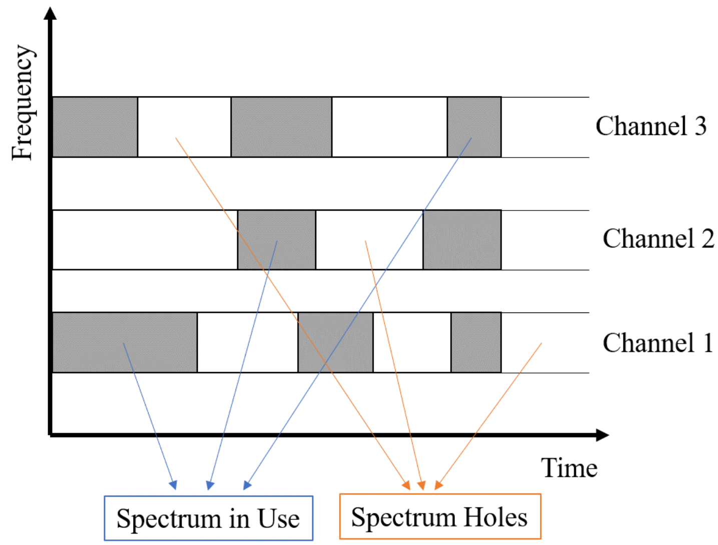

Figure 1.2: Spectrum holes in licensed channels.

a lot of research attention and it is adopted in our works. In order to detect the spectrum holes, the secondary users are required to monitor primary users' activities in the overlay method. It is performed by spectrum sensing, which is designed to detect whether primary signals exist or not.

1.2.1 Spectrum Sensing

Spectrum sensing is important to protect the QoS of primary transmissions[3] and detect as many spectrum holes as possible. Thus, the detection accuracy of spectrum sensing is crucial to the performance of cognitive radio networks. Two metrics, probability of false alarm (denoted as P_{f}) and probability of detection (denoted as P_{d}), are introduced to measure the detection performance [18]. Let busy denote the state that the licensed channel is occupied by primary users, while idle is denoted as the state that no primary signal exists in the licensed channel. Accordingly, the probability of false alarm P_{f} denotes the probability that the sensing result is busy given that no primary signal actually exists in the channel. Similarly, the probability of detection P_{d} denotes the probability that the sensing result is busy given that primary user are indeed transmitting over the channel. In cognitive radio networks, to protect the primary transmissions and improve the spectrum utilization, spectrum sensing is

[3]A primary transmission means a data transmission which is performed by a primary user. Similarly, a secondary transmission means a data transmission which is performed by a secondary user.

usually designed to minimize the probability of false alarm under a constraint on the probability of detection.

There are several spectrum sensing techniques, e.g., matched filter detection, cyclostationary feature detection, energy detection, and so on [19].

- Matched filter detection: Matched filter is a kind of coherent pilot sensor, which can be used to detect a known signal by maximizing the signal-to-noise ratio (SNR). Accordingly, if a prior knowledge about the primary user signal (e.g., a pilot sequence) is known by secondary users, the presence of the primary user signal in the received signal can be detected [20]. However, in most of cognitive radio networks, the information of primary user signal is generally hard to obtain for secondary users [21].

- Cyclostationary feature detection: Cyclostationary feature detection deals with cyclostationary features that might exist in primary user signals. Note that these features have a periodic statistics and spectral correlation that do not exist in interference signal or noises [22]. Accordingly, the primary user signal can be detected by exploiting the received signal periodicity. Under low SNR regions, the cyclostationary feature detection method can still obtain a good detection performance. However, this method has a large computation complexity. In addition, compared to other methods, significantly longer sensing time is needed for the cyclostationary feature detection method [23].

- Energy detection: In energy detection, the primary user signal is detected by a comparison of an energy detector's output (i.e., the energy level of the received signal) with a threshold ε [24]. If the output is larger than the threshold, the state of the licensed channel is estimated as busy. Otherwise, the state of the licensed channel is estimated as busy. In this method, little information of primary user signal is required. Moreover, it is easy to implement in reality.

Due to its low complexity, energy detection attracts the most attention, and thus, it is adopted in our work if spectrum sensing is needed.

1.2.2 Energy Detection

With energy detection, the received samples of a secondary user follow a binary hypothesis:

$$H_0 : y(i) = u(i), i = 1, 2, \cdots, f_\mathrm{s} T_\mathrm{s} \tag{1.1}$$

$$H_1 : y(i) = s(i) + u(i), i = 1, 2, \cdots, f_\mathrm{s} T_\mathrm{s} \tag{1.2}$$

where H_0 and H_1 mean that the channel state is idle and busy respectively, T_s denotes the sensing time, f_s is the sampling rate of the received signal, $y(\cdot)$ represents the signal that is received by the secondary user, $s(\cdot)$ is the primary user signal in the channel which is received by the secondary user, and $u(\cdot)$ is the background noise of the channel. Thus, the test statistic of the secondary user is given as

$$Y = \frac{1}{f_\mathrm{s} T_\mathrm{s}} \sum\nolimits_{i=1}^{f_\mathrm{s} T_\mathrm{s}} \left| y\left(i \right) \right|^2. \tag{1.3}$$

Then, given the threshold ε, the probability of detection and the probability of false alarm can be derived by [18]

$$P_\mathrm{d} = \Pr \left(Y \geq \varepsilon \left| H_1 \right. \right) \tag{1.4}$$

$$P_\mathrm{f} = \Pr \left(Y \geq \varepsilon \left| H_0 \right. \right) \tag{1.5}$$

where $\Pr(\cdot)$ means probability.

1.3 Opportunistic Scheduling

In current wireless communication systems, an available channel usually needs to support a number of users. To reduce the interference and guarantee the QoS of each transmission, the available channel is assigned to only one user for a transmission at each time duration (i.e., OMA). However, when the user is with poor channel condition, it can only achieve a low information rate by utilizing this channel. To solve this problem, *opportunistic scheduling*, which could enhance the throughput of the network by allocating the channel access opportunities to the users with good channel condition, is introduced [25]. Thus, a user needs to give up its channel access opportunity when its channel condition is poor. On the other hand, it will get channel access opportunities and achieve a large throughput when its channel condition is better than those of other users. By opportunistic scheduling, in a long term, all users in the network can benefit, and the overall spectrum efficiency is largely improved. Opportunistic scheduling has been widely adopted in existing wireless networks. For example, in the standard IEEE 802.11k, the access point determines the allocations of network resources [26]. There are two kinds of opportunistic scheduling in wireless networks, centralized opportunistic scheduling and distributed opportunistic scheduling (DOS).

1.3.1 Centralized Opportunistic Scheduling

A classic opportunistic scheduling scheme is given in Fig. 1.3. In this kind of scheme, a central entity (e.g., base station (BS)) is needed. At each slot, the information (e.g., channel state information (CSI), energy information, etc.) of each user can be collected by the central entity. Then, it makes a scheduling decision which means selecting a user to utilize the available channel. In general, the user with the best channel condition is selected by the central entity, and thus, the channel access opportunity is allocated to this user. In this case, centralized opportunistic scheduling has been well studied. For example, in [27], an adaptive centralized opportunistic scheduling scheme is proposed to maximize the profits of the network under the constraint of each user's QoS requirement. The centralized opportunistic scheduling with imperfect CSI is investigated in [28].

However, the communication overhead to obtain the CSI of all users may be intolerable in some scenarios [29]. In addition, in distributed wireless networks (e.g., *ad hoc* networks), there is no central entity, and thus, it is hard for a user to make an appropriate decision whether to give up the channel access opportunity. To address these problems, DOS is introduced in wireless networks.

1.3.2 Distributed Opportunistic Scheduling

A classic DOS scheme is given in Fig. 1.4. In a DOS scheme, users contend for the channel access opportunity, and a user with a successful contention decides whether to transmit data or give up the transmission opportunity according to its information

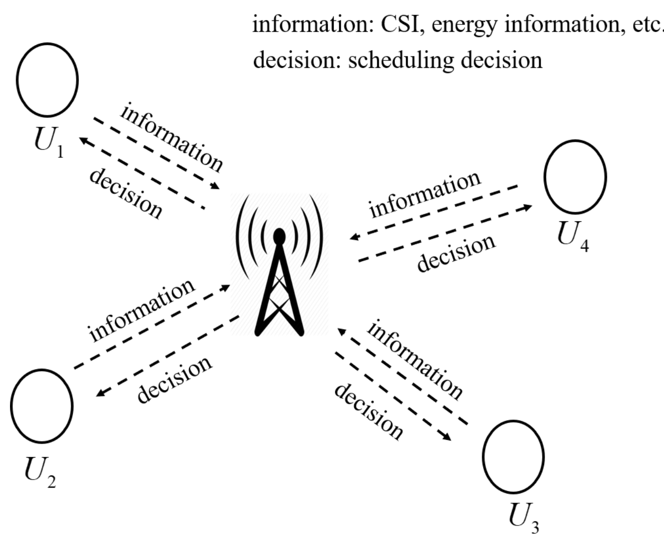

Figure 1.3: A classic centralized opportunistic scheduling scheme.

[30]. To enhance the throughput of the network, the user with a successful contention should utilize the channel for a transmission if its channel condition is good enough (i.e., larger than a pre-defined threshold). Otherwise, the user should give up the channel access opportunity. If the threshold takes a large value, a lot of time would be wasted on exploring a suitable channel, and thus, lots of channel access opportunities are given up. On the other hand, if the threshold takes a small value, a channel access opportunity may be assigned to a user with poor channel conditions. Accordingly, how to select the threshold is a challenging and meaningful problem in DOS.

Since cooperative communications have emerged as a promising technique to enhance communication efficiency, DOS in cooperative networks has received much attention [31]- [32]. In cooperative communication, one or more relay nodes help the source to forward data to the destination, and thus, the negative effect of channel fading can be greatly reduced [33]. There are two major realization schemes in cooperative communication, amplify and forward (AF) and decode and forward (DF). In the AF relay scheme, the relay node amplifies its received signal, and then forwards the amplified signal to the destination without decoding the received signal [34]. In the DF relay scheme, the relay node first decodes its received signal, and then forwards to the destination a re-encoded version of the signal[4] [35]. Compared to the DF scheme, the AF scheme is simpler to implement, but suffers from the noise

[4]It may use the same code-book with the source or an independent code-book.

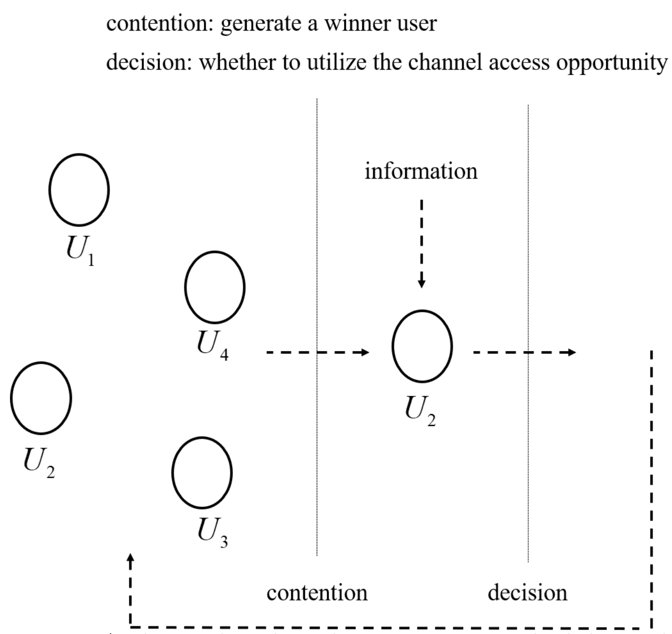

Figure 1.4: A classic distributed opportunistic scheduling scheme.

amplification problem. On the other hand, the DF scheme may suffer from the error propagation problem if the relay node incorrectly decodes the received signal [36].

In a cooperative transmission, there are generally two links, source-to-relay link and relay-to-destination link. Accordingly, to enhance the throughput of the cooperative network by using DOS, the user with a successful contention needs to takes both of the two links into consideration for deciding whether to give up the transmission opportunity.

1.4 Non-Orthogonal Multiple Access

For the conventional OMA techniques, only one user can be served by one resource block. The spectrum efficiency is low when the resource blocks are allocated to users with poor channel conditions. Similar to opportunistic scheduling, NOMA is another technique which can largely improve the spectrum efficiency in wireless networks [12]. It is another promising multiple access different from OMA. In NOMA, a single resource block can simultaneously serve multiple users, which is given in Fig. 1.5. NOMA has been proposed in 3GPP LTE Release 13 [37], and also included in various White Papers on 5G, such as the White Paper from ZTE Corporation, SK Telecom, an so on [38]. There are different NOMA solutions, code-domain NOMA and power-domain NOMA. In code-domain NOMA, multiple users can transmit simultaneously over the same frequency band by using different sequences that are sparse or with low cross-correlation [39]. However, in power-domain NOMA, different transmissions can share a wireless resource block, but adopt different power levels. Then, the receiver can distinguish the signal form different transmissions according to successive interference cancellation (SIC) [40]. Due to its low implementation complexity, power-domain NOMA attracts much more attention, and we also focus on power-domain NOMA in this thesis. Accordingly, in the sequel, "NOMA" means "power-domain NOMA".

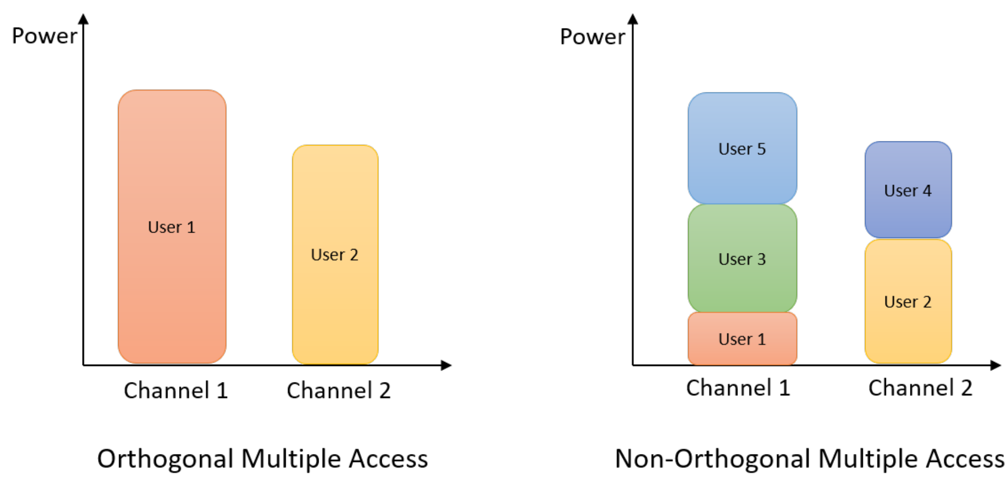

Figure 1.5: OMA vs NOMA [2].

1.4.1 Successive Interference Cancellation

By SIC, the signals of different users are decoded sequentially [41]. The decoding order (i.e., which signal is decoded first and which signal is next?) is usually decided by the signal strength, which decodes the strong signal first and then to decode the weak signal. To decode a user's signal, other users' signals, which are not yet decoded, are treated as interference. Then, when the user's signal is successfully decoded, the signal can be cancelled out from the received mixture of multiple users' signals, which benefits subsequent decoding of other users' signals. In the following, we give an example to demonstrate the principle of SIC.

Considering a downlink transmission with one transmitter (denoted A) and two receivers (denoted B_1 and B_2), the channel gains from A to B_1 and B_2 are denoted

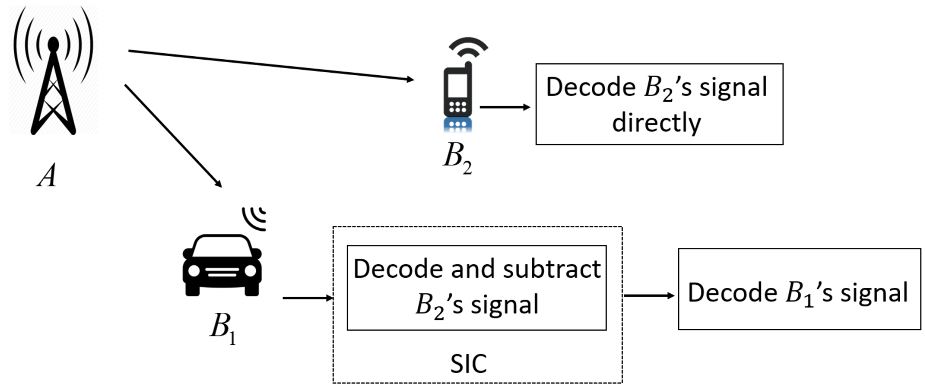

Figure 1.6: An example of downlink NOMA scheme.

as h_1 and h_2, respectively. It is given in Fig. 1.6. The channel condition from A to B_1 is assumed better than the channel condition from A to B_2, and thus, we have $|h_1| > |h_2|$. In power-domain NOMA, different power levels are allocated to different transmissions, i.e., $P_1 < P_2$, where P_1 and P_2 are the transmit power to B_1's signal and B_2's signal, respectively. Therefore, for B_1, the signal of B_2 is a stronger signal compared to its own signal, and thus, the detailed coding scheme is given as follows.

- The transmit signal is superposition coded signal of B_1 and B_2.
- For B_1, the signal of B_2 can be decoded first and subtracted from the received signal by SIC, and then B_1's signal can be detected.
- For B_2, it treats B_1's signal as interference and decodes itself data directly from the received signal.

1.5 Thesis Motivations and Contributions

Cognitive radio, opportunistic scheduling, and NOMA are promising techniques which can largely improve the spectrum efficiency in wireless networks. However, some challenges exist in deploying them in practical wireless networks. In this thesis, we aim to QoS provisioning of networks by solving these challenges.

1.5.1 Optimal Slot Length Configuration in Cognitive Radio Networks

In cognitive radio networks, a slotted time structure is popularly used. Time is partitioned into fixed-length slots. Spectrum sensing is used to detect possible primary signal at the beginning of each slot. If no primary signal is detected by spectrum sensing, secondary users can transmit over the channel during the remaining time of the slot. On the other hand, if primary signals are detected, secondary users should keep silent at this slot. In most of existing works, it is assumed that the channel state (idle or busy) does not change within a slot of secondary users. This assumption is actually not reasonable in practice. Primary transmission is a stochastic event, and thus, it may start at any moment within a slot of secondary users. However, the spectrum sensing operation is only taken at the beginning of a slot. Therefore, a collision between the primary transmission and secondary user may occur. Taking this into account, a continuous Markov model is proposed in [42], in which the state of a licensed channel alternates between "idle" and "busy" based on a continuous Markov model. Considering this model, the slot length of secondary users is a factor that largely affects the performance of cognitive radio networks. A smaller slot length can lead to shorter collision time[5], and thus, a higher QoS level for primary transmissions is provided. However, a higher frequency to perform the spectrum sensing is also required. This means that secondary users need to cost more time and energy for spectrum sensing, and thus, less time is left for secondary transmissions. Accordingly, there is a tradeoff between the QoS of primary users and the reward of secondary users by setting the slot length.

In this research, a slot length configuration scheme with imperfect spectrum sensing is proposed in Chapter 3. In the proposed scheme, the spectrum sensing result is jointly considered when configuring the slot length. Then, an optimization problem to find out the optimal slot length configuration is formulated and analyzed. And an algorithm is proposed to solve the problem.

1.5.2 Optimal Opportunistic Scheduling in Wireless Networks

- Event-Driven Centralized Opportunistic Scheduling in Wireless Networks

[5]Collision time means the duration that a primary transmission is interfered with by secondary transmissions.

In traditional centralized opportunistic scheduling schemes, the CSI of different links is needed. However, the communication overhead to get the CSI may be intolerable. In addition, it is not feasible to obtain the CSI in some scenarios. The implementation complexity is another challenge to take the traditional opportunistic scheduling schemes. In these schemes, time is divided to equal length slots[6], and then, the scheduling action has to be taken at each time slot. In this case, the implementation complexity is pretty large in practice.

Considering the limitations of existing centralized scheduling schemes, a new kind of opportunistic scheduling scheme is proposed in Chapter 4. We formulate the opportunistic scheduling problem as a semi-Markov decision process (SMDP), where the scheduling action is driven by events (i.e., task arrival event and task transmission completion event). Then, we propose a model-based scheduling method and a model-free scheduling method for fully explored networks and partially explored networks, respectively.

- DOS in Cooperative Networks with Energy Harvesting

  Since cooperative communications have emerged as a promising technique to enhance spectrum efficiency, DOS in cooperative networks receives more and more attention. In reality, relay nodes are usually battery limited [43], and thus, periodic replacement or recharging for the battery of relay nodes is needed. However, it may not be feasible in lots of scenarios [44]. As a promising solution to encourage the energy-constrained relays to provide cooperative services, energy harvesting technique has attracted much attention [45]. By collecting energy from ambient sources (e.g., solar, wind, and radio-frequency (RF) signal), relay nodes can forward the received data to destinations without external energy [46]. Compared to other sources, RF signal is a kind of predictable and controllable source. Thus, energy harvesting from RF signals is widely used in wireless networks, especially in cooperative networks [47]. Although wireless cooperative networks with energy harvesting attract more and more attention, there are still no efforts on designing optimal DOS schemes in wireless cooperative networks with RF energy harvesting.

[6] The length of a time slot is generally needed to smaller than the channel coherence time.

In this research, the problem of distributed opportunistic channel access in energy-limited wireless cooperative networks is investigated in Chapter 5. To cope with the energy limitation problem, RF energy harvesting relays are considered, and thus, no external energy is needed for relay nodes. Then, a DOS scheme is proposed. To maximize the average throughput of the network, an optimal threshold-based strategy of the proposed scheme is derived by optimal stopping theory. To obtain the threshold, a low complexity algorithm is proposed to derive the stationary probability distribution of the energy level of each relay, and then, the threshold can be calculated off-line by a proposed iterative algorithm.

1.5.3 Optimal Power Allocation in Fog Computing System with NOMA

Since multiple users in NOMA are distinguished by power levels, one of the most important methods to achieve the benefits of NOMA should be power allocation. The optimal power allocation in cellular networks with NOMA has been widely investigated. As a promising wireless technique to improve the spectrum efficiency, NOMA has also been shown important to the evolution of many types of applications or networks, e.g., vehicular *ad hoc* networks, digital TV broadcasting, terrestrial-satellite networks, fog computing, etc. [48]. Different from traditional cellular networks, other networks with NOMA have some specific properties. Thus, NOMA power allocation schemes in traditional cellular networks cannot be directly adopted in other networks. For example, in fog computing, the computing capacity of fog nodes and the computing latency of tasks need to be jointly considered when configuring NOMA power allocation.

Fog computing, which provides computing capabilities at the network edge, is a promising solution to satisfy the rapid growth of computation demands by mobile devices. By offloading computation tasks of the Internet of things (IoT) devices to fog nodes, better computation experience (for example, lower execution latency) can be achieved.

In this research, an optimal offloading scheme with NOMA is proposed in Chapter 6. To improve the offloading efficiency, downlink non-orthogonal multiple access (NOMA) is applied in fog computing systems such that the IoT device can perform simultaneous offloading to multiple fog nodes. However, due to the limited computation resources of fog nodes and limited wireless resources, designing an efficient offloading scheme, which allocates the amount of data to each fog node by task allocation and power allocation, is important for fog computing systems. Thus, to maximize the long-term average system utility, a task and power allocation problem for computation offloading is formulated, subject to task delay and energy costing constraints. By Lyapunov optimization method, the original problem is transformed to an online optimization problem in each time slot, which is non-convex. Accordingly, we propose an algorithm to solve the non-convex online optimization problem with polynomial complexity.

1.6 Thesis Outline

The thesis is organized as follows. In Chapter 2, the background information is introduced. In cognitive radio networks, optimal slot length configuration is considered in Chapter 3. In Chapter 4, a centralized opportunistic scheduling scheme, which is named as SDMP-based event-driven opportunistic scheduling scheme, is proposed. In Chapter 5, the distributed opportunistic scheduling is discussed in energy-limited cooperative networks. Chapter 6 presents the research results on power allocation in fog computing with NOMA. Chapter 7 concludes the thesis and gives future research directions. The structure of this thesis is given in Fig. 1.7.

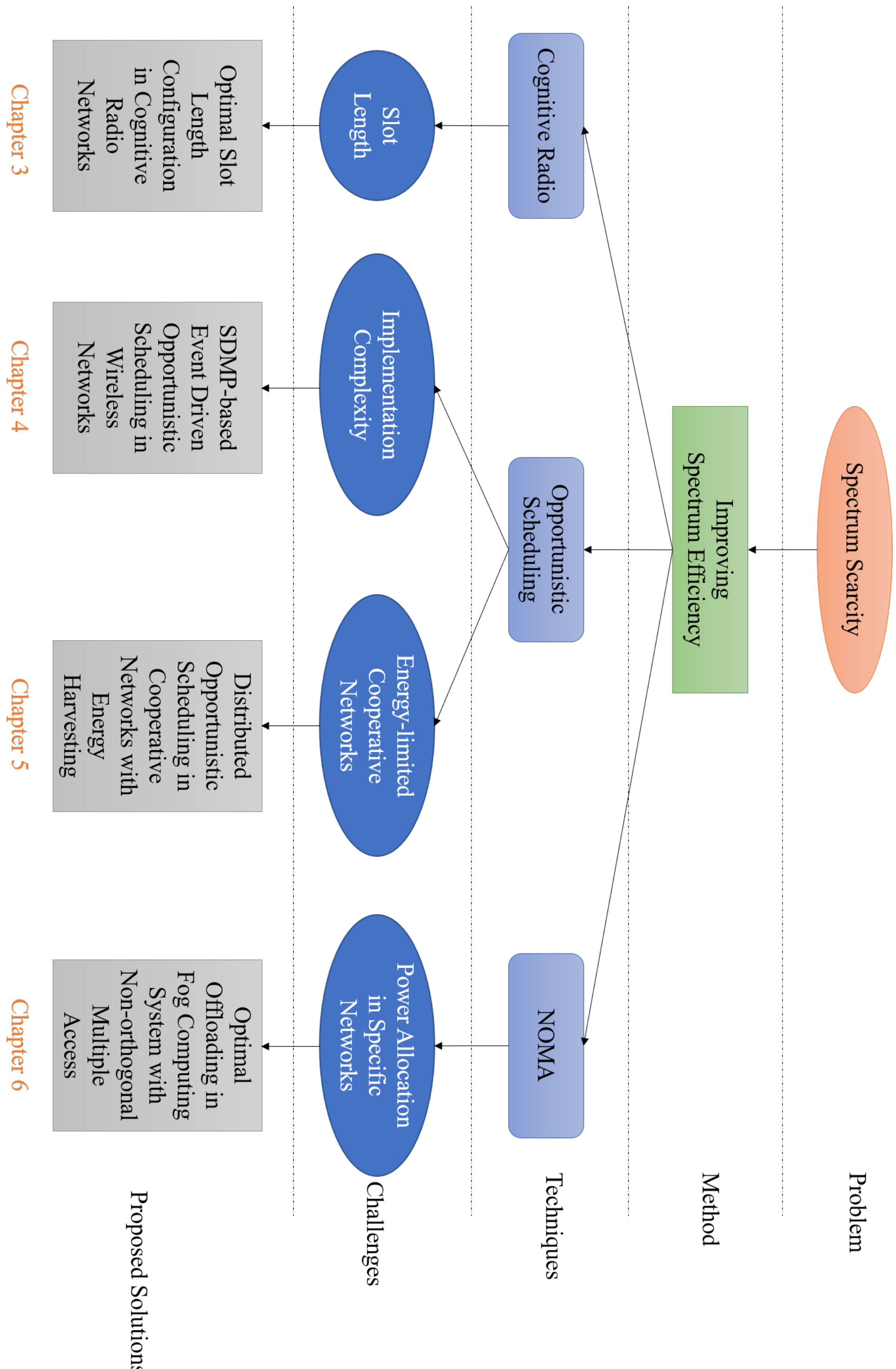

Figure 1.7: The structure of the thesis.

Chapter 2

Background

2.1 Stopping Rule Problems

The definition of a stopping rule problem[1] can be described as follows. A random process is sequentially observed. At each observation (indexed by $n = 1, 2, \cdots$), a random variable X_n can be observed. The probability distribution of $X_n = x_n$ is usually known to the user[2]. Thus, after observation $X_n = x_n$, the user needs to decide whether to stop the observation and receive the known reward $Y_n = y_n(x_1, x_2, \cdots, x_n)$ (i.e., the reward when stopping at moment n with observation $x_1, x_2, ..., x_n$) or to continue and observe X_{n+1}. The problem is to find the optimal stopping rule, i.e., at which moment the user should stop so that maximal expected reward can be achieved. For example, we consider a house-selling problem [49]. Let X_n denote the amount of the received offer on day n. There is a cost, denoted $c > 0$, for observing each offer (e.g., cost of living or maintenance). Then, after receiving an offer $X_n = x_n$, you decide whether to accept the offer or to wait for the next offer. It is assumed that the past offer cannot be recalled and accepted. When an offer $X_n = x_n$ is received, the reward function, denoted $Y_n = y_n(x_1, x_2, \cdots, x_n)$, should be

$$y_n = \begin{cases} 0, & n = 0 \\ -\infty, & n = \infty \\ x_n - nc, & 0 < n < \infty \end{cases} . \tag{2.1}$$

For a stopping rule problem, an optimal stopping rule may not exist. Thus, to solve a stopping rule problem, we need to verify whether an optimal stopping rule exists or not. The existence of an optimal stopping rule can be guaranteed when the

[1] More detailed information about stopping rule problems can be found in [49].

[2] Here we use capital letter X to denote a random variable, and lower-case letter x to denote a realization of the random variable.

following two conditions are satisfied.

$$C1. \quad E\left[\sup_n Y_n\right] < \infty \tag{2.2}$$

$$C2. \quad \limsup_{n\to\infty} Y_n \le Y_\infty \quad a.s. \tag{2.3}$$

where $E[\cdot]$ means expectation, Y_∞ is the reward if the user never stops, and $a.s.$ denotes "almost surely".

In general, for a stopping rule problem, it is hard to derive the optimal stopping rule with a closed-form. However, the optimal stopping rule of a finite horizon stopping rule problem can be obtained by the following method. A stopping rule problem is defined as a finite horizon stopping rule problem if the user must stop the observation after observing X_T. In this case, the problem has horizon T. For example, for the house-selling problem, if the house is required to sell within T days, the problem is a finite horizon stopping rule problem. A finite horizon stopping rule problem can be dealt with by using a backward induction method. Let $V_n^T(x_1, \cdots, x_n)$ denote the maximum reward the user can achieve at stage $n \in \{0, 1, 2, \cdots, T\}$ after observing $\{x_1, \cdots, x_n\}$. Then, we have

$$V_n^T(x_1, \cdots, x_n) = \max\left\{Y_n, E\left[V_{n+1}^T(x_1, \cdots, x_n, X_{n+1})\,|X_1 = x_1, \cdots, X_n = x_n\right]\right\}. \tag{2.4}$$

And the optimal stopping rule, denoted N^*, should be

$$N^* = \min\left\{n \ge 0 : Y_n \ge E\left[V_{n+1}^T(x_1, \cdots, x_n, X_{n+1})\,|X_1 = x_1, \cdots, X_n = x_n\right]\right\}. \tag{2.5}$$

Due to $V_T^T(x_1, \cdots, x_T) = Y_T$ (i.e., the user must stop the observation after observing X_T), the optimal action (i.e., stop or continue) at stage $(T-1)$ can be derived by (2.4) and (2.5). Then, the optimal action at stage $(T-2)$ can be found and so on back to stage 0. Therefore, the optimal stopping rule can be derived. However, the large time and space complexity is the major challenge of the backward induction method.

In reality, lots of stopping rule problems are generally not finite horizon problems. For an infinite horizon stopping rule problem, if it repeats in time, the optimal stopping rule N^* is to maximize the reward ratio (i.e., the average received reward per time unit), denoted $E[Y_N]/E[T_N]$ where T_N is the duration until a stop. For example, for the house-selling problem, the selling process would repeat in time if we have lots of houses to sell and need to sell them one by one. In this case, for an optimal stopping rule, the maximal average received reward per time unit should be achieved, e.g., selling 1 house per week with a reward \$1000 per sale is better than selling 1 house in a month with a reward \$2000. For this kind of problem, we have the following theorem [49].

Theorem 1. *For any* λ*, we have* $N(\lambda) = \arg\sup_N E[Y_N - \lambda T_N]$*. If* $\sup_N E[Y_N - \lambda T_N] = 0$ *is obtained at* λ^**, then* $\sup_N E[Y_N]/E[T_N] = \lambda^*$ *is satisfied and* $N(\lambda^*)$ *is the optimal stopping rule for the problem.*

However, it still has some challenging issues to solve this kind of problem. Firstly, we need to prove the conditions which guarantee the existence of an optimal stopping rule. Secondly, the stopping rule $N(\lambda)$ needs to be derived for any λ. Thirdly, we need to derive the value of λ^*.

In summary, to solve a stopping rule problem, we need to prove the existence of the optimal stopping rule and derive the optimal stopping rule.

2.2 Markov Decision Process

Markov decision processes (MDPs) are particularly useful for dealing with decision making problems. In a decision making problem, the time point with making a decision (i.e., taking an action) is referred to as decision epoch. There are two kinds of MDP, discrete time MDP (DTMDP) and continuous time MDP (CTMDP). In a DTMDP, the decision maker takes an action at each predetermined discrete time point. On the other hand, in a CTMDP, the time point to take an action is random [50].

2.2.1 Discrete Time Markov Decision Process

For a DTMDP, it can be represented by a tuple $\{\mathbf{S}, \mathbf{A}, p(s'|s,a), r(s,a)\}$, where each element is analyzed as follows.

- $\mathbf{S}$: It denotes the state space, which is the set of all possible states in the MDP. At each decision epoch, a state, denoted $s \in \mathbf{S}$, can be observed.

- **A**: It denotes the action space, which is the set of all possible actions in the MDP. When a state is observed at each decision epoch, an action, denoted $a \in \mathbf{A}$, is chosen to be taken.

- $p(s'|s, a)$: It denotes the transition probability, which means the probability that taking action a in state s leads to state s'.

- $r(s, a)$: It denotes the reward when an action a is taken in state s.

For an MDP, the core problem is to find a policy for the decision maker. A policy, denoted π, is defined as a mapping from the state space $\mathbf{S}$ to the action space $\mathbf{A}$. Accordingly, given a policy, the action is decided at any state. To evaluate a policy, there are generally two kinds of criteria, the discounted expected total reward criterion and the average reward criterion. Since the discounted expected total reward criteria is widely adopted in MDPs, we only discuss it here. For the average reward criterion, pleas refer to [51] for details.

For a discounted DTMDP (i.e., a DTMDP with the discounted expected total reward criterion), we try to find the optimal policy, denoted π^*, which maximizes the discounted expected total reward. Accordingly, the policy π^* is defined as

$$\pi^* = \arg\max_{\pi} \sum_{n=0}^{N} \alpha^n r(s_n, a_n) \tag{2.6}$$

where n is the index of the decision epoch, $N > 0$ is the range of the MDP,[3] s_n is the state of the nth decision epoch, $a_n = \pi(s_n)$ is the action which is taken in state s_n by following policy π. Accordingly, the objective of solving an MDP is to derive the optimal policy π^*. The Bellman equation of the discounted DTMDP is given as [52]

$$V^{\pi}(s) = r(s, a) + \alpha \sum_{s' \in \mathbf{S}} p(s'|s, a) V^{\pi}(s') \tag{2.7}$$

where $V^{\pi}(s)$ is the state value function of state s given a policy π (i.e., the discounted expected total reward given a policy π) and $a = \pi(s)$. Then, the optimal policy π^* should satisfy the Bellman optimality equation, which is given as

$$V^*(s) = \max_{a \in \mathbf{A}} \{r(s, a) + \alpha \sum_{s' \in \mathbf{S}} p(s'|s, a) V^*(s')\} \tag{2.8}$$

[3]The MDP is an infinite horizon MDP if $N = \infty$. Otherwise, the MDP is a finite horizon MDP.

where $V^*(s)$ is the optimal state value of state s. Then, we have

$$\pi^*(s) = \arg\max_{a \in \mathbf{A}} \{r(s,a) + \alpha \sum_{s' \in \mathbf{S}} p(s'|s,a) V^*(s')\}. \tag{2.9}$$

For a discounted DTMDP, it is said to be fully explored when the transition probability $p(s'|s,a)$ can be obtained. Then, a typical method, namely value iteration method [53], is introduced to derive the optimal policy π^*. In value iteration method, the optimal state value of any state s can be derived by the following iterative equation.

$$V^{n+1}(s) = \max_{a \in \mathbf{A}} \{r(s,a) + \alpha \sum_{s' \in \mathbf{S}} p(s'|s,a) V^n(s')\}. \tag{2.10}$$

Then, the optimal policy can be obtained by (2.9).

If the detailed information of the DTMDP is hard to be explored (e.g., the transition probability cannot be obtained), it is regarded as partial explored, and thus, the value iteration method is not working. To address this challenge, reinforcement learning algorithms are introduced [54]. In a reinforcement learning algorithm, it solves the Bellman optimality equation to obtained the optimal policy by asynchronous iteration. For example, in Q-learning which is a typical reinforcement learning method [55], a Q-value, denoted $Q^\pi(s,a)$, is introduced to express the expected long-term discounted reward of state-action pair (s,a) and $a = \pi(s)$. Then, after a long learning period, the optimal Q-value, denoted $Q^*(s,a)$, of state-action pair (s,a) can be obtained by the following iterative equation.

$$Q^{n+1}(s,a) = Q^n(s,a) + \kappa(r(s,a) + \alpha \max_{a' \in \mathbf{A}} Q^n(s',a') - Q^n(s,a)) \tag{2.11}$$

where $\kappa \in (0,1]$ is the learning rate and a' denotes the applied action of state s'. Then, the optimal policy can be obtained by

$$\pi^*(s) = \arg\max_{a \in \mathbf{A}} Q^*(s,a). \tag{2.12}$$

However, in lots of scenarios (e.g., queueing control systems), an action may need to be taken at any random time point instead of a set of predetermined discrete time points. Accordingly, a CTMDP should be modeled in these scenarios. The most general CTMDP model is semi-Markov decision process (SMDP) [50].

2.2.2 Semi-Markov Decision Process

For an SMDP, it can be represented by a tuple $\{\mathbf{S}, \mathbf{A}, p(s'|s,a), \tau(s,a,s'), r(s,a,s')\}$, where $\mathbf{S}$, $\mathbf{A}$, and $p(s'|s,a)$ have the similar definition with those in a DTMDP, and $\tau(s,a,s')$ and $r(s,a,s')$ are analyzed as follows.

- $\tau(s,a,s')$: It denotes the duration time from state s to the next state s' after taking action a in state s. Note that $\tau(s,a,s')$ is a random variable. An example of the time line for a DTMDP and an SMDP is given in Fig. 2.1, where t_i is the ith decision epoch.

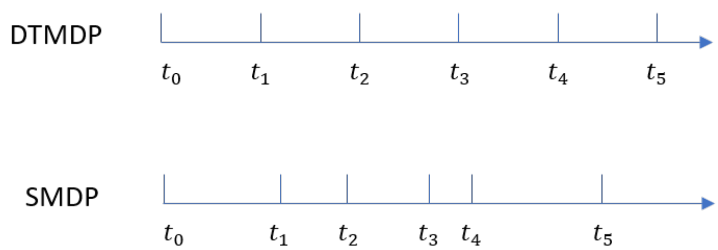

Figure 2.1: An example of time line for discrete time MDP and SMDP.

- $r(s,a,s')$: It denotes the reward function. In generally, it contains two parts, an instantaneous reward/cost $w(s,a)$ and a continuous reward/cost $c(s,a,s')$ during the period $\tau(s,a,s')$.

Similarly, deriving the optimal policy π^*, which maximizes the discounted expected total reward, is the major objective of solving an SMDP. Thus, the policy π^* satisfies

$$\begin{aligned}\pi^* &= \arg\max_{\pi} \sum_{n=0}^{\infty} e^{-\alpha t_n} r(s_n, a_n, s_{n+1}) \\ &= \arg\max_{\pi} \sum_{n=0}^{\infty} e^{-\alpha t_n} [w(s,a) + \textstyle\int_{t_n}^{t_{n+1}} e^{-\alpha(t-t_n)} uc(s,a,s')dt]\end{aligned} \tag{2.13}$$

where t_n denotes the nth decision epoch, α is the discount factor, and u denotes a weight. Note that $t_n \in (0, \infty)$ is a random variable, and thus, an SMDP is generally an infinite horizon MDP.

Different from the DTMDP, the value iteration method is not straightforward for a fully explored SMDP (i.e., the transition probability and the reward function can be obtained). The iterative equation (2.10) cannot obtain the optimal state value of a state in SMDP since it does not consider the non-identical transition time (i.e., the duration between two decision epochs). Thus, to address this challenge, we need to convert the SMDP into a DTMDP. Then, the value iteration method can be applied to derive the optimal policy. Accordingly, the core problem is how to convert an SMDP to an identical DTMDP. If the detailed information of the SMDP is hard to explore, a reinforcement learning algorithm is needed. However, due to the random decision epochs, the traditional Q-learning algorithm cannot derive the optimal policy of an SMDP. Thus, how to improve the traditional reinforcement learning methods to solve an SMDP problem is pretty meaningful.

2.3 Lyapunov Optimization

Lyapunov optimization is a powerful technique for optimally controlling a dynamical system [56]. It is widely adopted to the optimization problems, which are to stabilize the dynamic system (e.g., queues in the system) while optimizing a performance objective. For example, in a dynamic control system, a control action is assumed to be taken at each time slot (indexed by $t \in \{0, 1, 2, \cdots\}$). There are K queues and the backlog of jth ($j \in \{1, 2, \cdots, K\}$) queue at tth time slot is denoted as $Q_j(t)$. The action which is taken at each time slot affects the backlog of each queue (i.e., the action might affect the arrival rate or departure rate), and also incurs a penalty $y(t)$. The goal of this system is to select a reasonable action at each time slot (i.e., a policy ψ^*) which stabilizes all queues while minimizing the time average of penalty. Accordingly, the problem is formulated as follows.

$$\psi^* = \arg\min_{\psi} \lim_{\tau \to \infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} E[y(t)] \tag{2.14}$$

$$s.t. \lim_{t \to \infty} \frac{E[|Q_j(t)|]}{t} = 0 \tag{2.15}$$

where condition (2.15) guarantees that each queue can keep stable (i.e. mean rate stable).

To measure the size of all queues, a Lyapunov function [57], denoted $\Gamma(t)$, is defined as the sum of squares of backlog in all queues. Thus, we have

$$\Gamma(t) \triangleq \frac{1}{2} \sum_{j=1}^{K} w_j Q_j(t)^2 \tag{2.16}$$

where w_j denotes a weight for each queue. Intuitively, $\Gamma(t)$ takes a small value only when the backlogs of all queues are small. Then, a function, named conditional Lyapunov drift function, is defined as the difference of the Lyapunov function from one time slot to the next time slot. Thus, we have

$$\Delta(t) \triangleq E[\Gamma(t+1) - \Gamma(t)|\Theta(t)] \tag{2.17}$$

where $\Theta(t) \triangleq \{Q_1(t), Q_2(t), \cdots, Q_K(t)\}$. To evaluate both the system penalty and the queue stability, the drift-plus-penalty expression is defined as,

$$\Delta(t) + VE[y(t)|\Theta(t)] \tag{2.18}$$

where $V \geq 0$ is a control parameter which is used to tradeoff the average penalty and queue stability. Then, the Lyapunov optimization theorem [58] is given as follows.

Theorem 2. *Suppose there are constants $\beta > 0$ and $B \geq 0$ such that the following expression holds for $\forall t \in \{0, 1, 2, \cdots\}$.*

$$\Delta(t) + VE[y(t)|\Theta(t)] \leq B + Vy^* - \beta \sum_{j=1}^{K} Q_j(t) \tag{2.19}$$

where y^* is a target value for the time average of $y(t)$ (i.e., $y^* = \min \lim_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} E[y(t)]$). Then, if $E[y(t)] \geq y_{min}$ is given, the following two expressions about the time average penalty and backlog of queues are satisfied.

$$\lim_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} E[y(t)] \leq y^* + \frac{B}{V} \tag{2.20}$$

$$\lim_{\tau\to\infty} \frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{j=1}^{K} E[|Q_j(t)|] \leq \frac{B + V(y^* - y_{min})}{\beta}. \tag{2.21}$$

According to Theorem 2, there is an alternative method to solve the problem in (2.14)-(2.15). Instead of solving problem (2.14)-(2.15) directly, we can try to derive a policy, denoted ψ', to minimize the upper bound of the drift-plus-penalty expression. Then, no knowledge of the probability distributions of the random events (e.g., the arrival rate of each queue) is required, and thus, it can largely decrease the implementation complexity. In addition, the policy ψ' can still reach a good performance, i.e., the time average penalty can reach at most $\mathbf{O}(1/V)$ above y^* and each queue can keep stable, in which $\mathbf{O}(\cdot)$ means big O notation. Note that y^* is the minimum time average penalty which is obtained by policy ψ^*. Therefore, the core problem of Lyapunov optimization is how to derive a policy to minimize the upper bound of the drift-plus-penalty expression efficiently.

2.4 A Typical NOMA Scheme

In this section, a typical NOMA scheme is discussed. A typical NOMA scenario includes one transmitter A and N receivers B_i, $i \in \{1, 2, \cdots, N\}$. In the scenario, A transmits data to all receivers simultaneously with a constraint of total power P. Each link from A to B_i experiences independent and identically distributed (i.i.d.) Rayleigh block fading, and the channel gain is denoted as h_i. All links are sorted as $|h_1| > |h_2| > \cdots > |h_N|$. The transmit power to the signal of receiver B_i is set as a fraction, denoted β_i, of the total power. Then, A transmits the superposition coded signal of all receivers. According to the analysis of SIC, for B_i, it can decode and subtract the signal of $B_j, j \in \{i+1, i+2, \cdots, N\}$, and then, decodes itself signal with treating the signal of $B_k, k \in \{1, 2, \cdots, i-1\}$ as interference. Accordingly, the achievable data rate, denoted as R_i, of B_i is given as

$$R_i = W \log_2 \left(1 + \frac{\beta_i P |h_i|^2}{\sum\limits_{j=1}^{i-1} \beta_j P |h_i|^2 + \sigma^2} \right) \tag{2.22}$$

where W is the channel bandwidth and σ^2 is the variance of the additive white Gaussian noise (AWGN).

To evaluate the performance of the NOMA scheme, we consider the scenario with two receivers (i.e., Fig. 1.6). Then, the achievable data rate of B_1 and B_2 is given as

$$R_1 = W \log_2 \left(1 + \frac{\beta_1 P |h_1|^2}{\sigma^2} \right) \tag{2.23}$$

$$R_2 = W \log_2 \left(1 + \frac{(1-\beta_1) P |h_2|^2}{\beta_1 P |h_2|^2 + \sigma^2} \right). \tag{2.24}$$

With OMA, it is assumed that a fraction, denoted α, of the wireless resource block is allocated to the link from A to B_1, and thus, the rest of the wireless resource block is allocated to the link from A to B_2. Then, the achievable data rate of B_1 and B_2 is given as

$$R_1 = \alpha W \log_2\left(1 + \frac{P|h_1|^2}{\sigma^2}\right) \tag{2.25}$$

$$R_2 = (1-\alpha) W \log_2\left(1 + \frac{P|h_2|^2}{\sigma^2}\right). \tag{2.26}$$

It can be shown that the overall achievable data rate (i.e., $R_1 + R_2$) of NOMA is much higher than that of OMA. And a numerical performance comparison between NOMA and OMA is given in Fig. 2.2.

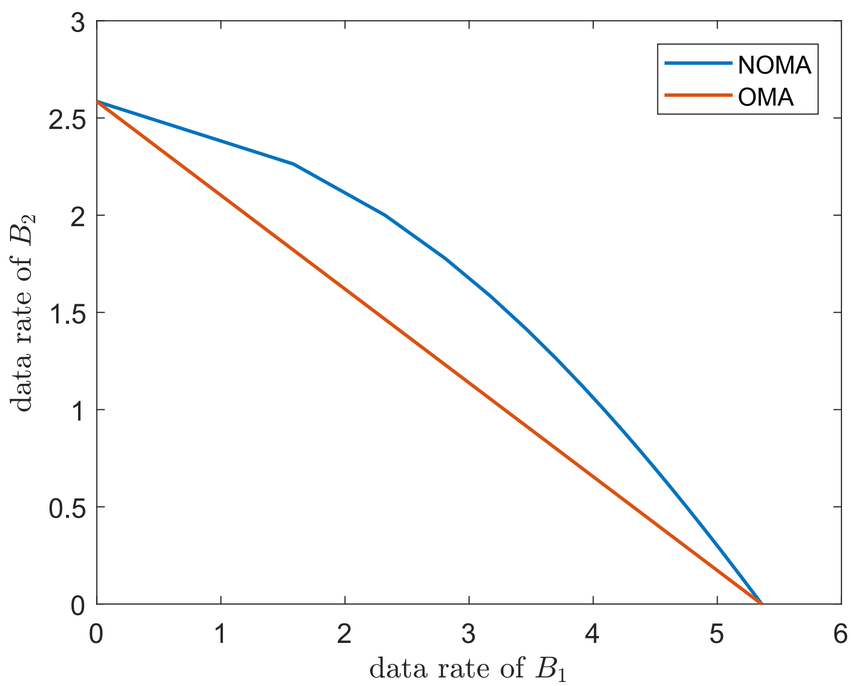

Figure 2.2: Data rate of NOMA and OMA.

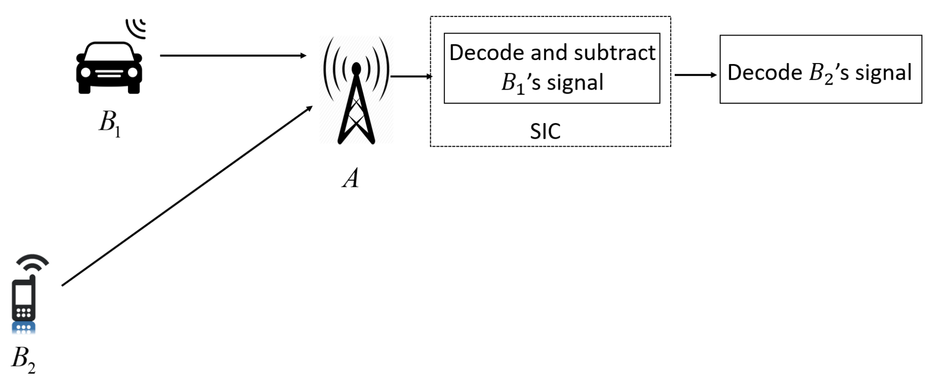

Figure 2.3: An example of uplink NOMA scheme.

Similarly, NOMA can be applied in uplink transmissions. For example, an example of uplink NOMA scheme is given in Fig. 2.3. Note that A would decode and subtract the signal from strong transmission, and then decode the signal from weak transmission. The details about uplink NOMA scheme can be found in [59] and [60].

In summary, with NOMA, not only the spectrum efficiency (i.e., the maximum capacity/throughput) can be largely improved, but also the number of supportable transmissions under the limited wireless resource blocks can be largely increased. In addition, under a reasonable power allocation for different users, the fairness of the users can be guaranteed [61].

Chapter 3

Optimal Slot Length Configuration in Cognitive Radio Networks

In this chapter, a slot length configuration scheme with imperfect spectrum sensing is proposed. In the proposed scheme, the spectrum sensing result is considered when configuring the slot length. Then, an optimization problem to find out the optimal slot length configuration is formulated and analyzed. And an algorithm is proposed to solve the problem.

3.1 Introduction

To cope with the problems of spectrum shortage [1,4] and spectrum under-utilization [62], cognitive radio is one possible solution, which largely improves the spectrum efficiency by allowing secondary users to access spectrum holes.

In cognitive radio networks, a slotted time structure is widely used [63]. Time is divided to fixed-duration slots. Spectrum sensing is performed at the beginning of each slot. If no primary signal is detected by spectrum sensing, secondary users can transmit over the channel during the remaining time of the slot. On the other hand, if primary signals are detected, secondary users should keep silent at this slot.

There are several spectrum sensing techniques, including energy detection, matched filter detection, cyclostationary feature detection, and so on [64]. Due to its low complexity, energy detection attracts the most attention, and thus, it is also adopted in this work. In energy detection, the detection accuracy of spectrum sensing is largely affected by the length of spectrum sensing within a slot. There are many existing works that consider configuration of the length of spectrum sensing. In [18], the optimal sensing duration in a cognitive radio network is derived to maximize the detection accuracy of spectrum sensing. In addition, the case with cooperative spectrum sensing is also considered. In [65], a problem to maximize the average throughput of secondary transmissions with energy harvesting is formulated. To guarantee the quality of service (QoS) of primary users and the energy efficiency, there are a collision constraint and an energy consumption constraint in the optimization problem. Accordingly, the optimal sensing duration is derived by solving the formulated optimization problem. In [66], a sensing scheduling optimization problem is considered in cognitive radio networks with cooperative spectrum sensing. Considering different scenarios, three different sensing strategies are proposed. In [67], the power control and spectrum sensing length are jointly investigated. Since the formulated problem is a non-convex optimization problem, an iterative algorithm is proposed to solve it. Then, the optimal length of spectrum sensing duration and the optimal power control strategy are derived.

On the other hand, the slot length configuration, which is another factor that can largely affect the performance of cognitive radio networks, does not receive enough attention in the literature. In most of existing works [65]- [67], it is assumed that the channel state (idle or busy) does not change within a slot of secondary users. Here an idle channel state denotes that no primary signal exists in the licensed channel, while a busy channel state denotes that primary users are transmitting in the licensed channel. This assumption is actually not reasonable in practice. Primary transmission is a stochastic event, and thus, it may start at any moment within a slot of secondary users. However, the spectrum sensing operation is only taken at the beginning of a slot. Consider the case that a secondary user senses a channel to be idle at the beginning of a time slot, and thus, transmits in the time slot. If primary users become active in the middle of the time slot, there is a collision between the primary transmission and the secondary transmission. Taking this into account, a continuous Markov model is proposed in [42], in which the state of a licensed channel alternates between "idle" and "busy" based on a continuous Markov model. The duration of each idle/busy state is an independent and identically distributed (i.i.d.) random variable. Considering this model, the slot length of secondary users is a factor that largely affects the performance of cognitive radio networks. A smaller slot length can lead to shorter collision time[1], and thus, a higher QoS level for primary transmissions is provided. However, a higher frequency to perform the spectrum sensing is also needed. This means that secondary users have to cost more time and energy for spectrum sensing, and thus, less time is left for secondary transmissions. Accordingly, there is a tradeoff between the QoS of primary users and the reward of secondary users by setting the slot length. In [42], the optimal slot length is derived to maximize the spectrum sensing efficiency. In [68], the slot length configuration problem with cooperative spectrum sensing is considered. In [69], an optimization problem which jointly considers the channel selection and slot length configuration is formulated. In the formulated problem, the reward, which equals the amount of data that a secondary user can transmit minus the penalty of collision time, is maximized. Then, an adaptive method is derived by solving the formulated problem.

We have the following observations for the above existing works [42, 68, 69] on slot length configuration.

- In these works [42, 68, 69], the length of a slot is set as a fixed value regardless of the spectrum sensing result. However, for a licensed channel, the sojourn time of idle state and busy state are usually different. Accordingly, we argue that the spectrum sensing result should be considered when configuring the slot length.

- In these works [42, 68, 69], it is assumed that the spectrum sensing is perfect which may not be practical. In the literature for cognitive radio with imperfect sensing, to protect primary users, the missed-detection probability of spectrum sensing is usually required to be less than a threshold (say α). Indeed, this setting can guarantee that the collision ratio of primary activities (i.e., the percentage of time in which the primary activities are interfered with) is below α if the primary user state (busy or idle) does not change within a time slot of secondary users. However, with a practical setting that primary user state may change within a time slot of secondary users, the collision ratio of primary activities may be more than α. Accordingly, we argue that we should have a constraint on the collision ratio of primary activities when configuring the slot

[1] Collision time means the duration that a primary transmission is interfered with by secondary transmissions.

length. To the best of our knowledge, the collision ratio of primary activities is overlooked in existing literature when designing cognitive radio systems.

Taking into account the two observations, an optimal slot length configuration scheme is proposed in this work. The major contributions of this work are summarized as follows.

1. A slot length configuration scheme with imperfect spectrum sensing is proposed. In the proposed scheme, the spectrum sensing result is considered when configuring the slot length. Therefore, slots with different sensing results may have different slot length.

2. In the proposed scheme, an optimization problem is formulated to derive the optimal slot length configuration. In the formulated problem, the throughput of secondary transmissions is maximized under a constraint on the collision ratio of primary activities and a constraint on sensing frequency. Then, we give a detailed analysis for the formulated problem.

3. An algorithm is proposed to solve the formulated problem, and thus, the optimal slot length configuration can be derived. In addition, if the spectrum sensing is perfect, an algorithm with much less complexity is proposed to derive the optimal slot length configuration.

4. The proposed slot length configuration scheme is evaluated by simulation. It shows that, by having different slot length with different sensing results, largely improved performance can be achieved.

The rest of the chapter is organized as follows. The system model and the slot length configuration scheme are proposed in Section 3.2. The optimization problem is formulated in Section 3.3, and analyzed and solved in Section 3.4. Section 3.5 shows simulation results. Finally, Section 3.6 concludes this chapter.

3.2 System Model and Proposed Scheme

3.2.1 System Model

Consider that a secondary user wishes to access a licensed channel. Over the licensed channel, primary users have the priority for channel access. The secondary user can utilize the channel only when the channel is sensed as idle by spectrum sensing. As shown in Fig. 3.1, the channel is modeled as an idle/busy process. Here an idle state means there are no primary activities over the channel, while a busy state means that primary users are transmitting over the channel. Over time, the sojourn durations of idle and busy states are i.i.d. random variables with probability density function (PDF) $f_{\mathrm{i}}(x)$ and $f_{\mathrm{b}}(x)$, respectively [69].

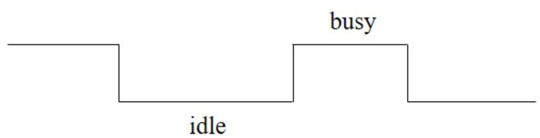

Figure 3.1: The idle/busy channel model.

To protect primary transmissions, spectrum sensing by using energy detection is performed by the secondary user at the beginning of each time slot. If the output of the energy detector (i.e., the energy level of the detected signal) is more than a predetermined threshold denoted ε, the channel is estimated to be busy; otherwise, the channel is estimated to be idle.

In general, two metrics, probability of detection (denoted as P_{d}) and probability of false alarm (denoted as P_{f}), are used to measure the detection performance of spectrum sensing. The probability of detection P_{d} denotes the probability that the sensing result is busy given that primary users are indeed transmitting over the channel. The probability of false alarm P_{f} denotes the probability that the sensing result is busy given that no primary signal actually exists in the channel. The detailed definitions of P_{d} and P_{f} are given in Section 1.2.2. The expressions of P_{d} and P_{f} are given as [18]

$$P_{\mathrm{d}} = Q\left((\frac{\varepsilon}{\sigma^2} - \gamma_{\mathrm{p}} - 1)\sqrt{\frac{T_{\mathrm{s}} f_{\mathrm{s}}}{\gamma_{\mathrm{p}} + 1}} \right), \tag{3.1}$$

$$P_{\mathrm{f}} = Q\left((\frac{\varepsilon}{\sigma^2} - 1)\sqrt{T_{\mathrm{s}} f_{\mathrm{s}}} \right), \tag{3.2}$$

where T_s is the spectrum sensing time, f_s is the sampling frequency, $T_\mathrm{s}f_\mathrm{s}$ is the total number of samples, σ^2 is the variance of additive white Gaussian noise (AWGN), γ_p means the signal-to-noise ratio (SNR) of primary signal received by the secondary user, and $Q(\cdot)$ is the complementary distribution function of the standard Gaussian. According to (3.1) and (3.2), the sensing time T_s can be derived if a pair of target probabilities $(P_\mathrm{d}, P_\mathrm{f})$ is given.

3.2.2 Proposed Slot Length Configuration Scheme

In existing works, the slot length is usually set as a fixed value without considering the state of the licensed channel. However, the sojourn time of the licensed channel's idle state and busy state are usually different. To better protect primary users and improve performance of the secondary user, the sensing result should be considered for configuring the slot length. Therefore, the proposed slot length configuration scheme is stated as follows. For a slot, spectrum sensing with duration T_s is carried out at the beginning of the slot. If the sensing result is idle, the slot length is set as T_i, and thus, the secondary user transmits within the subsequent time duration $(T_\mathrm{i} - T_\mathrm{s})$ of the slot. Otherwise, the slot length is set as T_b, and thus, the secondary user keeps silent within the subsequent time duration $(T_\mathrm{b} - T_\mathrm{s})$ of the slot. An example of the slot structure is shown in Fig. 3.2, which has four scenarios.

- If no primary signal exists in the channel and the sensing result is busy (i.e., a false alarm happens), the slot length is T_b;
- If no primary signal exists in the channel and the sensing result is idle (i.e., accurate sensing happens), the slot length is T_i;
- If the channel is occupied by primary users and the sensing result is idle (i.e., missed detection happens), the slot length is T_i;
- If the channel is occupied by primary users and the sensing result is busy (i.e., accurate sensing happens), the slot length is T_b.

As the sensing duration T_s is usually much smaller than the average sojourn time of the channel idle or busy state, we assume that the channel state does not switch within a sensing period. However, the channel state may switch within a slot. Fig. 3.3

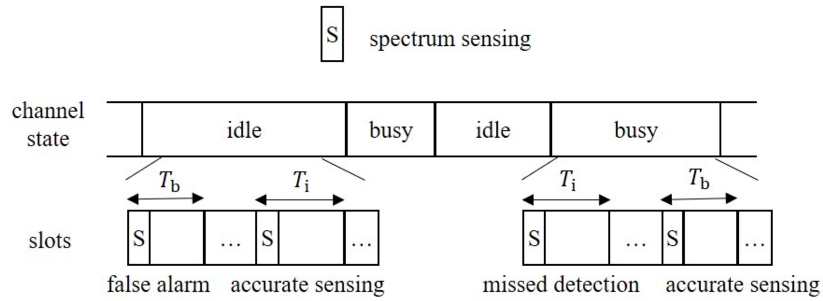

Figure 3.2: Example of the slot structure.

shows examples of one-time switching (i.e., the channel state switches once within a slot) and multi-times switching (i.e., the channel state switches two or more times within a slot). Switching within a time slot may lead to collision of primary and secondary transmissions. It may also lead to waste of transmission opportunity. We use the examples in Fig. 3.3 to demonstrate.

- For the one-time switching in Fig. 3.3, at the sensing period of the time slot, the channel is idle. If sensing is accurate, the secondary user transmits in the slot. When the switching happens (i.e., primary users become busy), we have a collision.

- for the multi-times switching in Fig. 3.3, at the sensing period of the time slot, the channel is busy. If sensing is accurate, the secondary user keeps silent within the slot. When the two switchings happen, we see that within the time slot there is a duration in which primary users are idle, i.e., in this duration there is a transmission opportunity. This transmission opportunity is wasted.

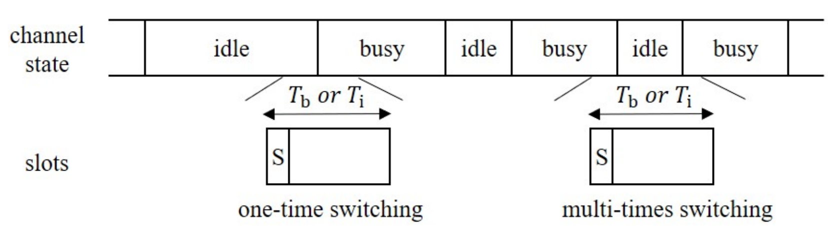

Figure 3.3: Examples of switching in a time slot.

In the proposed scheme, it is critical to determine the value of T_i and T_b. Therefore, to find optimal values of T_i and T_b, a problem, which maximizes the average throughput of secondary transmissions under a primary transmission QoS constrain and a sensing frequency constraint, is formulated, analyzed, and solved in the subsequent two sections.

3.3 The Formulated Problem

To protect primary transmissions, we introduce a *collision ratio* η_{c}, which is defined as

$$\eta_{\mathrm{c}} = \frac{T_{\mathrm{c-in-slot}}}{T_{\mathrm{b-in-slot}}} \tag{3.3}$$

where $T_{\mathrm{c-in-slot}}$ is the expected duration of collision (between primary and secondary transmissions) in any slot, and $T_{\mathrm{b-in-slot}}$ is the expected channel busy duration in any slot. Note that $T_{\mathrm{c-in-slot}}$ and $T_{\mathrm{b-in-slot}}$ can be obtained by (3.39) and (3.38), respectively. Thus, a smaller value of η_{c} means a higher level of protection to primary transmissions. Accordingly, to protect primary transmissions in our scheme, the collision ratio η_{c} is required to be not more than a predefined threshold, denoted ε_{c}.

To explore as many transmission opportunities as possible, the secondary user should take another spectrum sensing as soon as possible if the current sensing result is busy. However, by doing this, energy consumption will be high. And the energy of a secondary user is generally limited [71]. Therefore, to guarantee the energy efficiency, we introduce a *sensing ratio* η_{s}, which is defined as

$$\eta_{\mathrm{s}} = \frac{T_{\mathrm{s}}}{T_{\mathrm{slot}}}, \tag{3.4}$$

where T_{slot} is the expected time duration of a slot. A smaller value of η_{s} means a lower frequency of the spectrum sensing. Accordingly, to guarantee the energy efficiency in our scheme, the sensing ratio η_{s} is required to be not more than a predefined threshold, denoted ε_{s}.

To find out the optimal value of T_{i} and T_{b}, we formulate an optimization problem, named Problem P1, to maximize the average throughput of secondary transmissions.

$$\begin{aligned}
\text{P1}: \quad & \max_{T_\mathrm{i}, T_\mathrm{b}} E[R_\mathrm{slot}] && (3.5a)\\
s.t. \quad & \eta_\mathrm{c} \le \varepsilon_\mathrm{c} && (3.5b)\\
& \eta_\mathrm{s} \le \varepsilon_\mathrm{s} && (3.5c)\\
& T_\mathrm{i} \ge T_\mathrm{s} && (3.5d)\\
& T_\mathrm{b} \ge T_\mathrm{s} && (3.5e)
\end{aligned}$$

where R_slot denotes the throughput of secondary transmissions and $E[\cdot]$ means expectation operation. In constraints (3.5d) and (3.5e), intuitively, we set T_i and T_b to be more than the spectrum sensing duration T_b.

3.4 Problem Analysis and Optimal Solution

3.4.1 Analysis of collision and wasting in a slot

For a slot, the secondary user would utilize the licensed channel to the end of the slot if the spectrum sensing result is idle. Otherwise, the secondary user keeps silent until the end of the slot. Due to the imperfect spectrum sensing (i.e., *false alarm* and *missed detection* as shown in Fig. 3.2) and the possible channel state switching within a slot (i.e., *one-time switching* and *multi-times switching* as shown in Fig. 3.3), there might be collision or wasting of transmission opportunity. Some examples of collision and wasting are given in Fig. 3.4.

Recall that the expected collision duration in a slot is denoted as $T_\mathrm{c-in-slot}$. We denote the expected wasting duration in a slot as $T_\mathrm{w-in-slot}$.

In our system, each slot belongs to one of the following two types.

- *i-slot*: The actual channel state at the beginning of the slot is idle.
- *b-slot*: The actual channel state at the beginning of the slot is busy.

Let $C_\mathrm{i}(t)$ and $C_\mathrm{b}(t)$ denote the expected collision duration within a period $(t_\mathrm{s}, t_\mathrm{s} + t)$ if the channel state at moment t_s is idle and busy, respectively. Let $\tilde{C}_\mathrm{i}(t)$ and $\tilde{C}_\mathrm{b}(t)$ denote the expected channel busy duration within a period $(t_\mathrm{s}, t_\mathrm{s} + t)$ if the channel state switches from busy to idle and from idle to busy, respectively, at moment t_s. According to Fig. 3.4, an illustration about $C_\mathrm{i}(t)$, $C_\mathrm{b}(t)$, $\tilde{C}_\mathrm{i}(t)$, and $\tilde{C}_\mathrm{b}(t)$ is given in

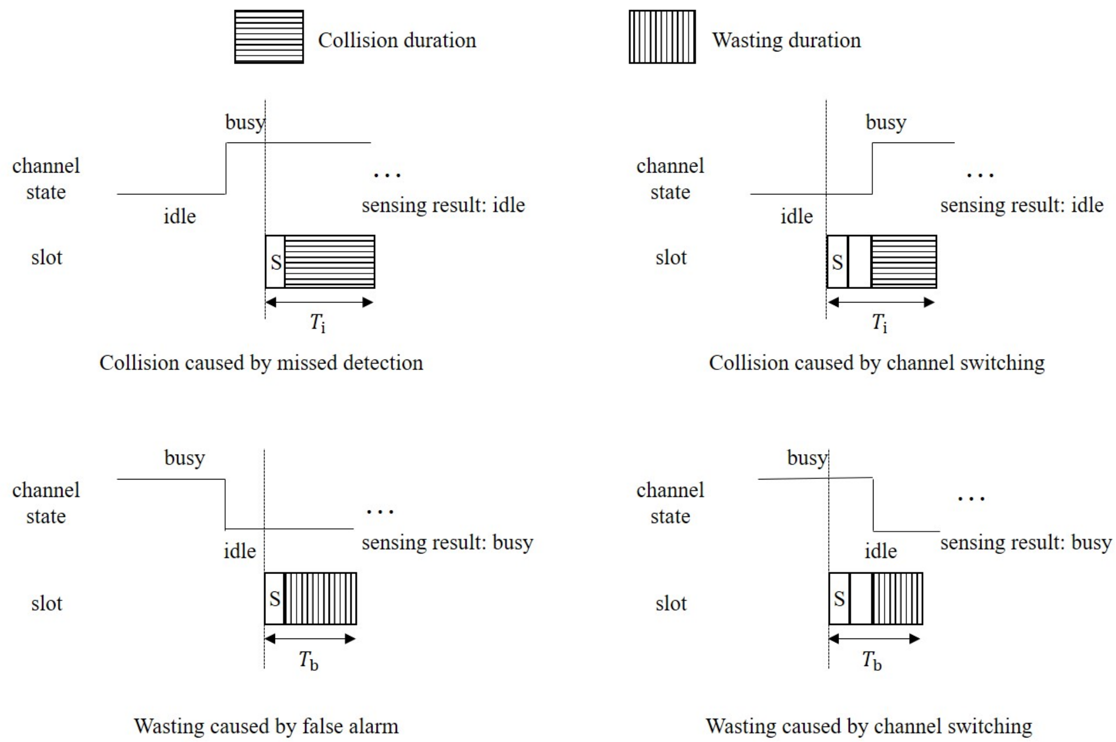

Figure 3.4: Examples of collision and wasting.

Fig. 3.5, where x is the remaining time duration to the next channel switching. If the channel state is idle at moment t_s, variable x denotes the sojourn time within which the channel state keeps idle starting from moment t_s, and thus, the channel state switches from idle to busy at moment $(t_\mathrm{s} + x)$. In this case, the PDF of variable x is expressed as [42]

$$g_\mathrm{i}(x) = \frac{\tilde{F}_\mathrm{i}(x)}{\tau_\mathrm{i}} \tag{3.6}$$

where τ_i is the expected sojourn time of a channel idle state, $\tilde{F}_\mathrm{i}(x) = 1 - F_\mathrm{i}(x)$, and $F_\mathrm{i}(x)$ denotes the cumulative distribution function (CDF) of the sojourn time of an idle state. Similarly, if the channel state is busy at moment t_s, variable x denotes the sojourn time within which the channel state keeps busy starting from moment t_s. Then, the PDF of variable x is expressed as

$$g_\mathrm{b}(x) = \frac{\tilde{F}_\mathrm{b}(x)}{\tau_\mathrm{b}} \tag{3.7}$$

where τ_b is the expected sojourn time of a channel busy state, $\tilde{F}_\mathrm{b}(x) = 1 - F_\mathrm{b}(x)$, and $F_\mathrm{b}(x)$ denotes the CDF of the sojourn time of a busy state.

If the channel state switches from busy to idle at moment t_s, x is sojourn time of

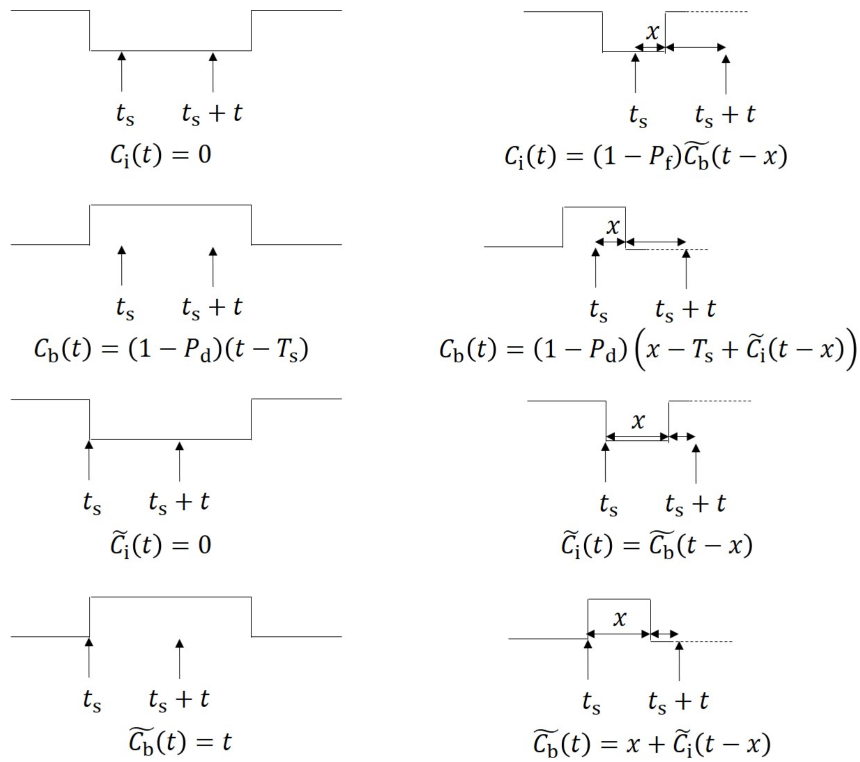

Figure 3.5: Illustration about $C_{\rm i}(t)$, $C_{\rm b}(t)$, $\tilde{C}_{\rm i}(t)$, and $\tilde{C}_{\rm b}(t)$.

the idle state, and thus, the PDF of variable x is $f_{\rm i}(x)$. Similarly, if the channel state switches from idle to busy at moment $t_{\rm s}$, x is the sojourn time of the busy state, and thus, the PDF of variable x is $f_{\rm b}(x)$.

Then, we have the following equations for $C_{\rm i}(t)$, $C_{\rm b}(t)$, $\tilde{C}_{\rm i}(t)$, and $\tilde{C}_{\rm b}(t)$ based on Fig. 3.5.

$$C_{\rm i}(t) = (1 - P_{\rm f}) \int_0^t \tilde{C}_{\rm b}\,(t - x)\,\frac{\tilde{F}_{\rm i}(x)}{\tau_{\rm i}} dx, \tag{3.8}$$

$$\begin{aligned} C_{\rm b}\,(t) \;&= (1 - P_{\rm d}) \textstyle\int_t^{\infty} (t - T_{\rm s}) \frac{\tilde{F}_{\rm b}(x)}{\tau_{\rm b}} dx + (1 - P_{\rm d}) \int_0^t \frac{\tilde{F}_{\rm b}(x)}{\tau_{\rm b}} (x - T_{\rm s} + \tilde{C}_{\rm i}\,(t - x))dx \\ &= (1 - P_{\rm d})\,(t \textstyle\int_t^{\infty} \frac{\tilde{F}_{\rm b}(x)}{\tau_{\rm b}} dx + \int_0^t \frac{\tilde{F}_{\rm b}(x)}{\tau_{\rm b}} (x + \tilde{C}_{\rm i}\,(t - x))dx - T_{\rm s}), \end{aligned} \tag{3.9}$$

$$\tilde{C}_{\rm i}\,(t) = \int_0^t f_{\rm i}\,(x)\,\tilde{C}_{\rm b}\,(t - x)dx, \tag{3.10}$$

$$\tilde{C}_{\rm b}\,(t) = t \int_t^{\infty} f_{\rm b}\,(x)dx + \int_0^t f_{\rm b}\,(x)\,(x + \tilde{C}_{\rm i}\,(t - x))dx. \tag{3.11}$$

Then, we perform the Laplace transform [72] on (3.8)-(3.11), and we have

$$C_{\mathrm{i}}^{*}(s)=(1-P_{\mathrm{f}})\frac{\tilde{F}_{\mathrm{i}}^{*}(s)\tilde{C}_{\mathrm{b}}^{*}(s)}{\tau_{\mathrm{i}}}, \tag{3.12}$$

$$C_{\mathrm{b}}^{*}(s)=(1-P_{\mathrm{d}})[\frac{\tilde{F}_{\mathrm{b}}^{*}(0)-\tilde{F}_{\mathrm{b}}^{*}(s)}{\tau_{\mathrm{b}}s^{2}}+\frac{\tilde{F}_{\mathrm{b}}^{*}(s)\tilde{C}_{\mathrm{i}}^{*}(s)}{\tau_{\mathrm{b}}}-\frac{T_{\mathrm{s}}}{s}], \tag{3.13}$$

$$\tilde{C}_{\mathrm{i}}^{*}(s)=f_{\mathrm{i}}^{*}(s)\tilde{C}_{\mathrm{b}}^{*}(s), \tag{3.14}$$

$$\tilde{C}_{\mathrm{b}}^{*}(s)=\frac{f_{\mathrm{b}}^{*}(0)-f_{\mathrm{b}}^{*}(s)}{s^{2}}+f_{\mathrm{b}}^{*}(s)\tilde{C}_{\mathrm{i}}^{*}(s), \tag{3.15}$$

where $X^{*}(s)$ is the Laplace transform of $X(t)$, $X\in\{C_{\mathrm{i}},C_{\mathrm{b}},\tilde{C}_{\mathrm{i}},\tilde{C}_{\mathrm{b}},\tilde{F}_{\mathrm{i}},\tilde{F}_{\mathrm{b}},f_{\mathrm{i}},f_{\mathrm{b}}\}$.

According to (3.14) and (3.15), $\tilde{C}_{\mathrm{i}}^{*}(s)$ and $\tilde{C}_{\mathrm{b}}^{*}(s)$ can be expressed as

$$\tilde{C}_{\mathrm{i}}^{*}(s)=\frac{f_{\mathrm{i}}^{*}(s)[f_{\mathrm{b}}^{*}(0)-f_{\mathrm{b}}^{*}(s)]}{s^{2}[1-f_{\mathrm{b}}^{*}(s)f_{\mathrm{i}}^{*}(s)]}, \tag{3.16}$$

$$\tilde{C}_{\mathrm{b}}^{*}(s)=\frac{f_{\mathrm{b}}^{*}(0)-f_{\mathrm{b}}^{*}(s)}{s^{2}[1-f_{\mathrm{b}}^{*}(s)f_{\mathrm{i}}^{*}(s)]}. \tag{3.17}$$

Accordingly, $C_{\mathrm{i}}^{*}(s)$ and $C_{\mathrm{b}}^{*}(s)$ are given as

$$C_{\mathrm{i}}^{*}\left(s\right)=\left(1-P_{\mathrm{f}}\right)\frac{\tilde{F}_{\mathrm{i}}^{*}\left(s\right)}{\tau_{\mathrm{i}}}\frac{f_{\mathrm{b}}^{*}\left(0\right)-f_{\mathrm{b}}^{*}\left(s\right)}{1-f_{\mathrm{i}}^{*}\left(s\right)f_{\mathrm{b}}^{*}\left(s\right)}\frac{1}{s^{2}}, \tag{3.18}$$

$$C_{\mathrm{b}}^{*}\left(s\right)=\left(1-P_{\mathrm{d}}\right)\left(\frac{1}{\tau_{\mathrm{b}}s^{2}}\left[\tilde{F}_{\mathrm{b}}^{*}\left(0\right)-\tilde{F}_{\mathrm{b}}^{*}\left(s\right)\frac{1-f_{\mathrm{i}}^{*}\left(s\right)f_{\mathrm{b}}^{*}\left(0\right)}{1-f_{\mathrm{b}}^{*}\left(s\right)f_{\mathrm{i}}^{*}\left(s\right)}\right]-\frac{T_{\mathrm{s}}}{s}\right). \tag{3.19}$$

Similarly, let $W_{\mathrm{i}}(t)$ and $W_{\mathrm{b}}(t)$ denote the expected wasting duration within a period $(t_{\mathrm{s}},t_{\mathrm{s}}+t)$ if the channel state at moment t_{s} is idle and busy, respectively. We also introduce $\tilde{W}_{\mathrm{i}}(t)$ and $\tilde{W}_{\mathrm{b}}(t)$ to denote the expected channel idle duration within a period $(t_{\mathrm{s}},t_{\mathrm{s}}+t)$ if the channel state switches from busy to idle and from idle to busy, respectively, at moment t_{s}. According to Fig. 3.4, an illustration about $W_{\mathrm{i}}(t)$, $W_{\mathrm{b}}(t)$, $\tilde{W}_{\mathrm{i}}(t)$, and $\tilde{W}_{\mathrm{b}}(t)$ is given in Fig. 3.6. Note that a secondary transmission cannot start at the spectrum sensing period. Accordingly, the spectrum sensing period is also viewed as a kind of wasting if the actual channel state is idle. Then, we have the following equations for $W_{\mathrm{i}}(t)$, $W_{\mathrm{b}}(t)$, $\tilde{W}_{\mathrm{i}}(t)$, and $\tilde{W}_{\mathrm{b}}(t)$.

$$W_{\mathrm{i}}\left(t\right)=P_{\mathrm{f}}\left(t\int_{t}^{\infty}\frac{\tilde{F}_{\mathrm{i}}\left(x\right)}{\tau_{\mathrm{i}}}dx+\int_{0}^{t}\frac{\tilde{F}_{\mathrm{i}}\left(x\right)}{\tau_{\mathrm{i}}}\left(x+\tilde{W}_{\mathrm{b}}\left(t-x\right)\right)dx\right)+(1-P_{\mathrm{f}})T_{\mathrm{s}}, \tag{3.20}$$

$$W_{\mathrm{b}}(t) = P_{\mathrm{d}} \int_0^t \tilde{W}_{\mathrm{i}}(t-x) \frac{\tilde{F}_{\mathrm{b}}(x)}{\tau_{\mathrm{b}}} dx, \tag{3.21}$$

$$\tilde{W}_{\mathrm{i}}(t) = t \int_t^\infty f_{\mathrm{i}}(x) dx + \int_0^t f_{\mathrm{i}}(x) \left(x + \tilde{W}_{\mathrm{b}}(t-x) \right) dx, \tag{3.22}$$

$$\tilde{W}_{\mathrm{b}}(t) = \int_0^t f_{\mathrm{b}}(x) \tilde{W}_{\mathrm{i}}(t-x) dx. \tag{3.23}$$

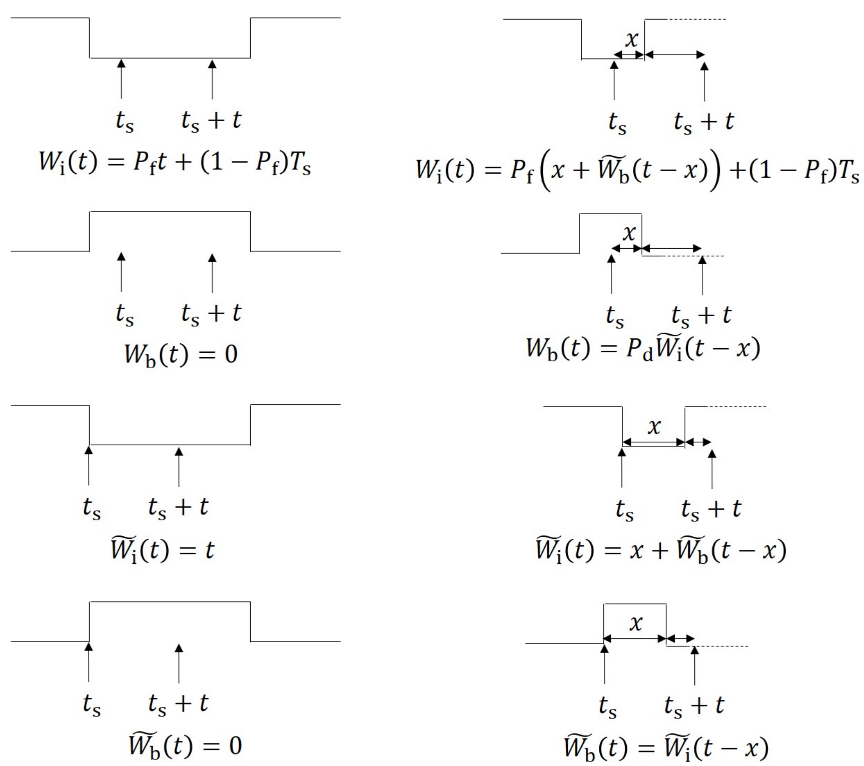

Figure 3.6: Illustration about $W_{\mathrm{i}}(t)$, $W_{\mathrm{b}}(t)$, $\tilde{W}_{\mathrm{i}}(t)$, and $\tilde{W}_{\mathrm{b}}(t)$.

Similar to the procedure of deriving $C_{\mathrm{i}}^*(s)$ and $C_{\mathrm{b}}^*(s)$, $W_{\mathrm{i}}^*(s)$ and $W_{\mathrm{b}}^*(s)$, which are Laplace transform of $W_{\mathrm{i}}(t)$ and $W_{\mathrm{b}}(t)$, respectively, can be expressed as

$$W_{\mathrm{i}}^*(s) = P_{\mathrm{f}} \frac{1}{\tau_{\mathrm{i}} s^2} \left[\tilde{F}_{\mathrm{i}}^*(0) - \tilde{F}_{\mathrm{i}}^*(s) \frac{1 - f_{\mathrm{b}}^*(s) f_{\mathrm{i}}^*(0)}{1 - f_{\mathrm{i}}^*(s) f_{\mathrm{b}}^*(s)} \right] + \frac{(1-P_{\mathrm{f}})T_{\mathrm{s}}}{s}, \tag{3.24}$$

$$W_{\mathrm{b}}^*(s) = P_{\mathrm{d}} \frac{\tilde{F}_{\mathrm{b}}^*(s)}{\tau_{\mathrm{b}}} \frac{f_{\mathrm{i}}^*(0) - f_{\mathrm{i}}^*(s)}{1 - f_{\mathrm{b}}^*(s) f_{\mathrm{i}}^*(s)} \frac{1}{s^2}. \tag{3.25}$$

Then, given the expression of $f_{\mathrm{i}}(x)$ and $f_{\mathrm{b}}(x)$, the exact expression of $C_{\mathrm{i}}(t)$, $C_{\mathrm{b}}(t)$, $W_{\mathrm{i}}(t)$ and $W_{\mathrm{b}}(t)$ can be obtained by performing the inverse Laplace transform on (3.18), (3.19), (3.24), and (3.25).

3.4.2 Analysis of $E[R_{\text{slot}}]$

If the sensing result is idle in a time slot, a secondary transmission is started by the secondary user. Due to possible missed detection event or possible channel state switching, a primary transmission may exist within the time slot although the sensing result is idle. For a secondary transmission, if the actual channel state is idle, the achievable data rate of the secondary transmission is

$$\text{rate1} = E\left[\log_2\left(1+\gamma_{\text{s}}\right)\right], \tag{3.26}$$

where γ_{s} is the SNR of the secondary signal received by the secondary receiver [73]. However, if the actual channel state is busy, the achievable data rate of the secondary transmission is

$$\text{rate2} = E\left[\log_2\left(1+\frac{\gamma_{\text{s}}}{\gamma_{\text{p}}+1}\right)\right]. \tag{3.27}$$

Within a slot, let T_{rate1} and T_{rate2} denote the expected time duration with transmission rate rate1 and rate2, respectively. To derive the average throughput of secondary transmission, we need to derive T_{rate1} and T_{rate2}. Let $T_{\text{i-slot}}$ and $T_{\text{b-slot}}$ denote the expected length of an i-slot and a b-slot, respectively. Due to possible missed detection event and false alarm event, we have

$$T_{\text{i-slot}} = P_{\text{f}}T_{\text{b}} + (1-P_{\text{f}})T_{\text{i}}, \tag{3.28}$$

$$T_{\text{b-slot}} = P_{\text{d}}T_{\text{b}} + (1-P_{\text{d}})T_{\text{i}}. \tag{3.29}$$

Denote $p_{\text{i-slot}}$ and $p_{\text{b-slot}}$ as the probability that a slot belongs to *i-slot* and *b-slot*, respectively. Then, we have

$$p_{\text{i-slot}} = \frac{\tau_{\text{i}}}{T_{\text{i-slot}}}\Bigg/\left(\frac{\tau_{\text{i}}}{T_{\text{i-slot}}}+\frac{\tau_{\text{b}}}{T_{\text{b-slot}}}\right), \tag{3.30}$$

$$p_{\text{b-slot}} = \frac{\tau_{\text{b}}}{T_{\text{b-slot}}}\Bigg/\left(\frac{\tau_{\text{i}}}{T_{\text{i-slot}}}+\frac{\tau_{\text{b}}}{T_{\text{b-slot}}}\right). \tag{3.31}$$

Therefore, the expected time duration of a slot, denoted T_{slot}, should be

$$T_{\text{slot}} = p_{\text{i-slot}}T_{\text{i-slot}} + p_{\text{b-slot}}T_{\text{b-slot}}. \tag{3.32}$$

For an i-slot, the expected channel busy duration is given as

$$C_{\rm i}(T_{\rm i}) + \frac{P_{\rm f}C_{\rm i}(T_{\rm b})}{1-P_{\rm f}}, \tag{3.33}$$

and the expected channel idle duration is given as

$$W_{\rm i}(T_{\rm b}) + \frac{(1-P_{\rm f})W_{\rm i}(T_{\rm i})}{P_{\rm f}} - \frac{(1-P_{\rm f})T_{\rm s}}{P_{\rm f}}. \tag{3.34}$$

For a b-slot, the expected channel busy duration is given as

$$C_{\rm b}(T_{\rm i}) + \frac{P_{\rm d}C_{\rm b}(T_{\rm b})}{(1-P_{\rm d})} + T_{\rm s} \tag{3.35}$$

and the expected channel idle duration is given as

$$W_{\rm b}(T_{\rm b}) + \frac{(1-P_{\rm d})W_{\rm b}(T_{\rm i})}{P_{\rm d}}. \tag{3.36}$$

Therefore, according to (3.34) and (3.36), the expected channel idle duration in a slot, denoted $T_{\rm i-in-slot}$, is given as

$$\begin{aligned} T_{\rm i-in-slot} &= p_{\rm i-slot}\left\{W_{\rm i}(T_{\rm b}) + \tfrac{(1-P_{\rm f})W_{\rm i}(T_{\rm i})}{P_{\rm f}} - \tfrac{(1-P_{\rm f})T_{\rm s}}{P_{\rm f}}\right\} \\ &\quad + p_{\rm b-slot}\left\{W_{\rm b}(T_{\rm b}) + \tfrac{(1-P_{\rm d})W_{\rm b}(T_{\rm i})}{P_{\rm d}}\right\}. \end{aligned} \tag{3.37}$$

According to (3.33) and (3.35), the expected channel busy duration in a slot, denoted $T_{\rm b-in-slot}$, is given as

$$\begin{aligned} T_{\rm b-in-slot} &= p_{\rm i-slot}\left\{C_{\rm i}(T_{\rm i}) + \tfrac{P_{\rm f}C_{\rm i}(T_{\rm b})}{1-P_{\rm f}}\right\} \\ &\quad + p_{\rm b-slot}\left\{C_{\rm b}(T_{\rm i}) + \tfrac{P_{\rm d}C_{\rm b}(T_{\rm b})}{(1-P_{\rm d})} + T_{\rm s}\right\}. \end{aligned} \tag{3.38}$$

Let $T_{\rm c-in-slot}$ and $T_{\rm w-in-slot}$ denote the expected collision duration and wasting duration, respectively, in a slot. We have

$$T_{\rm c-in-slot} = p_{\rm i-slot}C_{\rm i}(T_{\rm i}) + p_{\rm b-slot}C_{\rm b}(T_{\rm i}), \tag{3.39}$$

$$T_{\rm w-in-slot} = p_{\rm i-slot}W_{\rm i}(T_{\rm b}) + p_{\rm b-slot}W_{\rm b}(T_{\rm b}). \tag{3.40}$$

Accordingly, we can get the expression of $T_{\rm rate1}$ and $T_{\rm rate2}$ as follows.

$$T_{\rm rate1} = T_{\rm i-in-slot} - T_{\rm w-in-slot}, \tag{3.41}$$

$$T_{\rm rate2} = T_{\rm c-in-slot}. \tag{3.42}$$

Therefore, the expected throughput of secondary transmissions should be

$$E[R_{\mathrm{slot}}] = \frac{\mathrm{rate1} \times T_{\mathrm{rate1}} + \mathrm{rate2} \times T_{\mathrm{rate2}}}{T_{\mathrm{slot}}} \tag{3.43}$$

where rate1, rate2, T_{rate1}, T_{rate2}, and T_{slot} are given in (3.26), (3.27), (3.41), (3.42), and (3.32), respectively.

3.4.3 Analysis of η_{c} and η_{s}

According to the definition of η_{c} in (3.3), the constraint (3.5b) should be

$$\frac{T_{\mathrm{c-in-slot}}}{T_{\mathrm{b-in-slot}}} \leq \varepsilon_{\mathrm{c}} \tag{3.44}$$

where $T_{\mathrm{c-in-slot}}$ and $T_{\mathrm{b-in-slot}}$ are given in (3.39) and (3.38), respectively.

According to the definition of η_{s} in (3.4), the constraint (3.5c) should be

$$\frac{T_{\mathrm{s}}}{T_{\mathrm{slot}}} \leq \varepsilon_{\mathrm{s}} \tag{3.45}$$

where T_{slot} is given in (3.32).

3.4.4 Optimal Configuration of T_{i} and T_{b}

According to some practical measurements [68]- [75], we assume that the sojourn time of channel idle and busy state follow exponential distribution with parameter λ_{i} and λ_{b}, respectively. Thus, we have $\tau_{\mathrm{i}} = 1/\lambda_{\mathrm{i}}$ and $\tau_{\mathrm{b}} = 1/\lambda_{\mathrm{b}}$, and $f_{\mathrm{i}}(x)$ and $f_{\mathrm{b}}(x)$ are given as

$$f_{\mathrm{i}}(x) = \lambda_{\mathrm{i}} e^{-\lambda_{\mathrm{i}} x}, x > 0, \tag{3.46}$$

$$f_{\mathrm{b}}(x) = \lambda_{\mathrm{b}} e^{-\lambda_{\mathrm{b}} x}, x > 0. \tag{3.47}$$

By performing the inverse Laplace transform on (3.18), (3.19), (3.24), and (3.25), we have

$$C_{\mathrm{i}}(t) = \frac{\lambda_{\mathrm{i}}(1 - P_{\mathrm{f}})}{(\lambda_{\mathrm{i}} + \lambda_{\mathrm{b}})^2}[(\lambda_{\mathrm{i}} + \lambda_{\mathrm{b}})t + e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})t} - 1], \tag{3.48}$$

$$C_{\mathrm{b}}(t) = \frac{\lambda_{\mathrm{b}}(1 - P_{\mathrm{d}})}{(\lambda_{\mathrm{i}} + \lambda_{\mathrm{b}})^2}[\frac{\lambda_{\mathrm{i}}(\lambda_{\mathrm{i}} + \lambda_{\mathrm{b}})t}{\lambda_{\mathrm{b}}} - e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})t} + 1] - (1 - P_{\mathrm{d}})T_{\mathrm{s}}, \tag{3.49}$$

$$W_{\mathrm{i}}(t)=\frac{\lambda_{\mathrm{i}}P_{\mathrm{f}}}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})^2}[\frac{\lambda_{\mathrm{b}}(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})t}{\lambda_{\mathrm{i}}}-e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})t}+1]+(1-P_{\mathrm{f}})T_{\mathrm{s}}, \tag{3.50}$$

$$W_{\mathrm{b}}(t)=\frac{\lambda_{\mathrm{b}}P_{\mathrm{d}}}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})^2}[(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})t+e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})t}-1]. \tag{3.51}$$

For the sensing ratio η_{s}, we introduce the following lemma.

Lemma 1. *Given a value of T_{b}, the sensing ratio η_{s} is a monotonically decreasing function of T_{i}. Given a value of T_{i}, the sensing ratio η_{s} is a monotonically decreasing function of T_{b}.*

Proof. According to (3.32), T_{slot} is given as

$$T_{\mathrm{slot}}=\frac{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i-slot}}T_{\mathrm{b-slot}}}{\lambda_{\mathrm{i}}T_{\mathrm{i-slot}}+\lambda_{\mathrm{b}}T_{\mathrm{b-slot}}}.$$

After taking the first-order derivative of T_{slot} with respect to T_{i}, we have

$$\begin{aligned}\frac{\partial T_{\mathrm{slot}}}{\partial T_{\mathrm{i}}}=&\frac{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})\lambda_{\mathrm{i}}(1-P_{\mathrm{d}})(P_{\mathrm{f}}T_{\mathrm{b}}+(1-P_{\mathrm{f}})T_{\mathrm{i}})^2}{[\lambda_{\mathrm{i}}(P_{\mathrm{f}}T_{\mathrm{b}}+(1-P_{\mathrm{f}})T_{\mathrm{i}})+\lambda_{\mathrm{b}}(P_{\mathrm{d}}T_{\mathrm{b}}+(1-P_{\mathrm{d}})T_{\mathrm{i}})]^2}\\&+\frac{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})\lambda_{\mathrm{b}}(1-P_{\mathrm{f}})(P_{\mathrm{d}}T_{\mathrm{b}}+(1-P_{\mathrm{d}})T_{\mathrm{i}})^2}{[\lambda_{\mathrm{i}}(P_{\mathrm{f}}T_{\mathrm{b}}+(1-P_{\mathrm{f}})T_{\mathrm{i}})+\lambda_{\mathrm{b}}(P_{\mathrm{d}}T_{\mathrm{b}}+(1-P_{\mathrm{d}})T_{\mathrm{i}})]^2}>0.\end{aligned}$$

Accordingly, T_{slot} is increasing monotonically with T_{i} when T_{b} is given. Due to $\eta_{\mathrm{s}}=\frac{T_{\mathrm{s}}}{T_{\mathrm{slot}}}$, η_{s} is monotonically decreasing with T_{i} for a given T_{b}.

Then, we take the first-order derivative of T_{slot} with respect to T_{b}, and we have

$$\begin{aligned}\frac{\partial T_{\mathrm{slot}}}{\partial T_{\mathrm{b}}}=&\frac{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})\lambda_{\mathrm{i}}P_{\mathrm{d}}(P_{\mathrm{f}}T_{\mathrm{b}}+(1-P_{\mathrm{f}})T_{\mathrm{i}})^2}{[\lambda_{\mathrm{i}}(P_{\mathrm{f}}T_{\mathrm{b}}+(1-P_{\mathrm{f}})T_{\mathrm{i}})+\lambda_{\mathrm{b}}(P_{\mathrm{d}}T_{\mathrm{b}}+(1-P_{\mathrm{d}})T_{\mathrm{i}})]^2}\\&+\frac{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})\lambda_{\mathrm{b}}P_{\mathrm{f}}(P_{\mathrm{d}}T_{\mathrm{b}}+(1-P_{\mathrm{d}})T_{\mathrm{i}})^2}{[\lambda_{\mathrm{i}}(P_{\mathrm{f}}T_{\mathrm{b}}+(1-P_{\mathrm{f}})T_{\mathrm{i}})+\lambda_{\mathrm{b}}(P_{\mathrm{d}}T_{\mathrm{b}}+(1-P_{\mathrm{d}})T_{\mathrm{i}})]^2}>0.\end{aligned}$$

Therefore, T_{slot} is increasing monotonically with T_{b} for a given T_{i}. Due to $\eta_{\mathrm{s}}=\frac{T_{\mathrm{s}}}{T_{\mathrm{slot}}}$, η_{s} is monotonically decreasing with T_{b} for a given T_{i}. This completes the proof. □

Based on Lemma 1, we have the following remark.

Remark 1. *To satisfy constraint (3.5c), the value of T_{i} should satisfy $T_{\mathrm{i}}\geq\varphi_1(T_{\mathrm{b}})$ for a given T_{b}, where $\varphi_1(T_{\mathrm{b}})$ is the solution of $\eta_{\mathrm{s}}=\varepsilon_{\mathrm{s}}$. We have $\varphi_1(T_{\mathrm{b}})=\frac{-V_2+\sqrt{(V_2)^2-4V_1V_3}}{2V_1}$ where $V_1=\frac{\varepsilon_{\mathrm{s}}}{T_{\mathrm{s}}}(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})(1-P_{\mathrm{d}})(1-P_{\mathrm{f}})$, $V_2=\frac{\varepsilon_{\mathrm{s}}}{T_{\mathrm{s}}}(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})[P_{\mathrm{d}}(1-P_{\mathrm{f}})+P_{\mathrm{f}}(1-P_{\mathrm{d}})]T_{\mathrm{b}}-[\lambda_{\mathrm{i}}(1-P_{\mathrm{f}})+\lambda_{\mathrm{b}}(1-P_{\mathrm{d}})]$, and $V_3=\frac{\varepsilon_{\mathrm{s}}}{T_{\mathrm{s}}}(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})P_{\mathrm{d}}P_{\mathrm{f}}(T_{\mathrm{b}})^2-(\lambda_{\mathrm{i}}P_{\mathrm{f}}+\lambda_{\mathrm{b}}P_{\mathrm{d}})T_{\mathrm{b}}$. Accordingly, the feasible set of T_{i} is given as $\mathbf{k}_{T_{\mathrm{i}}}=[\max\{T_{\mathrm{s}},\varphi_1(T_{\mathrm{b}})\},\tau_{\mathrm{i}}]$. In other words, by using Lemma 1, we can shrink the feasible set of T_{i}.*

Based on Remark 1, we propose an algorithm, called Algorithm 3.1, to find T_{i}^* and T_{b}^* (the optimal T_{i} and T_{b}). Intuitively, we also know that T_{i} and T_{b} should be not more than the expected sojourn time of a channel idle state (τ_{i}) and the expected sojourn time of a channel busy state (τ_{b}), respectively.

Algorithm 3.1 The proposed algorithm to find T_{i}^* and T_{b}^*

1: Set initial values $T_{\mathrm{i}}^{\#} = 0$, $T_{\mathrm{b}}^{\#} = 0$, and $R^{\#} = 0$.
2: **for** $T_{\mathrm{b}} \leftarrow T_{\mathrm{s}}$ to τ_{b} **do**
3:     Calculate $\varphi_1(T_{\mathrm{b}})$.
4:     Get the feasible set of T_{i}, denoted $\mathbf{k}_{T_{\mathrm{i}}} = [\max\{T_{\mathrm{s}}, \varphi_1(T_{\mathrm{b}})\}, \tau_{\mathrm{i}}]$.
5:     Calculate $\hat{T}_{\mathrm{i}} = \arg\max_{T_{\mathrm{i}} \in \mathbf{k}_{T_{\mathrm{i}}}} \{E[R_{\mathrm{slot}}]\}$ and $\hat{R} = \max_{T_{\mathrm{i}} \in \mathbf{k}_{T_{\mathrm{i}}}} \{E[R_{\mathrm{slot}}]\}$.
6:     **if** $\hat{R} > R^{\#}$ **then**
7:         Update $T_{\mathrm{i}}^{\#} = \hat{T}_{\mathrm{i}}$ and $T_{\mathrm{b}}^{\#} = T_{\mathrm{b}}$.
8:         Update $R^{\#} = \hat{R}$.
9:     **end if**
10: **end for**
11: Set $T_{\mathrm{i}}^* = T_{\mathrm{i}}^{\#}$ and $T_{\mathrm{b}}^* = T_{\mathrm{b}}^{\#}$.

In Algorithm 3.1, $T_{\mathrm{i}}^{\#}$, $T_{\mathrm{b}}^{\#}$ and $R^{\#}$ represent the current (temporary) optimal value of T_{i}, T_{b}, and $E[R_{\mathrm{slot}}]$, respectively. Steps 3-4 are to find the feasible set of T_{i} for the given value of T_{b}. Step 5 is to find the optimal T_{i} for the given value of T_{b}. Steps 6-9 update the value of $T_{\mathrm{i}}^{\#}$, $T_{\mathrm{b}}^{\#}$ and $R^{\#}$. The time complexity of Algorithm 3.1 is $\mathbf{O}(N_{\mathbf{k}_{\mathrm{T_i}}} N_{\mathbf{k}_{\mathrm{T_b}}})$, where $\mathbf{O}$ means big O notation, $N_{\mathbf{k}_{\mathrm{T_i}}}$ and $N_{\mathbf{k}_{\mathrm{T_b}}}$ denote the searched space[2] of $\mathbf{k}_{\mathrm{T_i}}$ and $\mathbf{k}_{\mathrm{T_b}}$, respectively. Since $N_{\mathbf{k}_{\mathrm{T_i}}}$ in Algorithm 3.1 is much smaller than the searched space of the initial feasible set of $\mathrm{T_i}$ (i.e., $N_{[T_{\mathrm{s}},\tau_{\mathrm{i}}]}$). Accordingly, the time complexity of the proposed algorithm is decreased.

3.4.5 Optimal configuration of T_{i} and T_{b} if spectrum sensing is perfect

With the development of spectrum sensing technology, the detection accuracy of spectrum sensing has been largely improved. If the detection probability P_{d} and the false alarm probability P_{f} approach 1 and 0 respectively, it can be regarded as perfect spectrum sensing [76]. In this subsection, we will derive the optimal value of T_{i} and T_{b} in the case with perfect spectrum sensing.[3]

We have the following lemma.

[2]If the searched step is ς, the searched space of a feasible set $[a, b]$ should be $\frac{b-a}{\varsigma}$.
[3]Perfect spectrum sensing has been assumed in numerous existing research efforts.

Lemma 2. *The average throughput $E[R_{\mathrm{slot}}]$ is independent of T_{b}. In addition, The average throughput $E[R_{\mathrm{slot}}]$ is monotonically increasing with T_{i} when $0 < T_{\mathrm{i}} \leq \varphi_2$ and monotonically decreasing with T_{i} when $\varphi_2 < T_{\mathrm{i}} < \infty$, where $\varphi_2 = \frac{-W(\frac{V_4}{e})-1}{\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}}}$, $W(\cdot)$ is the Lambert W-Function, e is the exponential constant, and $V_4 = \frac{rate1 \times T_{\mathrm{s}}(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})^2}{\lambda_{\mathrm{i}}(rate1-rate2)} - 1$.*

Proof. For T_{rate1}, we have

$$T_{\mathrm{rate1}} = p_{\mathrm{i-slot}} \frac{\lambda_{\mathrm{i}}}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})^2}\Big[\frac{\lambda_{\mathrm{b}}(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}}{\lambda_{\mathrm{i}}} - e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}} + 1\Big] - p_{\mathrm{i-slot}}T_{\mathrm{s}}$$

where $p_{\mathrm{i-slot}} = \frac{\lambda_{\mathrm{b}}T_{\mathrm{b}}}{\lambda_{\mathrm{i}}T_{\mathrm{i}}+\lambda_{\mathrm{b}}T_{\mathrm{b}}}$.

For T_{rate2}, we have

$$T_{\mathrm{rate2}} = p_{\mathrm{i-slot}} \frac{\lambda_{\mathrm{i}}}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})^2}[(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}} + e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}} - 1].$$

For T_{slot}, we have

$$T_{\mathrm{slot}} = \frac{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}T_{\mathrm{b}}}{\lambda_{\mathrm{i}}T_{\mathrm{i}}+\lambda_{\mathrm{b}}T_{\mathrm{b}}}.$$

According to (3.43), the expression of $E[R_{\mathrm{slot}}]$ is given as

$$\begin{aligned} E[R_{\mathrm{slot}}] &= \tfrac{\lambda_{\mathrm{i}}\lambda_{\mathrm{b}}}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})^3T_{\mathrm{i}}}\{\mathrm{rate1} \times [\tfrac{\lambda_{\mathrm{b}}(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}}{\lambda_{\mathrm{i}}} - e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}} + 1] \\ &\quad +\mathrm{rate2} \times [(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}} + e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}} - 1]\} \\ &\quad -\mathrm{rate1} \times \tfrac{\lambda_{\mathrm{b}}T_{\mathrm{s}}}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}}, \end{aligned}$$

from which it can be seen that $E[R_{\mathrm{slot}}]$ is independent of T_{b}. Then, we take the first-order derivative of $E[R_{\mathrm{slot}}]$ with respect to T_{i}, and we have

$$\begin{aligned} \tfrac{dE[R_{\mathrm{slot}}]}{dT_{\mathrm{i}}} &= \tfrac{\lambda_{\mathrm{i}}\lambda_{\mathrm{b}}(\mathrm{rate1}-\mathrm{rate2})}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})^3(T_{\mathrm{i}})^2} \\ &\quad \times \left[e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}} + (\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}e^{-(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})T_{\mathrm{i}}} - 1\right] \\ &\quad + \tfrac{\mathrm{rate1}\times\lambda_{\mathrm{b}}T_{\mathrm{s}}}{(\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}})(T_{\mathrm{i}})^2}. \end{aligned}$$

For equation $\frac{dE[R_{\mathrm{slot}}]}{dT_{\mathrm{i}}} = 0$, the root is given as $T_{\mathrm{i}} = \varphi_2 = \frac{-W(\frac{V_4}{e})-1}{\lambda_{\mathrm{i}}+\lambda_{\mathrm{b}}}$. Accordingly, we have

$$\begin{cases} \frac{dE[R_{\mathrm{slot}}]}{dT_{\mathrm{i}}} > 0, & 0 < T_{\mathrm{i}} < \varphi_2; \\ \frac{dE[R_{\mathrm{slot}}]}{dT_{\mathrm{i}}} = 0, & T_{\mathrm{i}} = \varphi_2; \\ \frac{dE[R_{\mathrm{slot}}]}{dT_{\mathrm{i}}} < 0, & \varphi_2 < T_{\mathrm{i}} < \infty. \end{cases}$$

This completes the proof. □

For the collision ratio η_{c}, we introduce the following lemma.

Lemma 3. *Given a value of T_{b}, the collision ratio η_{c} is monotonically increasing with T_{i}. Given a value of T_{i}, the collision ratio η_{c} is monotonically increasing with T_{b}.*

Proof. For $T_{\text{c-in-slot}}$, we have

$$T_{\text{c-in-slot}} = p_{\text{i-slot}} \frac{\lambda_{\text{i}}}{(\lambda_{\text{i}} + \lambda_{\text{b}})^2} [(\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} + e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} - 1].$$

For $T_{\text{b-in-slot}}$, we have

$$\begin{aligned} T_{\text{b-in-slot}} &= p_{\text{i-slot}} \tfrac{\lambda_{\text{i}}}{(\lambda_{\text{i}}+\lambda_{\text{b}})^2} [(\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} + e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} - 1] \\ &+ p_{\text{b-slot}} \tfrac{\lambda_{\text{b}}}{(\lambda_{\text{i}}+\lambda_{\text{b}})^2} [\tfrac{\lambda_{\text{i}}(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}}{\lambda_{\text{b}}} - e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} + 1] \end{aligned}$$

where $p_{\text{b-slot}} = \frac{\lambda_{\text{i}} T_{\text{i}}}{\lambda_{\text{i}} T_{\text{i}} + \lambda_{\text{b}} T_{\text{b}}}$.

According to (3.3), the expression of η_{c} is given as

$$\eta_{\text{c}} = \frac{1}{1 + V_5}$$

where $V_5 = \frac{T_{\text{i}} \left[\frac{\lambda_{\text{i}}(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}}{\lambda_{\text{b}}} - e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} + 1 \right]}{T_{\text{b}} \left[(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}} + e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} - 1 \right]}$.

Then, we take the first-order derivative of V_5 with respect to T_{i}, and we have

$$\frac{\partial V_5}{\partial T_{\text{i}}} = \left[\frac{\lambda_{\text{i}}(\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{b}}}{\lambda_{\text{b}}} - e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} + 1 \right] \times \frac{\left[e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} + (\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} - 1 \right]}{T_{\text{b}} \left[(\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} + e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} - 1 \right]^2}.$$

Thus, we have $\frac{\partial V_5}{\partial T_{\text{i}}} < 0$, due to the facts $e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} < 1$ and $e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} + (\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} < 1$ when $T_{\text{i}} > 0$. Accordingly, given a value of T_{b}, η_{c} is monotonically increasing with T_{i}.

Similarly, the first-order derivative of V_5 with respect to T_{b} is given as

$$\frac{\partial V_5}{\partial T_{\text{b}}} = \left[(\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} + e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} - 1 \right] T_{\text{i}} \times \frac{\left[e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} + (\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{b}} e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} - 1 \right]}{(T_{\text{b}})^2 \left[(\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} + e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} - 1 \right]^2}.$$

Thus, we have $\frac{\partial V_5}{\partial T_{\text{b}}} < 0$, due to the facts $(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}} + e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}}} > 1$ and $e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} + (\lambda_{\text{i}} + \lambda_{\text{b}})T_{\text{i}} e^{-(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}}} < 1$ when $T_{\text{i}} > 0$. Accordingly, given a value of T_{i}, η_{c} is monotonically increasing with T_{b}. This completes the proof. □

Based on Lemma 1, we have the following remark.

Remark 2. *To satisfy constraint (3.5c), the value of* T_{i} *should satisfy* $T_{\text{i}} \geq \varphi_3(T_{\text{b}})$ *for a given* T_{b}*, where* $\varphi_3(T_{\text{b}}) = \frac{\lambda_{\text{b}} T_{\text{b}} T_{\text{s}}}{\varepsilon_{\text{s}}(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{b}} - T_{\text{s}}\lambda_{\text{i}}}$ *is the solution of* $\eta_{\text{s}} = \varepsilon_{\text{s}}$*. Similarly, to satisfy constraint (3.5c), the value of* T_{b} *should satisfy* $T_{\text{b}} \geq \varphi_4(T_{\text{i}})$ *for a given* T_{i}*, where* $\varphi_4(T_{\text{i}}) = \frac{\lambda_{\text{i}} T_{\text{i}} T_{\text{s}}}{\varepsilon_{\text{s}}(\lambda_{\text{i}}+\lambda_{\text{b}})T_{\text{i}} - T_{\text{s}}\lambda_{\text{b}}}$ *is the solution of* $\eta_{\text{s}} = \varepsilon_{\text{s}}$.

Based on Lemma 3, we have the following remark.

Remark 3. *To satisfy constraint (3.5b), the value of* T_i *should satisfy* $T_\mathrm{i} \leq \varphi_5(T_\mathrm{b})$ *for a given* T_b*, where* $\varphi_5(T_\mathrm{b})$ *is the solution of* $\eta_\mathrm{c} = \varepsilon_\mathrm{c}$*. Thus, the value of* $\varphi_5(T_\mathrm{b})$ *is derived as* $\varphi_5(T_\mathrm{b}) = \frac{1}{\lambda_\mathrm{i}+\lambda_\mathrm{b}} W(\frac{\lambda_\mathrm{i}+\lambda_\mathrm{b}}{V_6} e^{\frac{\lambda_\mathrm{i}+\lambda_\mathrm{b}}{V_6}}) - \frac{1}{V_6}$ *where* $V_6 = \left[\frac{\lambda_\mathrm{i}(\lambda_\mathrm{i}+\lambda_\mathrm{b})T_\mathrm{b}}{\lambda_\mathrm{b}} - e^{-(\lambda_\mathrm{i}+\lambda_\mathrm{b})T_\mathrm{b}} + 1\right] \frac{1}{T_\mathrm{b}} \frac{\varepsilon_\mathrm{c}}{1-\varepsilon_\mathrm{c}} - (\lambda_\mathrm{i}+\lambda_\mathrm{b})$*. To satisfy constraint (3.5b), the value of* T_b *should satisfy* $T_\mathrm{b} \leq \varphi_6(T_\mathrm{i})$ *for a given* T_i*, where* $\varphi_6(T_\mathrm{i})$ *is the solution of* $\eta_\mathrm{c} = \varepsilon_\mathrm{c}$*. Thus, the value of* $\varphi_6(T_\mathrm{i})$ *is derived as* $\varphi_6(T_\mathrm{b}) = \frac{1}{\lambda_\mathrm{i}+\lambda_\mathrm{b}} W(-\frac{\lambda_\mathrm{i}+\lambda_\mathrm{b}}{V_7} e^{-\frac{\lambda_\mathrm{i}+\lambda_\mathrm{b}}{V_7}}) + \frac{1}{V_7}$*, where* $V_7 = \left[(\lambda_\mathrm{i}+\lambda_\mathrm{b})T_\mathrm{i} + e^{-(\lambda_\mathrm{i}+\lambda_\mathrm{b})T_\mathrm{i}} - 1\right] \frac{1}{T_\mathrm{i}} \frac{1-\varepsilon_\mathrm{c}}{\varepsilon_\mathrm{c}} - \frac{\lambda_\mathrm{i}(\lambda_\mathrm{i}+\lambda_\mathrm{b})}{\lambda_\mathrm{b}}$.

According to Remark 2, Remark 3, and Lemma 2, we have the following remark for the optimal value of T_i and T_b.

Remark 4. *Considering* $T_\mathrm{i} = \varphi_2$*, we can obtain the feasible set of* T_b*, which is given as* $\mathbf{k}_{T_\mathrm{b}} = [T_\mathrm{s}, \min\{\varphi_6(T_\mathrm{i}), \tau_\mathrm{b}\}] \cap [\varphi_4(T_\mathrm{i}), \tau_\mathrm{b}]$*. Then, the optimal value of* T_i *and* T_b *can be obtained as follows.*

- *If* $\mathbf{k}_{T_\mathrm{b}} \neq \varnothing$ *(*$\varnothing$ *being an empty set), the optimal value of* T_i *is* $T_\mathrm{i}^* = \varphi_2$ *and the optimal value of* T_b *can be any value in the feasible set* $\mathbf{k}_{T_\mathrm{b}}$*. In other words, closed-form expression of the optimal value of* T_i *and* T_b *can be obtained. In other words, the time complexity is* $\mathbf{O}(1)$*.*

- *If* $\mathbf{k}_{T_\mathrm{b}} = \varnothing$*, the optimal value of* T_i *and* T_b *can be derived by Algorithm 3.2. In Algorithm 3.2, for each specific value of* T_b*, the optimal value of* T_i *can be found directly by step 5. Thus, we only need to search over the values of* T_b *in Algorithm 3.2. Accordingly, the time complexity of Algorithm 3.2 is* $\mathbf{O}(N_{[T_\mathrm{s},\tau_\mathrm{b}]})$*.*

3.5 Numerical Results

In this section, we will evaluate the performance of our proposed scheme by using simulation. In the simulation, the adopted parameters (unless otherwise specified) are given in Table 3.1.

It is desired to compare the performance of our proposed scheme with existing works. However, to our best knowledge, with a practical setting of imperfect spectrum sensing, no existing work considers the collision ratio of primary activities (as our constraint (3.5b) does). Recall that existing works consider a fixed slot length (i.e., they have the same slot length when the channel is sensed idle or busy). To

Algorithm 3.2 The proposed algorithm to obtain T_{i}^{*} and T_{b}^{*} with perfect spectrum sensing

Set initial values $T_{\mathrm{i}}^{\#}=0$, $T_{\mathrm{b}}^{\#}=0$, and $R^{\#}=0$.
for $T_{\mathrm{b}} \leftarrow T_{\mathrm{s}}$ to τ_{b} **do**
    Calculate $\varphi_3(T_{\mathrm{b}})$ and $\varphi_5(T_{\mathrm{b}})$.
    Get the feasible set of T_{i}, denoted $\mathbf{k}_{T_{\mathrm{i}}}=[T_{\mathrm{s}},\min\{\varphi_5(T_{\mathrm{b}}),\tau_{\mathrm{i}}\}]\cap[\varphi_3(T_{\mathrm{b}}),\tau_{\mathrm{i}}]$. Suppose $\mathbf{k}_{T_{\mathrm{i}}}$ is represented as $[T_{\mathrm{i}}^{\min},T_{\mathrm{i}}^{\max}]$.
    If $E[R_{\mathrm{slot}}]|_{T_{\mathrm{i}}=T_{\mathrm{i}}^{\min}} > E[R_{\mathrm{slot}}]|_{T_{\mathrm{i}}=T_{\mathrm{i}}^{\max}}$, then $\hat{T}_{\mathrm{i}}=T_{\mathrm{i}}^{\min}$ and $\hat{R}=E[R_{\mathrm{slot}}]|_{T_{\mathrm{i}}=T_{\mathrm{i}}^{\min}}$; otherwise, $\hat{T}_{\mathrm{i}}=T_{\mathrm{i}}^{\max}$ and $\hat{R}=E[R_{\mathrm{slot}}]|_{T_{\mathrm{i}}=T_{\mathrm{i}}^{\max}}$.
    if $\hat{R}>R^{\#}$ **then**
        Update $T_{\mathrm{i}}^{\#}=\hat{T}_{\mathrm{i}}$ and $T_{\mathrm{b}}^{\#}=T_{\mathrm{b}}$.
        Update $R^{\#}=\hat{R}$.
    end if
end for
Set $T_{\mathrm{i}}^{*}=T_{\mathrm{i}}^{\#}$ and $T_{\mathrm{b}}^{*}=T_{\mathrm{b}}^{\#}$.

Table 3.1: The parameters in the simulation

Parameters	Value
$f_{\mathrm{i}}(x)$	exponential distribution with mean 0.8
$f_{\mathrm{b}}(x)$	exponential distribution with mean 0.4
P_{d}	0.9
P_{f}	0.1
f_{s}	100 KHz
γ_{p}	8 dB
γ_{s}	10 dB
ε_{s}	0.05
ε_{c}	0.2

demonstrate the benefit of having different slot lengths for different sensing results, here we compare with a *classic scheme* that is the solution of our Problem P1 with additional constraint $T_{\mathrm{i}}=T_{\mathrm{b}}$. Fig. 3.7 shows the average throughput of secondary transmissions (called average secondary throughput) $E[R_{\mathrm{slot}}]$ of our proposed scheme and the classic scheme. We can see that our proposed scheme largely outperforms the classic scheme. For both schemes, average secondary throughput decreases with the increase of γ_{p}. The reason is that the achievable rate rate2 decreases with the growth of γ_{p}.

Then, we evaluate the effect of the collision ratio threshold ε_{c} on average secondary throughput. When ε_{c} varies from 0.05 to 0.35, Fig. 3.8 shows the average secondary throughput of our proposed scheme and the classic scheme. Our proposed scheme can achieve a larger average secondary throughput. When ε_{c} increases, average secondary throughput of our proposed scheme monotonically increases. This is because larger

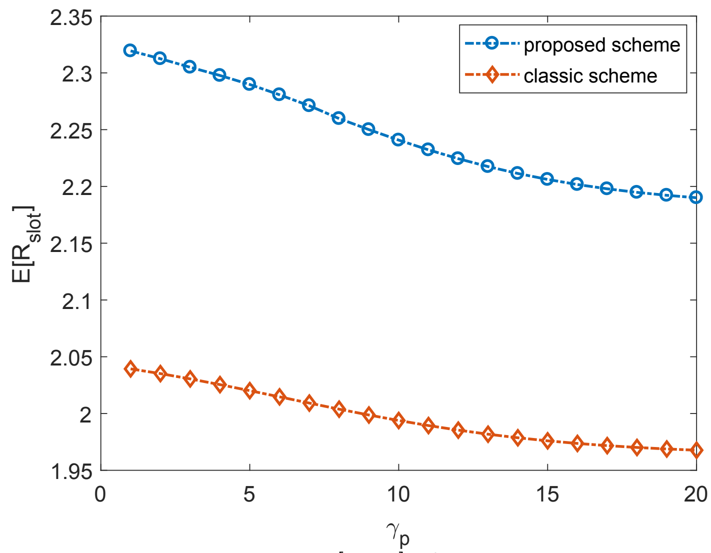

Figure 3.7: Average secondary throughput $E[R_{\text{slot}}]$ of our proposed scheme and the classic scheme versus the SNR of primary signal γ_{p}.

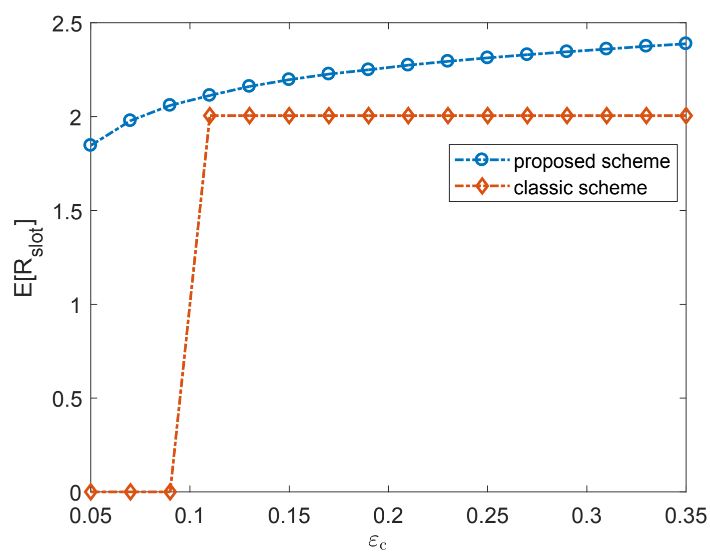

Figure 3.8: Average secondary throughput $E[R_{\text{slot}}]$ of our proposed scheme and the classic scheme versus the threshold ε_{c}.

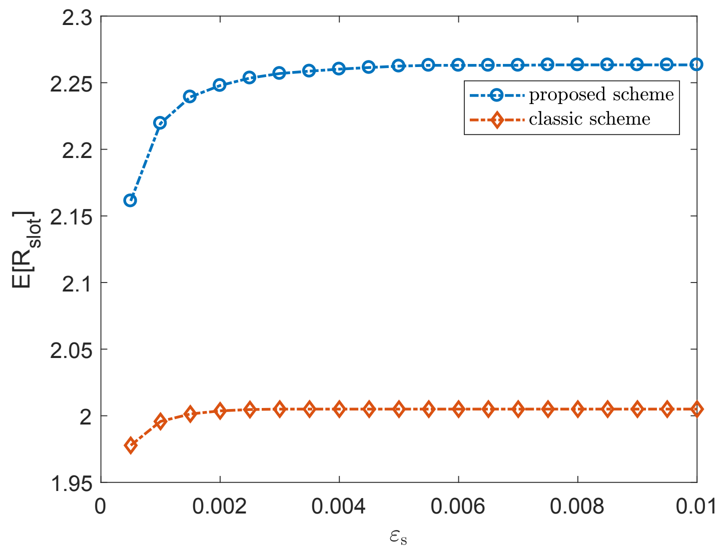

Figure 3.9: Average secondary throughput $E[R_{\mathrm{slot}}]$ of our proposed scheme and the classic scheme versus the threshold ε_{s}.

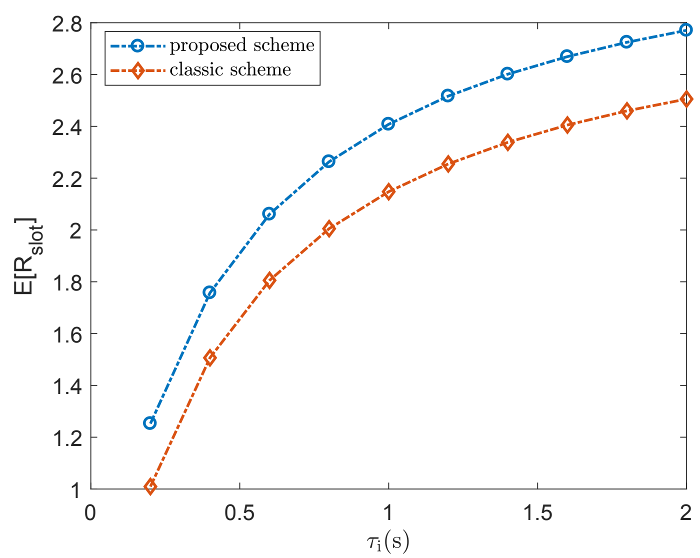

Figure 3.10: Average secondary throughput $E[R_{\mathrm{slot}}]$ of our proposed scheme and the classic scheme versus the expected sojourn time τ_{i} of an idea state.

$\varepsilon_{\rm c}$ means larger feasible set of $(T_{\rm i}, T_{\rm b})$ in our Problem P1. For the classic scheme, when $\varepsilon_{\rm c} < 0.1$, the average secondary throughput is zero. The reason is that with a stringent requirement on collision ratio (for example $\varepsilon_{\rm c} < 0.1$), no appropriate values of $T_{\rm i}$ and $T_{\rm b}$ can simultaneously satisfy constraints (3.5b), (3.5c), and the additional constraint $T_{\rm i} = T_{\rm b}$. When $\varepsilon_{\rm c} \geq 0.1$, constraints (3.5b), (3.5c), and the additional constraint $T_{\rm i} = T_{\rm b}$ can be satisfied simultaneously, and thus, the average secondary throughput is more than zero. We also notice that, for the setting with Fig. 3.8, when $\varepsilon_{\rm c} \geq 0.1$, the average secondary throughput of the classic scheme keeps stable. The reason is as follows. With $T_{\rm i} = T_{\rm b}$, the root of equation $\frac{dE[R_{\rm slot}]}{dT_{\rm i}} = 0$ is $T_{\rm i} = \varphi_7$, which is the optimal point of $T_{\rm i}$ (also $T_{\rm b}$) to maximize $E[R_{\rm slot}]$. Here $\varphi_7 = \frac{-W(\frac{V_8}{e})-1}{\lambda_{\rm i}+\lambda_{\rm b}}$ where V_8 is given as

$$V_8 = \frac{[\text{rate1} \cdot (1-P_{\rm f}) + \text{rate2} \cdot (1-P_{\rm d})]T_{\rm s}(\lambda_{\rm i}+\lambda_{\rm b})^2}{[\lambda_{\rm i}(1-P_{\rm f}) - \lambda_{\rm b}(1-P_{\rm d})](\text{rate1} - \text{rate2})} - 1.$$

In the simulation setting of Fig. 3.8, when $\varepsilon_{\rm c} \geq 0.1$, the value of φ_7 is always in the feasible region of $T_{\rm i}$. Thus, when $\varepsilon_{\rm c}$ increases, although feasible region of $T_{\rm i}$ is enlarged, the optimal point is still at $T_{\rm i} = \varphi_7$. Thus, the average secondary throughput keeps stable.

Fig. 3.9 shows the effect of the sensing ratio threshold $\varepsilon_{\rm s}$. When $\varepsilon_{\rm s}$ increases, the average throughput of both our proposed scheme and the classic scheme increase, which is intuitive. When $\varepsilon_{\rm s}$ increases beyond 0.006, the average throughput of both schemes keep stable. The reason is that with large enough $\varepsilon_{\rm s}$, the average secondary throughput is constrained by the collision ratio threshold $\varepsilon_{\rm c}$.

Fig. 3.10 shows the effect of the mean channel idle duration $\tau_{\rm i}$. It can be seen that a larger $\tau_{\rm i}$ leads to higher average secondary throughput in our proposed scheme and the classic scheme. Indeed, when the channel idle state tends to last for longer time, the secondary user has more transmission opportunities, resulting in higher throughput.

3.6 Conclusion

In this chapter, an optimal slot length configuration scheme, in which the sensing result is jointly considered to determine the slot length, is proposed. To find the optimal slot length configuration, an optimization problem is formulated, analyzed, and solved by our proposed algorithms.

Chapter 4

SDMP-based Event-Driven Centralized Opportunistic Scheduling in Wireless Networks

To cope with the spectrum scarcity problem, opportunistic scheduling is introduced to improve the spectrum efficiency. To address the limitations of existing centralized opportunistic scheduling schemes (e.g., large implementation complexity, CSI requirement, lack of QoS consideration), we formulate the opportunistic scheduling problem as a semi-Markov decision process (SMDP), in which the scheduling action is driven by events (i.e., task arrival event or task transmission completion event). Then, two scheduling methods are proposed to derive the optimal scheduling policy under fully explored networks and partially explored networks, respectively. We first propose a model-based scheduling method for the scenario with a fully explored network. In this method, to handle the challenge in deriving the transition probability and reward function, the formulated SMDP is transformed to a classic continuous time Markov decision process (CTMDP). Then, the CTMDP is uniformized, and thus, a value iteration algorithm is proposed to derive the optimal scheduling policy. For the scenario with a partially explored network (i.e., no much prior information is obtained), a model-free scheduling method is proposed. Considering the limitations of classic reinforcement learning algorithms, a revised Q-learning algorithm is proposed to derive the optimal scheduling policy in this scenario.

4.1 Introduction

By opportunistic scheduling, the spectrum efficiency can be largely improved by allocating the channel access opportunities to the users with good channel condition [77]. Some opportunistic scheduling schemes have been proposed in existing works. These schemes can be divided to static schemes and dynamic schemes. In a static scheme, a structure scheduling policy (e.g., a closed form or a threshold-based form) is generally derived. On the other hand, a scheduling policy which assigns a scheduling action for each particular state is generally derived in a dynamic scheme. Compared to static schemes, less prior information of the network is required in dynamic schemes.

In [78], a pure threshold scheduling policy is derived. In this policy, users contend for a data transmission opportunity. A user with a successful contention utilizes the channel for a transmission if the channel quality is above a certain threshold. Otherwise, the user gives up the transmission access opportunity. Thus, the optimal stopping theory is adopted to solve the access probability and the threshold. In [31], an opportunistic scheduling scheme is proposed in cooperative networks. Similar to [78], a threshold structure policy is derived. A scheduling scheme with partial channel knowledge in Device-to-Device (D2D) system is proposed in [79]. In this scheme, the rate adaption and mode selection are jointly considered, and the closed-form expressions are derived. In [80], to maximize the expected accumulated discounted reward, the channel scheduling problem is formulated as a restless multi-armed bandit (RMAB). Since the time complexity to solve the formulated problem is large, a greedy policy, which focuses on immediate reward maximization, is proposed. Then, the conditions which guarantee the optimality of the proposed greedy policy are derived. The above mentioned scheduling schemes are considered as static schemes. However, there are some limitations in these proposed schemes. For example, the channel state information (CSI) (e.g., the instantaneous CSI [31] or the statistical CSI [79]) is needed to implement these schemes. However, the communication overhead to get the CSI may be intolerable. In addition, it is not feasible to obtain the CSI in some scenarios. The implementation complexity is another challenge to take these opportunistic scheduling schemes. In these schemes, time is divided to equal length slots[1] and a scheduling action has to be taken at each time slot. In this case, the implementation complexity is pretty large in practice.

Another kind of scheduling scheme is summarized as dynamic schemes. In [81], an opportunistic scheduling scheme for multiuser wireless systems is proposed. In this scheme, a linear programming-based scheduling algorithm is derived to compute the scheduling decisions. In addition, the fairness of users is also considered in this work. A transmission scheduling scheme for the Internet of Things (IoT) is proposed in [82]. In this scheme, a reinforcement learning method, i.e., Q-learning algorithm, is introduced to obtain the optimal strategy for transmission scheduling. Then, a deep learning model is also adopted to accelerate the solution. In [83], a dynamic channel allocation problem, which focuses on maximization of the service blocking probability, is investigated in multi-beam satellite systems. To solve the formulated problem, it is modeled as a Markov decision process (MDP), and thus, a deep reinforcement learning algorithm is developed to solve the problem. In these dynamic schemes, the CSI may not be necessary information to make a scheduling decision. However, the scheduling action still needs to be taken at each time slot, which may not be very realistic. In addition, there are also some other challenges in existing scheduling schemes. In most of existing schemes, only the channel condition information is considered. The quality of service (QoS) of users or the priority of users' data is seldom considered. In addition, most of the above mentioned scheduling schemes are model-based (i.e., system models and system parameters need to be known for making a decision). In this case, some strong assumptions need to be made in these schemes (e.g., the input data rate of a user and the channel state information are assumed to follow a given distribution).

To cope with the limitations of existing works, we formulate the opportunistic scheduling problem as an SMDP, and then propose methods to derive the optimal scheduling policy under different scenarios. The major contributions of this work are summarized as follows.

1. We formulate the opportunistic scheduling problem in wireless networks as an SMDP. Different with existing opportunistic scheduling schemes, the scheduling

[1]The length of a time slot is generally needed to smaller than the channel coherence time.

action is driven by events (i.e., task arrival event or task transmission completion event), and thus, the implementation complexity is largely decreased. In addition, the priority of users (i.e., the importance of users' data), which is measured by the reward of a successful task transmission, is jointly considered in this work.

2. Considering the scenario with a fully explored network, a model-based method is proposed to derive the scheduling policy. In this method, to handle the challenge in deriving the transition probability and the reward function, the formulated SMDP is transformed to a classic continuous time Markov decision process (CTMDP). Then, the CTMDP is uniformized, and thus, a value iteration algorithm is proposed to derive the optimal scheduling policy.

3. Considering the scenario with a partially explored network, a model-free scheduling method is proposed to derive the scheduling policy. Considering the limitations of classic reinforcement learning algorithms, a revised Q-learning algorithm is proposed to derive the optimal scheduling policy.

4. Performance analysis is conducted for the proposed policies by simulation. It shows that our proposed opportunistic scheduling policies can obtain much more reward compared with other benchmark policies.

The rest of this chapter is organized as follows. Section 4.2 gives the system model and formulates the problem. We propose a model-based scheduling method in Section 4.3. A model-free scheduling method is proposed in Section 4.4. Section 4.5 shows simulation results and performance analysis. Finally, Section 4.6 concludes this chapter.

4.2 System Model and Problem Formulation

4.2.1 System Model

We consider a wireless network with N users (indexed by $1, 2, \cdots, N$) and K available channels (indexed by $1, 2, \cdots, K$). The N users contend for the K channels for data transmission. The unit of each user's data for transmission is denoted as a task. The size of a task is generally decided by the kind of application, and thus, the size of a task may be different for different users. If a user occupies a channel (e.g. channel i) to transmit a task, channel i will be occupied until the task has been transmitted to the destination. It means that the data transmission for a task cannot be interrupted. This model is reasonable for lots of scenarios. For example, in mobile edge computing systems [84], a user running a computation-intensive application is assisted by edge nodes. Computation offloading from the user to a edge node is performed over the wireless channels. The edge nodes can perform the data computing only after receiving a given amount of data (i.e., a task). Accordingly, the user should occupy a channel when it is transmitting to an edge node, and release the channel when the transmission is complete. In addition, compared to the tradition opportunistic scheduling model, the proposed model, in which the scheduling action does not need to be taken at each time slot, has a smaller implementation complexity. Note that the time needed for a task transmission relates to the transmission power, the instantaneous CSI, the size of the task, and so on. Accordingly, the transmission time of different task transmissions may be different.

A task buffer, which stores the tasks for transmission, is equipped in each user. The size of user i's task buffer is denoted as D_i. If a task buffer is full, any newly generated task has to be rejected, and thus, will be dropped. For any moment, the number of channels which are occupied by a user may be $k \in \{0, 1, \cdots, K\}$. The system model of the network is given in Fig. 4.1.

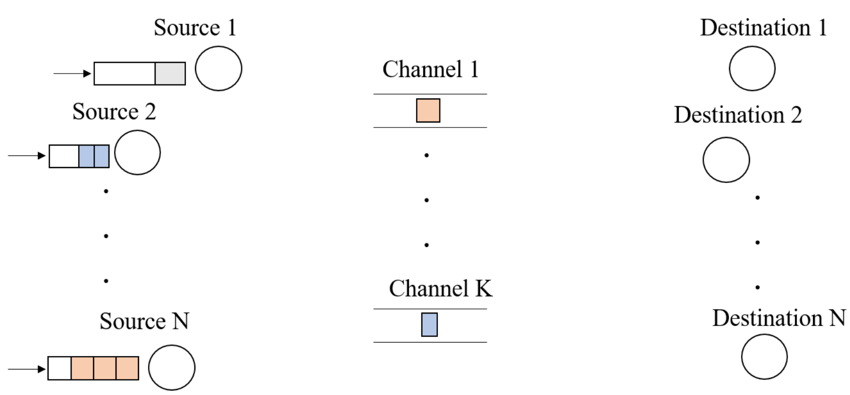

Figure 4.1: The model of the network.

In the network, the priority of different users (e.g., the importance of different users' data) may be different. In this work, it is measured by the reward of a successful task transmission. For user i, a successful task transmission can achieve a reward b_i. Similarly, a penalty related to b_i is caused if a task of user i is rejected and then dropped. The task with a large value of b_i can be regarded as a high priority [85]. Under the formulated model, we consider the opportunistic scheduling problem as follows.

4.2.2 SMDP Formulation

Once a task transmission is complete, the corresponding occupied channel will be released. Then, the next user who occupies this free channel for task transmission needs to be decided. A user i with large value of b_i can obtain large reward for a successful task transmission. On the other hand, if the task transmission of user i will be completed after a long time period, the obtained reward is reduced due to the time cost. In addition, if the task buffer of a user is full, the newly generated tasks of the user have to be dropped, and thus, a penalty is caused. Accordingly, in order to make an appropriate scheduling decision, the status of the network and the task transmission reward of different users need to be considered simultaneously.

We model the opportunistic scheduling process as an SMDP [86]. Generally, an SMDP can be represented by $\{\mathbf{S}, \mathbf{A}, t, p, r\}$, where $\mathbf{S}$ is the state space, $\mathbf{A}$ is the action space, t represents a decision epoch, p is the transition probability, and r means the reward function [87]. Then, the detailed information for each part of the formulated SMDP is stated in the following.

4.2.2.1 State Space

In our formulated SMDP, a state, denoted s, contains two components. The first component of a state is the status of each user. It is denoted as $s_1 \triangleq \langle \mathbf{d}, \mathbf{k} \rangle$ where $\mathbf{d} \triangleq \{d_1, d_2, \cdots, d_N\}$, $\mathbf{k} \triangleq \{k_1, k_2, \cdots, k_N\}$, d_i denotes the backlog of user i's task buffer, and k_i denotes the number of channels occupied by user i. Since the number of channels occupied by N users cannot be more than the maximum number of channels and the number of tasks in a task buffer cannot be more than the size of this task buffer, we have

$$\sum_{i=1}^{N} k_i \leq K, \tag{4.1}$$

$$d_i \le D_i, \quad \forall i \in \{1, 2, \cdots, N\}. \tag{4.2}$$

The second component of a state s is the event, which is expressed as $s_2 \triangleq \{e, l\}$ where $e \triangleq \{0, 1\}$ and $l \in \{1, 2, \cdots, N\}$. The element e represents the type of the event, where $e = 0$ stands for an arrival event of a task and $e = 1$ stands for a completion event of a task transmission. If $e = 0$, the element l denotes the index of the user with the arrival event. If $e = 1$, the element l denotes the index of the user with the completion event.

Thus, a state s is formulated as $s = \langle s_1, s_2 \rangle$. The state space $\mathbf{S}$ is the set of all possible states, represented as

$$\mathbf{S} \triangleq \{s | s = \langle s_1, s_2 \rangle\}. \tag{4.3}$$

4.2.2.2 Action Space

When an arrival event occurs (i.e., $e = 0$), the system needs to take one of the following actions $a_{\mathrm{r}} = \{0, 1, -1\}$. If one or more channels are free (which implies that no user has backlogged tasks), the arrival task will be allocated to a channel for transmission, and thus, $a_{\mathrm{r}} = 1$. If the corresponding user's task buffer is full, the arrival task will be dropped, and thus, $a_{\mathrm{r}} = -1$. Otherwise, the arrival task will be stored in the corresponding user's task buffer, and thus, $a_{\mathrm{r}} = 0$. Then, the action a_{r} for the arrival event is summarized as follows,

$$a_{\mathrm{r}} = \begin{cases} 1, & \sum_{i=1}^{N} k_i < K \\ -1, & d_l = D_l \\ 0, & o.w. \end{cases}. \tag{4.4}$$

When a completion event occurs (i.e., $e = 1$), the corresponding channel will be released, and thus, the system needs to take one of the following scheduling actions $a_{\mathrm{c}} = \{0, 1, 2, \cdots, N\}$, where $a_{\mathrm{c}} = 0$ means letting the released channel keep free and $a_{\mathrm{c}} = i, i \in \{1, 2, \cdots, N\}$ means that a task from user i's task buffer will be allocated to occupy the released channel for data transmission. If $a_{\mathrm{c}} = i$ where $i \neq 0$, the constraint $d_i > 0$ needs to be satisfied.

Thus, an action for a state s, denoted a, is formulated as

$$a = \begin{cases} a_{\mathrm{r}}, & e = 0 \\ a_{\mathrm{c}}, & e = 1 \end{cases}. \tag{4.5}$$

The action space $\mathbf{A}$ is the set of all available actions.

4.2.2.3 Decision Epoch

A decision is made at which point an arrival event or a completion event occurs. Thus, the decision epoch means the time point when an event occurs. Let $t_n, n \in \{0, 1, 2, \cdots\}$ denote nth decision epoch. Then, at the next decision epoch t_{n+1}, we have an arrival event or a completion event, whichever comes first. Note that t_0 means the first decision epoch, which is the start point of the process. The duration of adjacent decision epochs is decided by the probability distribution of the task arrival rate and the duration of a task transmission. A time line example of the formulated SMDP is given in Fig. 4.2.

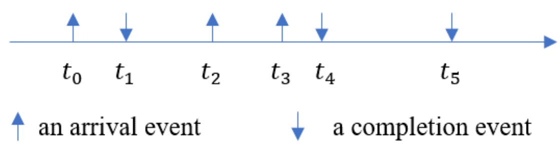

Figure 4.2: The timeline of the SMDP.

4.2.2.4 Transition Probability

The transition probability, denoted $p(s'|s,a)$, represents the probability that the system moves to state $s' \triangleq \{\mathbf{d}', \mathbf{k}', e', l'\}$ given that the previous state is s and the action a is taken at the previous decision epoch. Since the time interval of adjacent states is a random variable, the probability, that the interval from state s to the next state s' given action a is less than or equal t, is denoted as $F(t|s,a,s')$.[2] Accordingly, we have

$$p(s'|s,a) = \int_{t=0}^{\infty} dF(t|s,a,s'). \tag{4.6}$$

Given a state-action pair (s,a) and $s = \{\mathbf{d}, \mathbf{k}, e, l\}$, the element $\mathbf{d}'$ and $\mathbf{k}'$ of next state s' can be derived according to the following principles.

1. If an arrival event occurs ($e = 0$) and action $a_{\mathrm{r}} = 1$ is taken, we have

$$d_i' = d_i, \quad \forall i \in \{1, 2, \cdots, N\}, \tag{4.7}$$

[2]It means the cumulative distribution function.

$$k'_i = \begin{cases} k_i + 1, & i = l \\ k_i, & o.w. \end{cases}. \tag{4.8}$$

2. If an arrival event occurs ($e = 0$) and action $a_\mathrm{r} = 0$ is taken, we have

$$d'_i = \begin{cases} d_i + 1, & i = l \\ d_i, & o.w. \end{cases}, \tag{4.9}$$

$$k'_i = k_i, \quad \forall i \in \{1, 2, \cdots, N\}. \tag{4.10}$$

3. If an arrival event occurs ($e = 0$) and action $a_\mathrm{r} = -1$ is taken, we have

$$d'_i = d_i, \quad \forall i \in \{1, 2, \cdots, N\}, \tag{4.11}$$

$$k'_i = k_i, \quad \forall i \in \{1, 2, \cdots, N\}. \tag{4.12}$$

4. If a completion event occurs ($e = 1$) and action $a_\mathrm{c} = j, j \in \{1, 2, \cdots, N\}$ is taken, we have

$$d'_i = \begin{cases} d_i - 1, & i = j \\ d_i, & o.w. \end{cases}, \tag{4.13}$$

   If $j = l$, then, we have

$$k'_i = k_i, \quad \forall i \in \{1, 2, \cdots, N\}, \tag{4.14}$$

   If $j \neq l$, then, we have

$$k'_i = \begin{cases} k_i + 1, & i = j \\ k_i - 1, & i = l \\ k_i, & o.w. \end{cases}. \tag{4.15}$$

5. If a completion event occurs ($e = 1$) and action $a_\mathrm{c} = 0$ is taken, we have

$$d'_i = d_i, \quad \forall i \in \{1, 2, \cdots, N\}, \tag{4.16}$$

$$k'_i = \begin{cases} k_i - 1, & i = l \\ k_i, & o.w. \end{cases}. \tag{4.17}$$

If the network is fully explored (i.e., the probability distribution of task arrivals and the duration of a task transmission are known), $p(s'|s, a)$ can be derived.

4.2.2.5 Reward Function

The reward function, denoted $r(s, a)$, means the achieved reward at state s with taking action a. It is defined as follows,

$$r(s, a) \triangleq w(s, a) - uc(s, a) \tag{4.18}$$

where $w(s, a)$ denotes the profits of the task transmissions, $c(s, a)$ denotes the system cost, and u is a weight to balance the profits and the system cost.

The profits of the task transmissions are defined as,

$$w(s, a) \triangleq \begin{cases} b_l, & e = 1 \\ 0, & e = 0 \& a_{\mathrm{r}} \neq -1 \\ -b_l, & e = 0 \& a_{\mathrm{r}} = -1 \end{cases} \tag{4.19}$$

where the first case means a task of user l is successfully transmitted, and thus, a reward b_l can be obtained, the second case means a new arrival task is stored to the corresponding task buffer, the third case means the new arrival task of user l is discarded, and thus, a penalty b_l is caused.

The system cost $c(s, a)$ is defined as,

$$c(s, a) \triangleq o(s, a)\tau(s, a) \tag{4.20}$$

where $o(s, a)$ represents the cost rate between two consecutive decision epochs and $\tau(s, a)$ means the expected time interval to the next state s'. When a discounted model is considered, the discount factor should be added to $\tau(s, a)$ since $c(s, a)$ applies throughout a period (i.e., from state s to s'). For $o(s, a)$, it is defined as the number of occupied channels, and thus, we have

$$o(s, a) \triangleq \sum_{i=1}^{N} k_i'. \tag{4.21}$$

4.2.3 Problem Formulation

In this work, we try to find an optimal opportunistic scheduling policy to obtain the maximum reward over a long-term. Under the formulated SMDP, we formulate the scheduling problem as an infinite-horizon discounted reward semi-Markov decision problem. Let π denote the scheduling policy, which is defined as a mapping from the state space $\mathbf{S}$ to the action space $\mathbf{A}$, $\pi : \mathbf{S} \rightarrow \mathbf{A}$.

For a policy π, let $v^{\pi}(s_0)$ denote the expected infinite-horizon discounted reward given initial state s_0 at the first decision epoch. Then, $v^{\pi}(s_0)$ is given as [50]

$$v^{\pi}(s_0) = E_{s_0}^{\pi}[\sum_{n=0}^{\infty} e^{-\alpha t_n} r(s_n, a_n|s_0)] \tag{4.22}$$

where $t_0, t_1, \cdots$ denotes the successive decision epochs, α is the discount factor, s_n and a_n are the state and the corresponding applied action at nth decision epoch t_n, respectively. Then, according to the definition of transition probabilities, (4.22) can be converted to

$$\begin{aligned} v^{\pi}(s_0) &= r(s_0, \pi(s_0)) + E_{s_0}^{\pi}[e^{-\alpha t_1} v^{\pi}(s_1)] \\ &= r(s_0, \pi(s_0)) + \sum_{s_1 \in \mathbf{S}} \int_0^{\infty} e^{-\alpha t} p(s_1|s_0, \pi(s)) dF(t|s_0, a, s_1) v^{\pi}(s_1). \end{aligned} \tag{4.23}$$

The objective of this work is to find an optimal policy which can obtain the maximum expected long-term discounted system reward $v^{\pi}(s_0)$. Let π^* denote the optimal policy. Accordingly, we have

$$\pi^* = \arg\max_{\pi} v^{\pi}(s), \quad \forall s \in \mathbf{S}. \tag{4.24}$$

In our formulated problem, the action space $\mathbf{A}$ is finite for all states $s \in \mathbf{S}$, and thus, there is an optimal stationary deterministic policy π^{∞} (i.e., π^*) for our formulated problem [50]. In the following two sections, we will propose two methods, model-based and model-free, to derive the optimal stationary policy under different scenarios.

4.3 Model-based Scheduling Method

In reality, some information of the network, such as, the probability distribution of users' task arrival rate and the expected duration of a task transmission for each user, may be known to the network controller (e.g., by training the historic data). In this scenario, the model of the network is fully explored, and thus, the transition probabilities and the reward function can be derived. Accordingly, in this section, we present a model-based scheduling method to solve the opportunistic scheduling problem under the scenario with a fully explored network. The task arrival event of user i is assumed to follow Poisson distribution with parameter λ_i and the task transmission time of user i is assumed to follow exponential distribution with parameter μ_i.[3] These are reasonable assumptions according to practical measurements [88]. Then, the time interval between adjacent decision epochs also follows an exponential distribution. Consequently, to handle the challenge in deriving the transition probability and the reward function, the formulated SMDP is transformed to a classic continuous-time Markov decision process (CTMDP).

4.3.1 Model of CTMDP

The state space and the action space of the formulated CTMDP are same as those of the formulated SMDP in Section 4.2. Thus, to model the CTMDP, we need to derive the state transition probability and the reward function.

Firstly, the state transition probability is considered. Let $s = \{\mathbf{d}, \mathbf{k}, e, l\}$ and $s' = \{\mathbf{d}', \mathbf{k}', e', l'\}$ denote the current state and the next state, respectively. If an arrival event or a completion event occurs, an action needs to be taken. To derive the state transition probability, we introduce the following lemma about the exponential distribution.

Lemma 4. *Let $\mathbf{X} = \{X_1, X_2, \cdots, X_j\}$ denote a group of independent variables. For any element $X_i, i \in \{1, 2, \cdots, j\}$, it follows exponential distribution with parameter θ_i. Then, the minimum of $\mathbf{X}$ is also an exponentially distributed random variable with parameter $\sum\limits_{i=1}^{j} \theta_i$.*

It is easily to prove Lemma 4 according to the properties of the exponential distribution. The similar proof can also be found in [87], and thus, is omitted here. According to Lemma 4, for a state-action pair (s, a), the time interval to the first event occurrence (i.e., an arrival event or a completion event) follows an exponential distribution with a parameter $\gamma(s, a)$. In addition, $\gamma(s, a)$ can be derived as follows,

$$\gamma(s, a) = \sum_{i=1}^{N} \lambda_i + \sum_{i=1}^{N} k_i' \mu_i \tag{4.25}$$

where k_i' is derived by (4.7)-(4.17). Therefore, $F(t|s, a, s')$ is given as

$$F(t|s, a, s') = 1 - e^{-\gamma(s,a)t}, t > 0. \tag{4.26}$$

[3] For simplicity, for a user, the task transmission time under different channels is assumed to follow the same distribution. By adding some elements to a state s, our work can be easily extended to the case with different distributions.

Then, we introduce another lemma about the exponential distribution.

Lemma 5. *For two independent exponentially distributed variables X_1 and X_2 with parameters θ_1 and θ_2, respectively, the probability of the event $X_1 < X_2$ is $P(X_1 < X_2) = \frac{\theta_1}{\theta_1+\theta_2}$.*

For a given state-action pair (s, a), the next state s' should be one of the following states,

$$s' = \begin{cases} \{\mathbf{d}', \mathbf{k}', 0, l'\} \\ \{\mathbf{d}', \mathbf{k}', 1, l'\} \end{cases} \tag{4.27}$$

where $\mathbf{d}'$ and $\mathbf{k}'$ can be derived by (4.7)-(4.17). For the first case in (4.27), it means that the first event occurrence after current decision epoch is an arrival event of user l'. Let X_1 denote the event that the first event occurrence after current decision epoch is an arrival event of user l' and let X_2 denote the event that the first event occurrence after current decision epoch is any possible event other than an arrival event of user l'. Thus, X_1 and X_2 should follow exponential distribution with parameters $\lambda_{l'}$ and $\gamma(s, a) - \lambda_{l'}$, respectively. Then, according to Lemma 5, the transition probability of the case, which is $s' = \{\mathbf{d}', \mathbf{k}', 0, l'\}$, should be

$$p(s'|s, a) = \frac{\lambda_{l'}}{\gamma(s, a)}. \tag{4.28}$$

Similarly, for the second case in (4.27), it means that the first event occurrence after current decision epoch is a completion event of user l'. Let X_3 denote the event that the first event occurrence after current decision epoch is a completion event of user l' and let X_4 denote the event that the first event occurrence after current decision epoch is any possible event other than a completion event of user l'. Thus, X_3 and X_4 should follow exponential distribution with parameters $k'_{l'}\mu_{l'}$ and $\gamma(s, a) - k'_{l'}\mu_{l'}$, respectively. Then, according to Lemma 5, the transition probability of the case, which is $s' = \{\mathbf{d}', \mathbf{k}', 1, l'\}$, should be

$$p(s'|s, a) = \frac{k'_{l'}\mu_{l'}}{\gamma(s, a)}. \tag{4.29}$$

Then, we consider the reward function $r(s, a)$. For a state-action pair (s, a), the expected time interval to the next state s' should be $\tau(s, a) = \frac{1}{\gamma(s,a)}$. Since the discounted reward is considered, the discount factor α for continuous time should be added to $\tau(s,a)$ to meet the practical condition. Accordingly, the expected discounted reward function is given as

$$\begin{array}{rl} r(s,a) & = w(s,a) - uo(s,a)E[\int_0^{\tau(s,a)} e^{-\alpha t}dt] \\ & = w(s,a) - uo(s,a)E[\frac{1-e^{-\alpha\tau(s,a)}}{\alpha}] \\ & = w(s,a) - \frac{uo(s,a)}{\alpha+\gamma(s,a)} \end{array} \tag{4.30}$$

where $w(s,a)$ and $o(s,a)$ are given in (4.19) and (4.21), respectively.

4.3.2 Proposed Value Iteration Algorithm

In this section, the optimal policy of the formulated CTMDP will be derived. First, we need to derive the Bellman optimality equation, and thus, a lemma is introduced as follows.

Lemma 6. *The Bellman optimality equation of the formulated CTMDP is given as follows*

$$v(s) = \max_{a\in\mathbf{A}}\{r(s,a) + \varsigma \sum_{s'\in\mathbf{S}} p(s'|s,a)v(s')\} \tag{4.31}$$

where $\varsigma = \frac{\gamma(s,a)}{\gamma(s,a)+\alpha}$.

Proof. According to (4.23) and (4.26), the infinite-horizon discounted reward $v^\pi(s)$ is given as

$$\begin{array}{l} v^\pi(s) \\ = r(s,\pi(s)) + \sum\limits_{s'\in\mathbf{S}} \int_0^\infty e^{-\alpha t} p(s'|s,\pi(s))dF(t|s,a,s')v^\pi(s') \\ = r(s,\pi(s)) + \sum\limits_{s'\in\mathbf{S}} p(s'|s,\pi(s))v^\pi(s')\int_0^\infty e^{-\alpha t}\gamma(s,a)e^{-\gamma(s,a)t}dt \\ = r(s,\pi(s)) + \frac{\gamma(s,a)}{\gamma(s,a)+\alpha} \sum\limits_{s'\in\mathbf{S}} p(s'|s,\pi(s))v^\pi(s'). \end{array} \tag{4.32}$$

Therefore, the Bellman optimality equation of the formulated CTMDP is given as (4.31). □

Different from DTMDP, there are no standard methods to derive the optimal policy for a CTMDP. Thus, we try to convent the formulated CTMDP to a discrete-time Markov chain by uniformization, so that the process can be analyzed and the optimal policy can be derived. In order to guarantee the effectiveness of the uniformization, we first introduce a parameter $\varphi \triangleq \sum\limits_{i=1}^{N} \lambda_i + \max\limits_{i\in\{1,2,\cdots,N\}}\{Ku_i\}$, and thus, the following condition is satisfied [50].

$$[1-p(s'|s,a)]\gamma(s,a) \le \varphi, \forall s \in \mathbf{S} \,\&\, \forall a \in \mathbf{A}. \tag{4.33}$$

Then, the steps of uniformization are given as follows. The components of the formulated CTMDP after uniformization are denoted by $\sim$.

1. State space and action space: we have $\tilde{\mathbf{S}} = \mathbf{S}$ and $\tilde{\mathbf{A}} = \mathbf{A}$.

2. Transition probability: The formulated CTMDP is transformed to a new process, which is observed after random time intervals with exponential distribution with parameter φ. Thus, the uniformized transition probabilities are formulated as

$$\tilde{p}(s'|s,a) = \begin{cases} 1 - \frac{\gamma(s,a)[1-p(s'|s,a)]}{\varphi}, & s' = s \\ \frac{\gamma(s,a)p(s'|s,a)}{\varphi}, & s' \neq s \end{cases} \tag{4.34}$$

3. Reward function: The uniformized reward function is formulated as

$$\tilde{r}(s,a) = r(s,a)\frac{\gamma(s,a)+\alpha}{\varphi+\alpha} \tag{4.35}$$

Under the aforementioned uniformization, the Bellman optimality equation (4.31) is transformed to the following equation.

$$\tilde{v}(s) = \max_{a\in\mathbf{A}}\{\tilde{r}(s,a) + \tilde{\varsigma}\sum_{s'\in\mathbf{S}}\tilde{p}(s'|s,a)\tilde{v}(s')\} \tag{4.36}$$

where $\tilde{\varsigma} = \frac{\varphi}{\varphi+\alpha}$. About the Bellman optimality equality (4.31) and (4.36), we have the following lemma.

Lemma 7. *For the Bellman optimality equality (4.31) and (4.36), we have*

$$v^{\pi^\infty}(s) = \tilde{v}^{\pi^\infty}(s) \tag{4.37}$$

where π^∞ denotes the optimal stationary policy.

Proof. Given a policy π and a state s, the number of transitions from state s to itself for the uniformized Markov process is denoted as $\tilde{z}_{\mathrm{s}}$. Then, the probability of the event $\tilde{z}_{\mathrm{s}} = j, j \in \{0,1,2,\cdots\}$ is

$$P\{\tilde{z}_{\mathrm{s}} = j\} = [\tilde{p}(s|s,a)]^j[1-\tilde{p}(s|s,a)] \tag{4.38}$$

where $a = \pi(s)$ is the applied action for state s under the policy π.

Thus, the expected total discounted reward during sojourns in state s is formulated as $E_s^\pi[\sum_{n=0}^{\tilde{z}_{\mathrm{s}}} e^{-\alpha\tilde{t}_n}\tilde{r}(s,a)]$, where $\tilde{t}_n$ is the time interval of adjacent states in the uniformized Markov process. Accordingly, $\tilde{t}_n$ follows exponential distribution with parameter φ. Then, $E_s^\pi[\sum\limits_{n=0}^{\tilde{z}_s} e^{-\alpha\tilde{t}_n}\tilde{r}(s,a)]$ can be calculated by

$$E_s^\pi[\sum_{n=0}^{\tilde{z}_s} e^{-\alpha\tilde{t}_n}\tilde{r}(s,a)] = \tilde{r}(s,a)E_s^\pi[\sum_{n=0}^{\tilde{z}_s}\tilde{\varsigma}^n] \tag{4.39}$$

where $\tilde{\varsigma}$ is defined in (4.36). According to (4.38), we have

$$\begin{aligned} E_s^\pi[\sum_{n=0}^{\tilde{z}_s}\tilde{\varsigma}^n] &= \sum_{\tilde{z}_s=0}^{\infty}[\tilde{p}(s|s,a)^{\tilde{z}_s}[1-\tilde{p}(s|s,a)]\sum_{n=0}^{\tilde{z}_s}\tilde{\varsigma}^n] \\ &= \tfrac{1}{1-\tilde{\varsigma}\tilde{p}(s|s,a)}. \end{aligned} \tag{4.40}$$

Therefore, $E_s^\pi[\sum\limits_{n=0}^{\tilde{z}_s} e^{-\alpha\tilde{t}_n}\tilde{r}(s,a)]$ is given as

$$E_s^\pi[\sum_{n=0}^{\tilde{z}_s} e^{-\alpha\tilde{t}_n}\tilde{r}(s,a)] = \frac{\tilde{r}(s,a)}{1-\tilde{\varsigma}\tilde{p}(s|s,a)}. \tag{4.41}$$

Similarly, for the original formulated CTMDP, the expected total discounted reward during sojourns in state s, denoted $E_s^\pi[\sum\limits_{n=0}^{z_s} e^{-\alpha t_n}r(s,a)]$, should be

$$E_s^\pi[\sum_{n=0}^{z_s} e^{-\alpha t_n}r(s,a)] = \frac{r(s,a)}{1-\varsigma p(s|s,a)} \tag{4.42}$$

where z_s is the number of transitions from state s to itself for the original formulated CTMDP.

According to (4.34) and (4.35), we have

$$\begin{aligned} \tfrac{\tilde{r}(s,a)}{1-\tilde{\varsigma}\tilde{p}(s|s,a)} &= \tfrac{r(s,a)(\gamma(s,a)+\alpha)}{\alpha+\gamma(s,a)(1-p(s|s,a))} \\ &= \tfrac{r(s,a)}{1-\frac{\gamma(s,a)}{\gamma(s,a)+\alpha}p(s|s,a)} \\ &= \tfrac{r(s,a)}{1-\varsigma p(s|s,a)}. \end{aligned} \tag{4.43}$$

Accordingly, $E_s^\pi[\sum\limits_{n=0}^{\tilde{z}_s} e^{-\alpha\tilde{t}_n}\tilde{r}(s,a)] = E_s^\pi[\sum\limits_{n=0}^{z_s} e^{-\alpha t_n}r(s,a)]$ is obtained. In addition, the original formulated CTMDP and the uniformized Markov process have the same state occupancy distributions [50], and thus, the expected total discounted reward during sojourns in any state $s \in \mathbf{S}$ is the same. Therefore, (4.37) is obtained. This completes the proof. □

According to Lemma 7, we can obtain the optimal stationary π^∞ by considering the Bellman optimality equation (4.36). Accordingly, we propose a value iteration algorithm, denoted Algorithm 4.1. If $\xi \to 0$, the policy obtained by Algorithm 4.1 converges to the optimal stationary policy for the formulated opportunistic scheduling problem [54]. For each iteration of Algorithm 4.1, the computation complexity is $\mathbf{O}(|\mathbf{A}||\mathbf{S}|^{\mathbf{2}})$, where $|\mathbf{A}|$ and $|\mathbf{S}|$ are the size of the action space and state space, respectively. The number of iterations to reach the convergence relates to the discount factor α. The space complexity of Algorithm 4.1 is $\mathbf{O}(|\mathbf{S}|)$. Note that the optimal policy is obtained offline by Algorithm 4.1.

Algorithm 4.1 Value Iteration Algorithm

Set the reward value $\tilde{v}^0(s) = 0$ for all states,
Set a small constant $\xi > 0$ and set $j = 0$,
loop:
For each state $s \in \mathbf{S}$, calculate $\tilde{v}^{j+1}(s)$ according to:

$$\tilde{v}^{j+1}(s) = \max_{a \in \mathbf{A}} \{\tilde{r}(s,a) + \tilde{\varsigma} \sum_{s' \in \mathbf{S}} \tilde{p}(s'|s,a)\tilde{v}^j(s')\} \tag{4.44}$$

if $||\tilde{\mathbf{v}}^{j+1} - \tilde{\mathbf{v}}^j|| > \xi$ where $||\cdot||$ is the norm function, which is defined as $||\tilde{\mathbf{v}}^{j+1} - \tilde{\mathbf{v}}^j|| = \max_{s \in \mathbf{S}} |\tilde{v}^{j+1}(s) - \tilde{v}^{j+1}(s)|$ **then**
    $j = j + 1$
    goto *loop*
else
    The optimal stationary policy for any state $s \in \mathbf{S}$ is

$$\pi^*(s) = \arg\max_{a \in \mathbf{A}} \{\tilde{r}(s,a) + \tilde{\varsigma} \sum_{s' \in \mathbf{S}} \tilde{p}(s'|s,a)\tilde{v}^j(s')\} \tag{4.45}$$

end if

4.4 Model-free Scheduling Method

In the model-based scheduling method, the network model is assumed to be fully explored so that the transition probability and the reward function can be derived directly. However, this assumption may not be always valid [89]. In many scenarios, the prior information of the network, e.g., the probability distribution of users' task arrival rate and the expected duration of a task transmission, is hard to obtain, and thus, the network is partially explored. In these scenarios, the model-based scheduling policy may not be optimal. To address this challenge, a model-free reinforcement learning method is proposed to obtain the scheduling policy in partially explored networks.

In model-free reinforcement learning, it solves the Bellman optimality equation to obtain the optimal policy by asynchronous iteration. Q-learning, which is a typical reinforcement learning method, is introduced. Let $Q^{\pi}(s,a)$ (named Q-value [54]) denote the expected long-term discounted reward of state-action pair (s,a), where $a=\pi(s)$ is the applied action at state s under the policy π. Therefore, the expected infinite-horizon discounted reward $v^{\pi}(s)$, which is defined in (4.22) can be obtained by

$$v^{\pi}(s) = \max_{a\in\mathbf{A}} Q^{\pi}(s,a). \tag{4.46}$$

For an optimal policy π^*, its optimal Q-value of the state-action pair (s,a) is denoted as $Q^*(s,a)$. Thus, if the optimal Q-value of each state-action pair can be obtained, the optimal scheduling policy π^* is given as,

$$\pi^*(s) = \max_{a\in\mathbf{A}} Q^*(s,a), \forall s \in \mathbf{S}. \tag{4.47}$$

Accordingly, given a state s, the optimal action a for this state is decided by the optimal Q-value $Q^*(s,a), \forall a \in \mathbf{A}$. For a classic discrete MDP problem, $Q^*(s,a)$ can be obtained by the following iterative equation (i.e., Q-learning method) [54]

$$Q(s,a) = Q(s,a) + \kappa(r(s,a) + \rho \max_{a'\in\mathbf{A}} Q(s',a') - Q(s,a)) \tag{4.48}$$

where $\kappa \in (0,1]$ is the learning rate, $\rho \in (0,1)$ means a constant discount factor, s' and a' denotes the next state of state s and the applied action of state s', respectively. However, for our formulated SMDP, (4.48) is not valid to calculate the Q-value due to the random decision epochs. Accordingly, a revised Q-learning method is proposed to derive the optimal policy.

In this work, we first derive the equation (refer to (4.50)) to derive the optimal value of $Q(s,a)$ in our formulated SMDP. Then, the revised iterative equation (refer to (4.51)) for our formulated SMDP is proposed. Note that the revised iterative equation is different to the equation in classic Q-learning method (i.e., (4.48)). At last, the proposed Q-learning algorithm is given in Algorithm 4.2. The detailed procedures are given as follows.

According to (4.46), (4.23) can be rewritten as

$$Q^{\pi}(s,a) = r(s,a) + \sum_{s'\in\mathbf{S}} \int_0^{\infty} e^{-\alpha t} p(s'|s,a) dF(t|s,a,s') \max_{a'\in\mathbf{A}} Q^{\pi}(s',a') \tag{4.49}$$

where s' and a' are the next state of current state s and the applied action of state s', respectively. Therefore, $Q^*(s,a)$ satisfies the equation

$$\begin{aligned} Q^*(s,a) = (w(s,a) - uo(s,a) \sum_{s' \in \mathbf{S}} \int_0^\infty \int_0^t e^{-\alpha h} p(s'|s,a) dh dF(t|s,a,s')) \\ + \sum_{s' \in \mathbf{S}} \int_0^\infty e^{-\alpha t} p(s'|s,a) dF(t|s,a,s') \max_{a' \in \mathbf{A}} Q^*(s',a'). \end{aligned} \tag{4.50}$$

According to the Q-learning method and (4.50), the following Q-learning rule for our formulated SMDP is introduced to obtain the optimal Q-value for each state-action pair.

$$\begin{aligned} & Q_{n+1}(s,a) \\ & = Q_n(s,a) + \kappa_n[w(s,a) - \tfrac{1-e^{-\alpha t_n}}{\alpha} uo(s,a) + e^{-\alpha t_n} \max_{a' \in \mathbf{A}} Q_n(s',a') - Q_n(s,a)] \end{aligned} \tag{4.51}$$

where κ_n is the learning rate of the nth decision epoch, and t_n is the actual time interval between the nth decision epoch and the $(n+1)$th decision epoch. If every state-action pair is iterated infinitely by (4.51), $Q_{n+1}(s,a)$ would converge to the optimal Q-value $Q^*(s,a)$ [90], and then, the optimal scheduling policy can be obtained by (4.47). In practice, the Q-value of each state-action pair is considered to converge to the optimal Q-value if the Q-value of each state-action pair keeps stable.

Algorithm 4.2 Proposed Q-learning algorithm

Set the Q-value $Q_0(s,a) = 0$ for all possible state-action pairs,
Set the number of state-action visits $\psi(s,a) = 0$ for all possible state-action pairs,
Set the parameters κ_0 and α,
Set the index of the decision epoch $n = 0$,
loop:
Observe current state s,
Take a random action $\hat{a}$ with probability $\varepsilon(s)$, Otherwise take the action $\hat{a} = \arg\max_{a \in \mathbf{A}} Q_n(s,a)$,
Update κ_n,
Update $\psi(s,\hat{a}) = \psi(s,\hat{a}) + 1$,
Update $\varepsilon(s)$ according to (4.52),
Monitor the first next event occurrence and observe the next state s',
Update $Q_{n+1}(s,\hat{a})$ according to (4.51),
Set $Q_{n+1}(\bar{s},\bar{a}) = Q_n(\bar{s},\bar{a})$ where $(\bar{s},\bar{a})$ means the other state-action pairs except $(s,\hat{a})$,
Set $s = s'$ and $n = n+1$,
goto *loop*

To obtain the optimal Q-value $Q^*(s,a)$ for every state-action pair (s,a), we need to run a long-term to update the value of $Q(s,a)$ by (4.51). Accordingly, at nth decision epoch, an action a for current state s needs to be chosen. To receive a large reward, the action $\hat{a} = \arg\max_{a \in \mathbf{A}} Q_{n-1}(s,a)$ may be preferred to be taken. This is named as the exploration operation. However, to discover the possible more effective actions, an action other than that suggested by the exploration operation should also be chosen. This is named as the exploration operation. By the exploration operation, it can guarantee that all states of the Markov chain are visited and all possible actions for each state are tried out. Accordingly, the exploration operation and exploitation operation should be applied simultaneously in the Q-learning algorithm. In our work, we adopt the ε-greedy exploration-exploitation policy [54], which is the most common exploration method. In this policy, the system in state s takes the action by the exploration operation with probability $1-\varepsilon(s)$ and takes a random action with probability $\varepsilon(s)$. To guarantee the convergence of the Q-learning algorithm, the exploitation operation should not be stopped, but the probability $\varepsilon(s)$ needs to be reduced over time. Therefore, we introduce an variable, denoted $\psi(s,a)$, to record the number of visits of state-action pair (s,a). Then, the value of $\varepsilon(s)$ is defined as

$$\varepsilon(s) = \frac{1}{ln(\max_{a \in \mathbf{A}} \psi(s,a) + 2)}. \tag{4.52}$$

Then, we propose a Q-learning algorithm, denoted Algorithm 4.2, to derive the optimal Q-value for all possible state-action pairs.

In Algorithm 4.2, the learning procedure is repeated until the end of the learning period. Note that, if the learning period reaches to a pre-defined value [87] or the Q-value of all possible state-action pairs reaches a convergence [54], the learning period can be regarded as an end. Then, the opportunistic scheduling policy is obtained by (4.47). The number of iterations to reach the convergence relates to the size of space state and action state. However, it is still an open question to derive the exact number of iterations. In our formulated SMDP, the optimal action of a state s is already given in (4.4) if the element e satisfies the constraint $e = 0$. Accordingly, the number of iterations to reach a convergence for Algorithm 4.2 can be largely decreased.

4.5 Numerical Results

In this section, simulation results are provided to evaluate the performance of our proposed policies. The parameters adopted in the simulation are given in Table 4.1.

Table 4.1: The parameters in the simulation

Parameters	Value
N	5
K	4
$D_i, i = \{1, 2, \cdots, N\}$	5
$\{b_1, b_2, b_3, b_4, b_5\}$	$3, 2.5, 2, 1.5, 1$
$\lambda_i, i = \{1, 2, \cdots, N\}$	1
$\{\frac{1}{\mu_1}, \frac{1}{\mu_2}, \frac{1}{\mu_3}, \frac{1}{\mu_4}, \frac{1}{\mu_5}\}$	$1, 0.5, 0.8, 0.7, 0.9$
ξ	10^{-10}
α	0.1
κ_0	0.1
u	1.5

Firstly, we compare the proposed model-based scheduling policy and model-free scheduling policy[4], and investigate the convergence of the proposed Q-learning algorithm. For the model-based scheduling policy, the network is assumed to be fully explored, and thus, the transition probability and reward function can be derived. For the model-free scheduling policy, the network has no knowledge of these information, and thus, it costs a long time for the learning process and gets the optimal Q-value for each state-action pair. The average reward, which is defined as $\frac{R}{T}$ where R is the accumulated reward and T is the running time, is obtained by the model-based scheduling policy and the model-free scheduling policy, given in Fig. 4.3. It can be seen that the reward obtained by the model-free scheduling policy is almost the same with that of the model-based scheduling policy.

In order to evaluate the performance of our proposed policy, we compare with the following benchmark policies.

1. Large backlog go first policy (BF policy): In this policy, if a completion event occurs (i.e., a channel is free), the user, which has the largest backlog, is assigned to utilize the free channel for a task transmission. If the backlog of two users' task buffers are the same, user i with a larger value of b_i has a higher priority to utilize the free channel for task transmission.

2. High priority go first policy (PF policy): In this policy, if a completion event occurs, user i, which has the highest priority (i.e., the largest value of b_i), is

[4]Model-based scheduling policy means that the policy is obtained by our proposed model-based method, and model-free scheduling policy means that the the policy is obtained by our proposed model-free method.

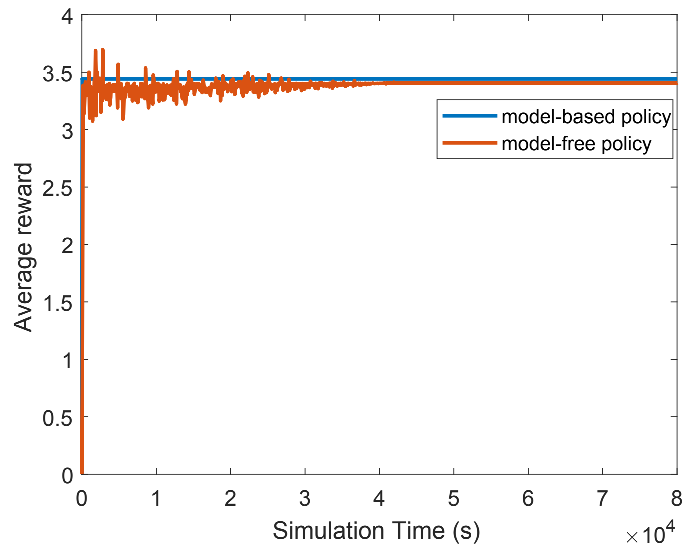

Figure 4.3: The average reward with the model-based policy and model-free policy.

assigned to utilize the free channel for a task transmission. Thus, in our simulation, user 1 has the highest priority to utilize the channel. In contrast, user 5 has the lowest priority to utilize the channel.

3. Greedy policy: In this policy, a parameter for user i, denoted $g_i = b_i\mu_i$, is defined as the expected obtained reward per unit of time when a task of user i is transmitted. Thus, if a completion event occurs, a task of user i with the largest value of g_i is assigned to be transmitted by the free channel. Thus, in our simulation, user 2 has the highest priority to utilize the channel. In contrast, user 5 has the lowest priority to utilize the channel.

Then, we compare the performance between the benchmark policies and our proposed policy. Fig. 4.4-Fig. 4.6 show the simulation results under different size of task buffers. The size of different users' task buffers are set as the same and vary from 3 to 10. The simulation time is set as 10^5 s. The simulation results of the accumulated reward with different policies are shown in Fig. 4.4. It can be seen that our proposed policy can receive the largest reward, which means our proposed policy has a better performance compared to these benchmark policies. In contrast, the PF policy receives the least reward compared to other policies. The obtained reward by the PF policy is close to that of the greedy policy. This is because the number of tasks which is rejected and dropped is almost the same by these two policies, which can be validated by Fig. 4.5 and Fig. 4.6. In addition, we observe that the obtained reward grows with the increase of D. The reason is that the number of rejected tasks is reduced with the increase of D. The reward keeps stable when the value of D keeps increasing beyond a large value. The reason is that the number of tasks, which can be transmitted, is limited due to the limited number of channels in the network.

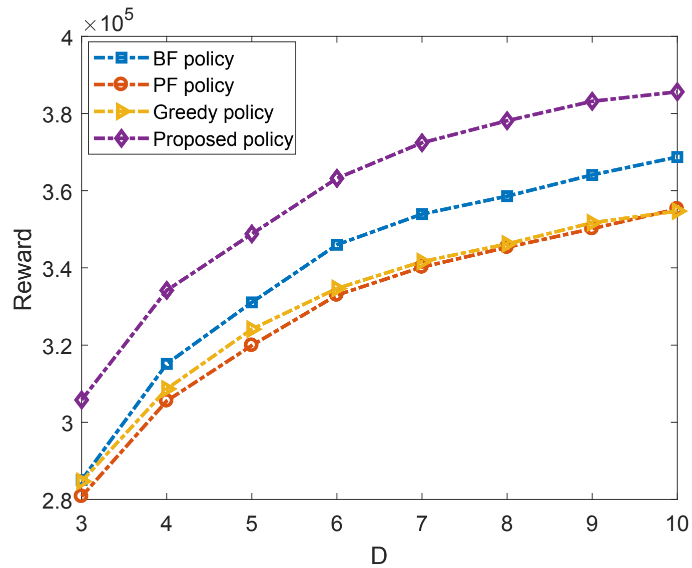

Figure 4.4: The reward versus the size (i.e., D) of the task buffer.

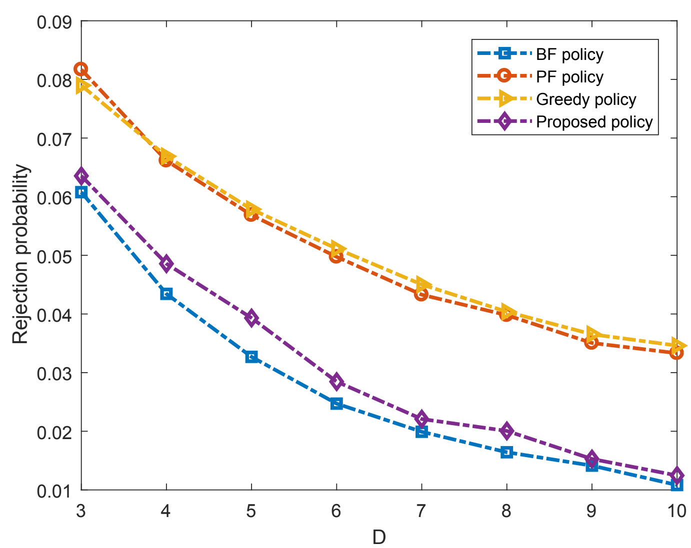

Figure 4.5: The overall rejection probability of task buffers versus the size (i.e., D) of the task buffer.

The simulation results of the overall rejection probability with different policies are given in Fig. 4.5. The overall rejection probability, denoted P_{r}, is defined as

$$P_r = \frac{num_{\mathrm{r}}}{num_{\mathrm{all}}} \tag{4.53}$$

where num_{r} is the number of tasks which are rejected and dropped in the network, and num_{all} means the number of total generated tasks. If a task buffer is full, the newly generated tasks will be rejected and dropped. In Fig. 4.5, we observe that the BF policy has the smallest rejection probability compared to other policies. The reason is that the BF policy make a scheduling decision according to the backlog of different task buffers. Our proposed policy also has a small rejection probability. When the value of D takes a large value, the rejection probability of the BF policy and our proposed policy are very small, and thus, the BF policy and our proposed policy can obtain a much larger reward compared to the PF policy and the greedy policy, which can be observed in Fig. 4.4.

Fig. 4.6 gives the rejection probability of different user's task buffer. The rejection probability of user i, denoted $P_{\mathrm{r},i}$, is defined as

$$P_{\mathrm{r},i} = \frac{num_{\mathrm{r},i}}{num_i} \tag{4.54}$$

where $num_{\mathrm{r},i}$ is the number of user i's tasks which are rejected and dropped in the network, and num_i means the number of user i's total generated tasks. For the PF policy, the rejection probability of user 1 is the smallest, on the other hand, the rejection probability of user 5 is the largest. It confirms our intuitive understanding that user 1 and user 5 have the highest priority and lowest priority, respectively, to utilize the free channel in the PF policy. Since the user's order to utilize the free channel in the greedy policy is $\{2, 1, 3, 4, 5\}$, the rejection probability of the greedy policy should have the following feature $P_{\mathrm{r},2} > P_{\mathrm{r},1} > P_{\mathrm{r},3} > P_{\mathrm{r},4} > P_{\mathrm{r},5}$, which is validated by the simulation results of Fig. 4.6. In Fig. 4.6, the rejection probability of our proposed scheme has the same feature with the greedy policy, which is $P_{\mathrm{r},2} > P_{\mathrm{r},1} > P_{\mathrm{r},3} > P_{\mathrm{r},4} > P_{\mathrm{r},5}$. However, our proposed scheme has a much smaller rejection probability compared to the greedy policy, and thus, our proposed scheme can receive a larger reward.

At last, we consider the accumulated reward of different policies under the cases with different number of available channels (K). The simulation results are shown

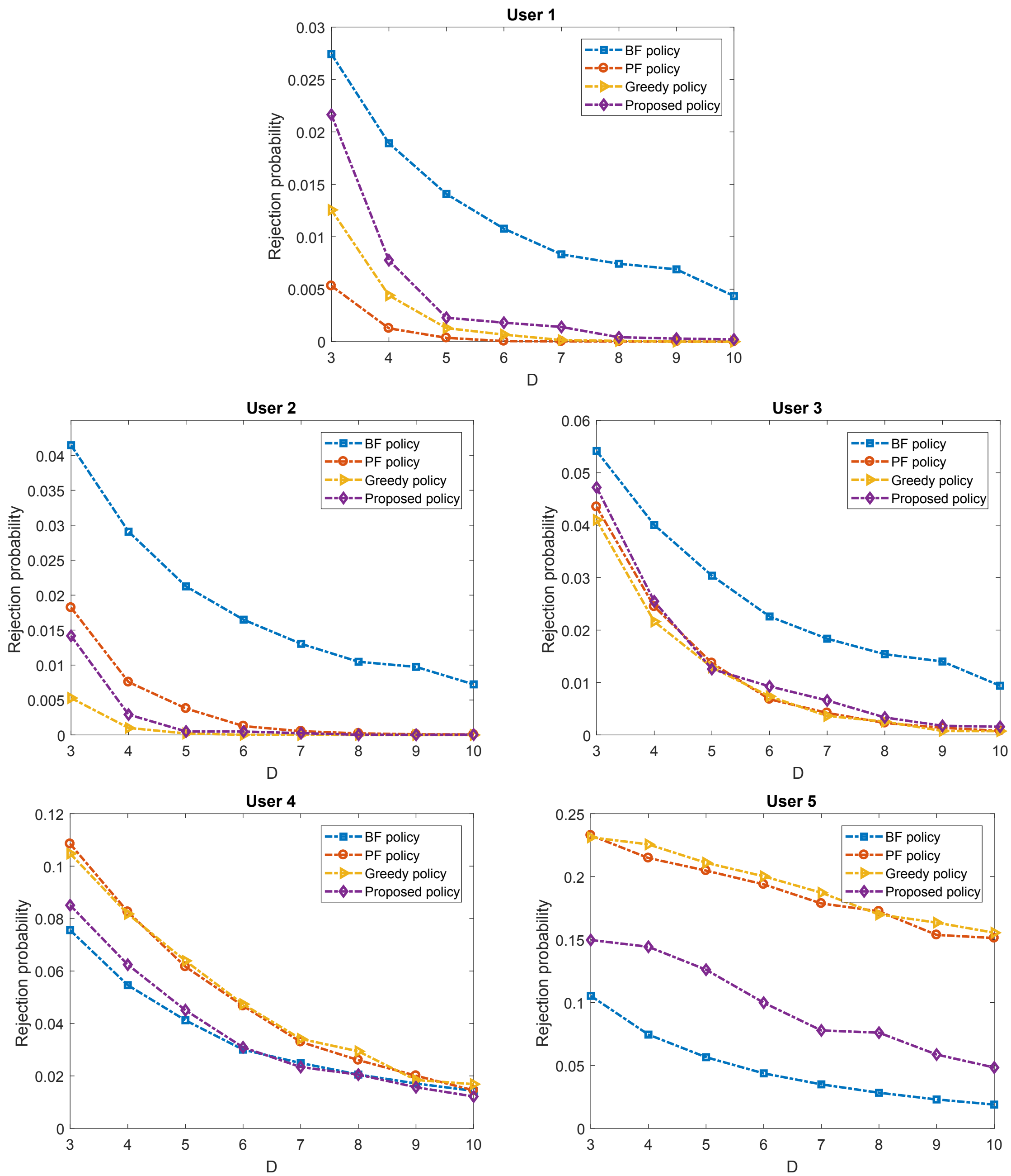

Figure 4.6: The rejection probability of different user's task buffer versus the size (i.e., D) of the task buffer.

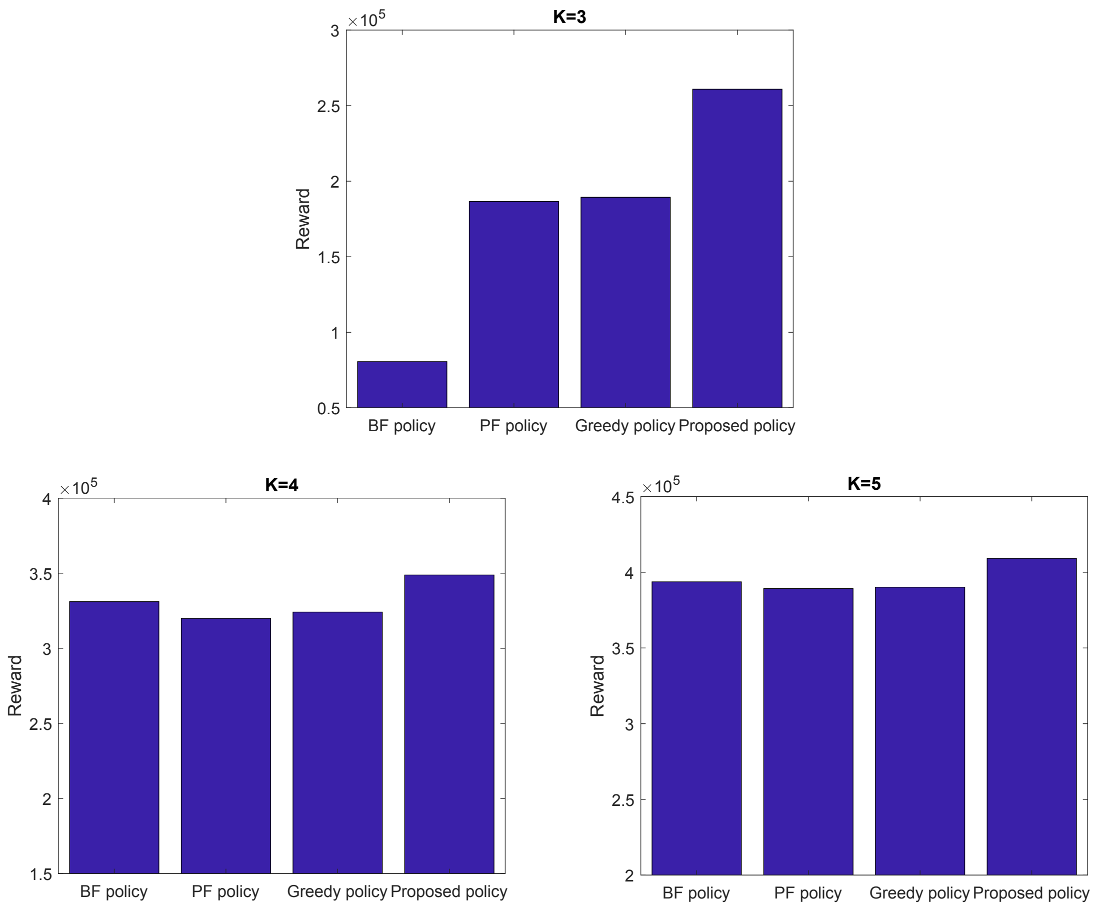

Figure 4.7: The reward verse the number (i.e., K) of channels

in Fig. 4.7. It can be seen that our proposed policy can achieve the largest reward no matter what the number of available channels is. With the growth of the number of available channels, the obtained reward keeps increasing. It confirms our intuitive understanding that larger number of channels the network has, less tasks would be rejected and larger reward can be received. When the number of available channels grows to a large value (e.g., $K = 5$), the reward of different policies are similar. The reason is that the number of free channels can satisfy the input rate of tasks, and thus, almost all tasks can be transmitted in any opportunistic scheduling scheme. When the number of available channels takes a small value (e.g., $K = 3$), the BF policy receives the smallest reward. The PF policy receives the smallest reward when the number of available channels takes a large value (e.g., $K = 4$). The reason is stated as follows. When the number of available channels takes a small value, the number of rejected tasks is large for any policy. However, the BF policy drops tasks without considering the reward of different tasks, and thus, a lot of tasks with large value of g_i are rejected and dropped. Therefore, the BF policy receives the smallest reward. When the number of available channels takes a large value, the number of rejected tasks of the BF policy and our proposed policy are much smaller than the PF policy, and thus, the PF policy receives the smallest reward.

4.6 Conclusion

In this chapter, we propose two methods, named model-based method and model-free method, to derive the optimal opportunistic scheduling policy under different scenarios. For the case with a fully explored network, a model-based scheduling method is proposed. A model-free scheduling method is proposed for the partially explored network.

Chapter 5

Distributed Opportunistic Scheduling in Cooperative Networks with RF Energy Harvesting

In this chapter, the problem of distributed opportunistic channel access in wireless cooperative networks is investigated. To cope with the energy limitation problem of relay nodes, radio-frequency (RF) energy harvesting is considered, and thus, no external energy is needed for each relay node. Then, a distributed opportunistic scheduling scheme is proposed. In the scheme, users contend for the channel access opportunity by random access, and then, the user with a successful contention makes a decision whether to give up the opportunity after probing the source-to-relay link and relay-to-destination link. To maximize the average throughput of the network, an optimal strategy of the proposed scheme is derived by optimal stopping theory. The obtained optimal strategy has a threshold-based structure, and thus, it is easy to implement in practice. To derive the threshold, an algorithm is proposed to derive the stationary probability distribution of the energy level for each relay, and then, the threshold can be calculated off-line by a proposed iterative algorithm.

5.1 Introduction

In distributed wireless networks, multiple users generally need to contend for transmission. For example, in cognitive radio networks, multiple secondary users are generally required to contend for the transmission opportunity when a primary channel is sensed as free [91]. Accordingly, the opportunistic scheduling scheme is proposed to maximize the average throughput of the network by allocating the channel access opportunities to users with good channel condition [25].

In centralized wireless networks, the channel state information (CSI) of each user can be collected by the central entity, e.g., the base station. Hence, the user which is allocated to utilize a free channel is easy to be selected by the central entity. In [28], an adaptive centralized opportunistic scheduling is proposed to maximize the profits of the network. In addition, the case with imperfect CSI is also investigated. However, the communication overhead to obtain the CSI of all users may be intolerable in some scenarios. Moreover, in distributed wireless networks (e.g., *ad hoc* networks), a central entity may not exist. To address these issues, distributed opportunistic scheduling (DOS) is introduced, and several DOS schemes have been proposed. In DOS schemes, users contend for the channel access opportunity, and then, a user with a successful contention decides whether to exploit or give up the channel access opportunity. In [30], a pure threshold scheduling policy is derived. In this policy, the user who obtains a channel access opportunity would give up the channel access opportunity if the transmission rate is lower than a certain threshold λ. The optimal stopping theory is adopted to derive the value of λ. A DOS scheme which jointly optimizes the throughput and the fairness of each user is proposed in [78]. Similar to [30], the access probability and the threshold λ are optimized in this scheme. In [92], a DOS scheme with quality-of-service (QoS) constraints is proposed. In addition, it considers hybrid links in wireless networks. In [93], a DOS framework for single-hop *ad hoc* networks is introduced. In this framework, the sources are assumed to obtain energy by a renewable energy source. Then, the scheduling policy is derived by one-dimension search. A transmission scheduling problem for the Internet of Things (IoT) is formulated in [82]. To obtain an optimal strategy for transmission scheduling, a reinforcement learning method, i.e., Q-learning algorithm, is proposed in this work. Then, a deep learning model is adopted to accelerate the algorithm. In [80], a greedy scheduling policy, which focuses on immediate reward maximization, is proposed. Then, the conditions which guarantee the optimality of the proposed greedy policy are derived.

Since cooperative communications have emerged as a promising technique to enhance communication efficiency, DOS in cooperative networks receives more and more attention. In [31], a DOS scheme is proposed in distributed networks with decode-and-forward (DF) relay. The user with a successful contention decides to give up a transmission opportunity or transmit its data to the relay by probing the channel condition of the source-to-relay link. If the user transmits its data to the relay, the relay then keeps probing the relay-to-destination link until the achievable rate is not less than the rate of the source-to-relay link. Another DOS scheme in cooperative networks is proposed in [32]. Compared to the scheme in [31], the user who wins a successful contention has one more option, further probing. Thus, it benefits the network throughput, at the cost of possible complexity of implementation. The DOS problem in cooperative networks with amplify-and-forward (AF) relay is investigated in [95]. In [94], a resource (e.g., link, power, etc.) allocation problem is considered to maximize the number of successful task transmissions in a cooperative satellite networks. Then, the problem is formulated to a mixed-integer nonlinear program (MINLP) optimization problem, and then, an algorithm is proposed to solve the formulated problem.

Although some DOS schemes have been proposed in cooperative networks, energy constraint is rarely considered. In reality, relay nodes are usually battery limited [43], and thus, periodic replacement or recharging for the battery of relay nodes is needed. However, it may not be feasible in lots of scenarios [44]. As a promising solution to encourage the relay to provide cooperative services and prolong the lifetime of energy constrained wireless networks, energy harvesting technique attracts much attention [45]. By harvesting energy from the outside environments (e.g., solar, wind, and radio-frequency (RF) signal), relay nodes can forward the received data to destinations without external energy [46]. Compared to other sources, RF signal is a kind of predictable and controllable source. Thus, energy harvesting from RF signals is widely used in wireless networks, especially in cooperative networks [47]. Although wireless cooperative networks with energy harvesting attract more and more attention, there are still no efforts on designing optimal DOS schemes in wireless cooperative networks with RF energy harvesting.

In this work, we investigate the opportunistic scheduling problem in distributed cooperative networks with RF energy harvesting. The major contributions of this work are summarized as follows.

1. A DOS scheme is proposed in a cooperative network with RF energy harvesting relays. No external energy is needed for the relay nodes. In the scheme, users contend for the transmission opportunity by random access, and then, the user with a successful contention makes a decision whether to give up the opportunity after probing the source-to-relay link and relay-to-destination link.

2. To maximize the average throughput of the network, the optimal strategy for the winner user is derived. In addition, the obtained optimal strategy has a threshold-based structure, and thus, it is easy to implement in reality. The method to calculate the threshold is also proposed as follows. First, a low complexity algorithm is proposed to derive the stationary probability distribution of the energy level for each relay. Then, the threshold can be calculated off-line by a proposed iterative algorithm. Accordingly, the user who obtains the channel access opportunity can make a fast decision by comparing the achievable transmission rate with the threshold.

3. Performance analysis is conducted for the proposed scheme by simulation. It shows that our proposed DOS scheme can obtain much more average network throughput compared with benchmark DOS schemes.

The rest of this chapter is organized as follows. Section 5.2 gives the system model and proposes the DOS scheme. The optimal strategy of the proposed DOS scheme is derived in Section 5.3. Section 5.4 shows simulation results. Finally, Section 5.5 concludes this chapter.

5.2 System Model and Proposed Schemes

The system model is given in this section and also with the proposed DOS scheme.

5.2.1 System Model

We consider a distributed cooperative network with DF relaying. In the network, there are I source-destination pairs and one available channel for transmission. A halfduplex relay is assigned for each pair[1]. To contend for the channel access opportunity, random access, which is easy to implement in reality, is adopted by each source [70]. For each channel contention, each source transmits a request-to-send (RTS) packet with a probability p_c, and does not transmit anything with probability $1 - p_c$. A successful channel contention occurs when only one source transmits RTS. Thus, a successful channel contention happens with probability $Ip_{\mathrm{c}}(1-p_{\mathrm{c}})^{(I-1)}$. In a successful channel contention, the source who transmits RTS is the winner source, and thus, gets the channel access opportunity. If a channel contention is unsuccessful (i.e., no source transmits RTS, or a collision occurs), all the I sources re-contend in subsequent channel contentions. For each relay, the energy for forwarding the data is harvested from the RF signal of sources. A rechargeable battery with capacity B_{max} is equipped to each relay. The battery of each relay is quantized to L levels, with the amount, denoted $\sigma = \frac{B_{\mathrm{max}}}{L}$, of energy for each level. The harvested energy is stored to the battery. If source $i \in \{1, 2, \cdots, I\}$ is the winner source and decides to transmit its data, the relays except relay i can harvest energy from the RF signal of source i. Relay i consumes energy for forwarding the received data to destination i. Note that the energy consumption of operations other than data transmission is assumed to be negligible. Thus, the residual energy of each relay would be variant only for a successful transmission[2]. The transmission power of a source is set as P_{s}. Let h_{ij} and g_{ji} denote the channel gain from source i to relay j and the channel gain from relay j to destination i, respectively. For the case $i \neq j$, it is assumed that h_{ij} and g_{ji} follow a complex Gaussian distribution with zero mean and a variance δ_{h}^2 and δ_{g}^2, respectively. Similarly, for the case $i = j$, h_{ij} and g_{ji} are assumed to follow the same distribution with the former case, but with a different variance being ν_{h}^2 and ν_{g}^2, respectively. The background noise is assumed to be Gaussian with zero mean and unit variance. The coherence time of the channel is denoted as τ_{tx}. In other words, the channel gains h_{ij} and g_{ji} remain constant during a slot with duration τ_{tx}, but vary independently from a slot to another.

[1]In the following parts of this chapter, each pair means each source-destination pair.
[2]A successful transmission means a source wins the channel contention and starts a transmission.

5.2.2 The Proposed DOS Scheme

Define the tth cycle as the period from the end of the $(t-1)$th successful transmission to the end of the tth successful transmission. During the tth cycle, if source i is the winner source (the Nth winner in the tth cycle[3]), relay i can receive source i's RTS, and can estimate the channel gain of its first-hop (i.e., the link from source i to relay i). After that, relay i transmits an RTS to destination i, which accordingly replies with a clear-to-send (CTS) packet. By using the CTS, relay i can estimate the channel gain of its second-hop (i.e., the link from relay i to destination i). Without loss of generality, we assume the RTS and CTS packets have the same transmission duration denoted as τ_c. Then, relay i selects one of the following two options.

1. Relay i decides to give up the current transmission opportunity, and thus, it transmits a CTS to all sources to announce the decision. After that, all sources start the next channel contention immediately.

2. Relay i decides to exploit the current transmission opportunity, and thus, it transmits a CTS to source i to announce the decision. Subsequently, we have two transmissions each with duration $\frac{\tau_{\text{tx}}}{2}$. The first transmission is from source i to relay i at a rate $R_{i,N}(t)$, while the second transmission is from relay i to destination i with a transmission power $P^{\text{r}}_{i,N}(t) = \min\left\{\frac{P_{\text{s}}\left|h_{ii,N}(t)\right|^2}{\left|g_{ii,N}(t)\right|^2}, \frac{2l_i(t)\sigma}{\tau_{\text{tx}}}\right\}$, where $h_{ii,N}(t)$ is the channel gain between source i and relay i, $g_{ii,N}(t)$ is the channel gain between relay i and destination i, and $l_i(t) \in \{0, 1, 2, \cdots, L\}$ denotes the residual energy level of relay i at the beginning of the tth cycle. Thus, the transmission rate of the tth cycle is derived as $R_{i,N}(t) = \log_2\left(1 + P^{\text{r}}_{i,N}(t)|g_{ii,N}(t)|^2\right)$. At the end of the tth cycle, the residual battery level, denoted $l_i(t+1)$, of relay i is $l_i(t+1) = l_i(t) - \left\lceil\frac{P^{\text{r}}_{i,N}(t)\tau_{\text{tx}}}{2\sigma}\right\rceil$, where $\lceil x \rceil$ means the ceiling function. The relays except relay i can harvest energy from the RF signal of source i. For relay j ($j \neq i$), the amount of harvested energy is $B_{j,N}(t) = \frac{\beta P_{\text{s}}\left|h_{ij,N}(t)\right|^2\tau_{\text{tx}}}{2}$ where β denotes the energy conversion efficiency [96]. Thus, the residual battery level of relay j after the tth cycle is $l_j(t+1) = l_j(t) + l^{\text{h}}_{j,N}(t)$ where $l^{\text{h}}_{j,N}(t) = \min\left\{\left\lfloor\frac{B_{j,N}(t)}{\sigma}\right\rfloor, L - l_j(t)\right\}$ and $\lfloor x \rfloor$ means the floor function.

[3] It means that the former $(N-1)$ winner sources give up the transmission opportunity in the tth cycle.

An example of this scheme is given in Fig. 5.1. In order to estimate the CSI of the source-to-relay link and the relay-to-destination link, two RTS packets and two CTS packets need to be sent. Thus, the duration of the two periods, 'win and give up' and 'win and transmit', is $4\tau_{\mathrm{c}}$.

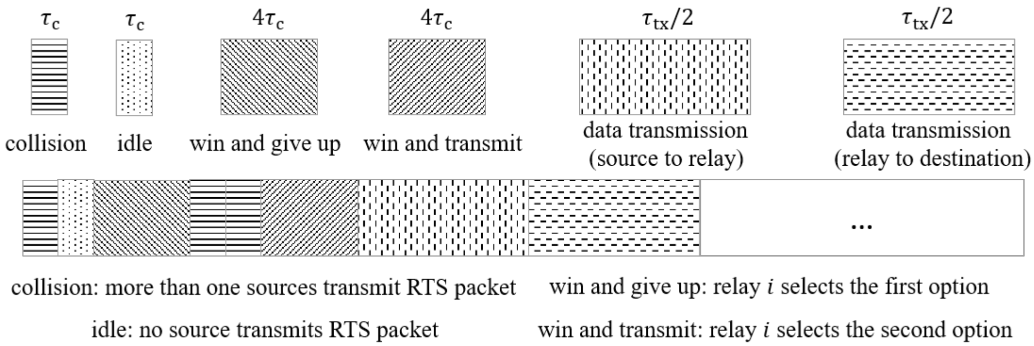

Figure 5.1: An example of the proposed DOS scheme.

In the proposed DOS scheme, relay i needs to decide whether to stop (i.e., start a transmission) or continue (i.e., give up the transmission opportunity) according to the channel condition and its residual energy. If a transmission is started under the case with poor channel condition and low residual energy, a throughput degradation is caused due to the low transmission rate. On the other hand, if a transmission is started only for the case with good channel condition and large residual energy, a cost in terms of the probing time, which is used to explore the channel, is caused. Accordingly, there is a tradeoff between the current transmission throughput and the cost for channel probing. In this work, to maximize the average network throughput, we try to derive an optimal stopping strategy for the proposed DOS scheme. In the following section, the optimal stopping strategy of the proposed DOS scheme is derived.

5.3 Stopping Strategy of The Proposed DOS Scheme

In this section, we target to derive the optimal stopping strategy for the proposed DOS scheme. According to the channel contention in the proposed DOS scheme, the expected duration of generating a winner source $i \in \{1, 2, \cdots, I\}$ is $\bar{\tau}_0 = \frac{1-Ip_{\mathrm{c}}(1-p_{\mathrm{c}})^{I-1}}{Ip_{\mathrm{c}}(1-p_{\mathrm{c}})^{I-1}}\tau_{\mathrm{c}} + 4\tau_{\mathrm{c}} = (\frac{1}{Ip_{\mathrm{c}}(1-p_{\mathrm{c}})^{I-1}} + 3)\tau_{\mathrm{c}}$. After a successful contention (denoted the Nth successful contention), relay i decides to continue (i.e., give up the transmission opportunity and starts a new contention) or stop (i.e., start the tth successful transmission). If relay i selects the stop action, the traffic volume which can be sent in the tth successful transmission is $Y_{i,N}(t) = \frac{R_{i,N}(t)\tau_{\text{tx}}}{2}$. Thus, the average network throughput is expressed as $\lambda = \lim_{t' \to \infty} \frac{\sum_{t=1}^{t'} Y_{i,N}(t)}{\sum_{t=1}^{t'} T_{i,N}(t)} = \frac{E[Y_{i,N}]}{E[T_{i,N}]}$, where $E[\cdot]$ means expectation and $T_{i,N}(t)$ is the total time from the end of the $(t-1)$th cycle to the end of the tth cycle. $T_{i,N}(t)$ can be calculated by $T_{i,N}(t) = \sum_{n=1}^{N} \tau_0^n + \tau_{\text{tx}}$ where τ_0^n (with mean $\bar{\tau}_0$) is the duration to generate the nth winner source. For each successful transmission, N is called the stopping rule. To maximize the average network throughput, the optimal stopping strategy is to find the optimal stopping rule, denoted N^*. Thus, the optimization problem is formulated as

$$P1: \quad N^* = \arg\sup_{N \in \mathbf{D}} \frac{E[Y_{i,N}]}{E[T_{i,N}]} \tag{5.1}$$

where $\mathbf{D} = \{N : N \geq 1, E[T_{i,N}] < \infty\}$ denotes the collection of all stopping rules. Hence, the maximum average network throughput of the proposed DOS scheme, denoted λ^*, is

$$\lambda^* = \sup_{N \in \mathbf{D}} \frac{E[Y_{i,N}]}{E[T_{i,N}]}. \tag{5.2}$$

Instead of solving Problem P1, the following problem, which is an equivalent problem to Problem P1 [49], is considered.

$$P2: \quad N^* = \arg\sup_{N \in \mathbf{D}} E[Y_{i,N} - \lambda^* T_{i,N}] \tag{5.3}$$

where λ^* satisfies $V^*(\lambda^*) = \sup_{N \in \mathbf{D}} E[Y_{i,N} - \lambda^* T_{i,N}] = 0$.

According to the expression of $Y_{i,N}(t)$ and $T_{i,N}(t)$, we have

$$V^*(\lambda^*) = \sup_{N \in \mathbf{D}} E\left[\frac{R_{i,N}\tau_{\text{tx}}}{2} - \lambda^* \left(\sum_{n=1}^{N} \tau_0^n + \tau_{\text{tx}}\right)\right] \tag{5.4}$$

where $R_{i,N} = \log_2\left(1 + P_{i,N}^{\text{r}} |g_{ii,N}|^2\right)$ and $P_{i,N}^{\text{r}} = \min\left\{\frac{P_s|h_{ii,N}|^2}{|g_{ii,N}|^2}, \frac{2l_i\sigma}{\tau_{\text{tx}}}\right\}$.[4]

Before deriving the optimal stopping rule N^*, two conditions (5.5) and (5.6) [49] should be satisfied such that an optimal solution of Problem P2 exists for $\forall\lambda > 0$.

$$C1. \quad E[\sup_{N \in \mathbf{D}} Y_{i,N} - \lambda T_{i,N}] < \infty \tag{5.5}$$

[4]Since the expectation of $Y_{i,N}$ and $T_{i,N}$ is considered to derive N^*, (t) is taken out in the expression.

$$C2. \quad \limsup_{N\to\infty}\{Y_{i,N} - \lambda T_{i,N}\} \leq Y_{i,\infty} - \lambda T_{i,\infty} = -\infty. \tag{5.6}$$

Then, we have the following lemma.

Lemma 8. *The first condition $C1$ is satisfied for Problem P2 since the expression $E\left[\sup_{N\in\mathbf{D}}\left\{\frac{R_{i,N}\tau_{tx}}{2} - \lambda\left(\sum_{n=1}^{N}\tau_0^n + \tau_{tx}\right)\right\}\right] < \infty$ is satisfied. The second condition $C2$ is satisfied for Problem P2 since the expression $\limsup_{N\to\infty}\left\{\frac{R_{i,N}\tau_{tx}}{2} - \lambda\left(\sum_{n=1}^{N}\tau_0^n + \tau_{tx}\right)\right\} = -\infty$ is satisfied.*

The proof is similar to that in [95], and thus, is omitted here.

According to Lemma 8, the optimal stopping rule N^* exists. To derive the rule N^*, the following lemma is given.

Lemma 9. *The optimal stopping rule N^* is given by $N^* = \min\{N \geq 1 : R_{i,N} \geq 2\lambda^*\}$, where λ^* is the maximum average network throughput and is the unique solution of*

$$E\left[\left(\frac{R_{i,N}\tau_{\text{tx}}}{2} - \lambda\tau_{\text{tx}}\right)^+\right] = \lambda\bar{\tau}_0. \tag{5.7}$$

Lemma 9 is easy to be proven by the principle of optimality in stopping rule problems [49], and thus, the proof is omitted here.

Although λ^* is the unique solution of (5.7), it is still difficult to solve this equation directly. Thus, we introduce the following lemma to calculate λ^* efficiently.

Lemma 10. *For any nonnegative λ_0, the fixed-point iteration $\lambda_{z+1} = \frac{E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\text{tx}}}{2}\right]}{E\left[\sum_{n=1}^{N^*_{\lambda_z}}\tau_0^n + \tau_{\text{tx}}\right]}$ for $z = \{0, 1, \cdots\}$ would converge to λ^*, where $N^*_{\lambda_z} = \min\{N \geq 1 : R_{i,N} \geq 2\lambda_z\}$, $E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\text{tx}}}{2}\right] = \sum_{l_i=0}^{L} p_{l_i}\frac{\tau_{\text{tx}}}{2}\frac{2\lambda_z\exp\left(-S_0\left(2^{2\lambda_z}-1\right)\right)+\exp(S_0)E_1\left(S_0 2^{2\lambda_z}\right)/\ln 2}{\exp\left(-S_0\left(2^{2\lambda_z}-1\right)\right)}$, $E\left[\sum_{n=1}^{N^*_{\lambda_z}}\tau_0^n + \tau_{\text{tx}}\right] = \sum_{l_i=0}^{L} p_{l_i}\frac{\bar{\tau}_0}{\exp\left(-S_0\left(2^{2\lambda_z}-1\right)\right)} + \tau_{\text{tx}}$, $S_0 = \frac{2l_i\sigma\nu_{\text{g}}+P_{\text{s}}\nu_{\text{h}}\tau_{\text{tx}}}{2l_i\sigma\nu_{\text{g}}P_{\text{s}}\nu_{\text{h}}}$, p_{l_i} is the stationary probability that the battery level of relay i is l_i, and $E_1(\cdot)$ is the exponential integral function.*

Proof. we define a function $\rho(\lambda_z) = \frac{E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\text{tx}}}{2}\right]}{E\left[\sum_{n=1}^{N^*_{\lambda_z}}\tau_0^n + \tau_{\text{tx}}\right]}$. According to Lemma 9, λ^* is the unique solution of $\rho(\lambda) = \lambda$. According to (5.2), λ^* is also the maximum value of function $\rho(\lambda)$. Then, due to $\rho(0) > 0$, we have $\rho(\lambda) > \lambda$ for $\lambda < \lambda^*$ and $\rho(\lambda) < \lambda$ for $\lambda > \lambda^*$. We discuss three cases according to the value of λ_0.

If $\lambda_0 < \lambda^*$, $\lambda_{z+1} = \rho(\lambda_z) \leq \rho(\lambda^*) = \lambda^*$ for $z = \{0, 1, 2, \cdots\}$ is obtained. In addition, we also have $\lambda_{z+1} = \rho(\lambda_z) > \lambda_z$ for any $z \geq 0$. Thus, $\{\lambda_z\}$ is an increasing set with an upper bound λ^*. Due to $\lim\limits_{z\to\infty}(\lambda_{z+1} - \lambda_z) = \lim\limits_{z\to\infty}(\rho(\lambda_z) - \lambda_z) = \rho(\lambda_\infty) - \lambda_\infty = 0$, $\rho(\lambda_\infty) = \lambda_\infty$ is obtained. Since λ^* is the unique solution of $\lambda = \rho(\lambda)$, $\lambda_\infty = \lambda^*$ is obtained.

If $\lambda_0 > \lambda^*$, $\lambda_1 = \rho(\lambda_0) < \lambda^*$. Thus, this case is equivalent to the case $\lambda_0 < \lambda^*$.

If $\lambda_0 = \lambda^*$, $\lambda_z = \lambda^*$ for any $z \geq 0$.

In summary, λ_z would converge to λ^* by the iteration $\lambda_{z+1} = \rho(\lambda_z)$. Next, how to calculate $\rho(\lambda_z)$ is analyzed in the following.

For the term $E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\rm tx}}{2}\right]$, we have

$$\begin{aligned} E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\rm tx}}{2}\right] &= \frac{\tau_{\rm tx}}{2}E\left[R_{i,N^*_{\lambda_z}}\right] \\ &= \frac{\tau_{\rm tx}}{2}E\left[R_{i,N}\left|R_{i,N} \geq 2\lambda_z\right.\right] \\ &= \sum_{l_i=0}^{L} p_{l_i}\frac{\tau_{\rm tx}}{2}\frac{\int_{2\lambda_z}^{\infty} r dF_{R_{i,N}}(r)}{1-F_{R_{i,N}}(2\lambda_z)} \end{aligned}$$

where $F_{R_{i,N}}(r)$ is the cumulative distribution function (CDF) of the achievable rate $R_{i,N}$. The term p_{l_i} is derived in Lemma 11. For the term $F_{R_{i,N}}(r)$, we have

$$\begin{aligned} F_{R_{i,N}}(r) &= p\left(R_{i,N} \leq r\right) \\ &= 1 - p\left(R_{i,N} > r\right) \\ &= 1 - \exp\left(-\frac{2l_i\sigma\nu_{\rm g} + P_{\rm s}\nu_{\rm h}\tau_{\rm tx}}{2l_i\sigma\nu_{\rm g}P_{\rm s}\nu_{\rm h}}\left(2^r - 1\right)\right) \\ &= 1 - \exp\left(-S_0\left(2^r - 1\right)\right). \end{aligned}$$

Therefore, we have

$$E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\rm tx}}{2}\right] = \sum_{l_i=0}^{L} p_{l_i}\frac{\tau_{\rm tx}}{2}\frac{2\lambda_z\exp\left(-S_0\left(2^{2\lambda_z}-1\right)\right)+\exp(S_0)E_1\left(S_0 2^{2\lambda_z}\right)/\ln 2}{\exp\left(-S_0\left(2^{2\lambda_z}-1\right)\right)}.$$

Similarly, for the term $E\left[\sum\limits_{n=1}^{N^*_{\lambda_z}}\tau_0^n + \tau_{\rm tx}\right]$, we have

$$\begin{aligned} E\left[\sum_{n=1}^{N^*_{\lambda_z}}\tau_0^n + \tau_{\rm tx}\right] &= E\left[N^*_{\lambda_z}\right]\bar{\tau}_0 + \tau_{\rm tx} \\ &= \sum_{l_i=0}^{L} p_{l_i}\frac{1}{1-F_{R_{i,N}}(2\lambda_z)}\bar{\tau}_0 + \tau_{\rm tx} \\ &= \sum_{l_i=0}^{L} p_{l_i}\frac{\bar{\tau}_0}{\exp\left(-S_0\left(2^{2\lambda_z}-1\right)\right)} + \tau_{\rm tx}. \end{aligned}$$

This completes the proof. $\square$

According to Lemma 10, the value of $p_{l_i=d}$ for $d \in \{0, 1, 2, \cdots, L\}$ is needed to derive the threshold λ^*. Accordingly, we will propose an algorithm to calculate the stationary probability $p_{l_i=d}$.

In the proposed DOS scheme, the battery level of each relay would be variant only for a successful transmission. We model the battery level of each relay as a Markov chain, denoted M_1. For convenience, we omit the index N when deriving the stationary probability. For any relay $i \in \{1, 2, \cdots, I\}$, let $l_i(t) = u, u \in \{0, 1, 2, \cdots, L\}$ and $l_i(t+1) = v, v \in \{0, 1, 2, \cdots, L\}$ denote the battery level at the beginning of the tth cycle and $(t+1)$th cycle, respectively. Then, the transition probability from $l_i(t) = u$ to $l_i(t+1) = v$, denoted $p_{u \to v}(t)$, is derived as follows.

1. $v > u$ and $v < L$. It means that a pair $j (j \neq i)$ occupies the channel for the tth successful transmission. Let $\Xi_j(t)$ denote the event which the tth successful transmission is occupied by pair j. In addition, relay i gets the amount, denoted $(v-u)\sigma$, of energy by energy harvesting. Thus, the transition probability is given as (5.8), where $p(\Xi_j(t))$ denotes the probability that event $\Xi_j(t)$ occurs.

$$\begin{aligned} & p_{u \to v}(t) \\ & = \sum_{j \neq i} p(\Xi_j(t)) p\left(v - u \leq l_i^h(t) < v + 1 - u\right) \\ & = (I-1) p_c (1-p_c)^{I-1} p(R_j(t) \geq 2\lambda^*) p\left(\tfrac{2\sigma(v-u)}{\beta P_s \tau_{tx}} \leq |h_{ji}(t)|^2 < \tfrac{2\sigma(v+1-u)}{\beta P_s \tau_{tx}}\right). \end{aligned} \tag{5.8}$$

2. $v > u$ and $v = L$. It means that a pair $j (j \neq i)$ occupies the channel for the tth successful transmission. Relay i gets at least the amount, denoted $(L-u)\sigma$, of energy by energy harvesting. Thus, the transition probability is given as (5.9).

$$\begin{aligned} & p_{u \to v}(t) \\ & = \sum_{j \neq i} p(\Xi_j(t)) p\left(l_i^h(t) \geq (L-u)\right) \\ & = (I-1) p_c (1-p_c)^{I-1} p(R_j(t) \geq 2\lambda^*) p\left(|h_{ji}(t)|^2 \geq \tfrac{2\sigma(L-u)}{\beta P_s \tau_{tx}}\right). \end{aligned} \tag{5.9}$$

3. $v = u$. It means that a pair $j (j \neq i)$ occupies the channel for the tth successful transmission. Relay i gets less than the amount, denoted σ, of energy by energy

harvesting. Thus, the transition probability is given as (5.10).

$$\begin{aligned} & p_{u\to v}(t) \\ & = \sum_{j\neq i} p(\Xi_j(t))p\left(l_i^h(t) < 1\right) \\ & = (I-1)p_c(1-p_c)^{I-1}p(R_j(t)\geq 2\lambda^*)p\left(|h_{ji}(t)|^2 < \tfrac{2\sigma}{\beta P_s \tau_{tx}}\right). \end{aligned} \tag{5.10}$$

4. $v < u$ and $v > 0$. It means that pair i occupies the channel for the tth successful transmission. Relay i consumes the amount, denoted $(u-v)\sigma$, of energy for the tth successful transmission. Thus, the transition probability is given as (5.11).

$$\begin{aligned} & p_{u\to v}(t) \\ & = p(\Xi_i(t))p\left(\tfrac{2\sigma(u-v-1)|g_{ii}(t)|^2}{\tau_{tx}} < P_s|h_{ii}(t)|^2 \leq \tfrac{2\sigma(u-v)|g_{ii}(t)|^2}{\tau_{tx}}\,|R_i(t)\geq 2\lambda^*\right) \\ & = p_c(1-p_c)^{I-1}p\left(R_i(t)\geq 2\lambda^*\right)p\left(\tfrac{2\sigma(u-v-1)}{\tau_{tx}P_s} < \tfrac{|h_{ii}(t)|^2}{|g_{ii}(t)|^2} \leq \tfrac{2\sigma(u-v)}{\tau_{tx}P_s}\,|R_i(t)\geq 2\lambda^*\right). \end{aligned} \tag{5.11}$$

5. $v < u$ and $v = 0$. It means that pair i occupies the channel for the tth successful transmission. Relay i consumes the amount, denoted $u\sigma$, of energy for the tth successful transmission. Thus, the transition probability is given as (5.12).

$$\begin{aligned} & p_{u\to v}(t) \\ & = p(\Xi_i(t))p\left(P_s|h_{ii}(t)|^2 > \tfrac{2\sigma(u-1)|g_{ii}(t)|^2}{\tau_{tx}}\,|R_i(t)\geq 2\lambda^*\right) \\ & = p_c(1-p_c)^{I-1}p\left(R_i(t)\geq 2\lambda^*\right)p\left(\tfrac{|h_{ii}(t)|^2}{|g_{ii}(t)|^2} > \tfrac{2\sigma(u-1)}{\tau_{tx}P_s}\,|R_i(t)\geq 2\lambda^*\right). \end{aligned} \tag{5.12}$$

Let $\Pi_i \triangleq \{p_{l_i=0}, p_{l_i=1}, p_{l_i=2}, \cdots, p_{l_i=L}\}$ denote the stationary distribution of battery levels for relay i. Then, we consider the following iteration $\Pi_i(\kappa+1) = \Pi_i(\kappa)\mathbf{P}_i(\kappa)$, where κ denotes the index of the iteration, $\Pi_i(\kappa)$ denotes the probability distribution of battery levels for relay i at the κth iteration, and $\mathbf{P}_i(\kappa)$ denotes the transition probability matrix of energy levels for relay i at the κth iteration. Accordingly, we have $\Pi_i(\kappa) = \{p_{l_i=0}(\kappa), p_{l_i=1}(\kappa), p_{l_i=2}(\kappa), \cdots, p_{l_i=L}(\kappa)\}$ and $\mathbf{P}_i(\kappa) = \{p_{u\to v}(\kappa), \forall u \in \{0,1,2,\cdots,L\}, \forall v \in \{0,1,2,\cdots,L\}\}$. Then, $\mathbf{P}_i(\kappa)$ can be obtained according to (5.8)-(5.12). The matrix $\mathbf{P}_i(\kappa)$ is a stochastic matrix with dimension $(L+1)\times(L+1)$. However, $\mathbf{P}_i(\kappa)$ varies with the variation of κ, and thus, there is no standard method to calculate the stationary distribution Π_i. In this work, we propose the following algorithm, denoted Algorithm 5.1, to derive the stationary distribution Π_i, where $\|\Pi_i(\kappa) - \Pi_i(\kappa-1)\| = \max\limits_{0\leq d\leq L} |p_{l_i=d}(\kappa) - p_{l_i=d}(\kappa-1)|$. Then, Lemma 11 is introduced to demonstrate that Algorithm 5.1 can obtain the unique stationary distribution Π_i.

Algorithm 5.1 Calculate the stationary distribution of relay i's battery level

Initialize $\Pi_i(0)$, ς_0, and $\kappa = 0$,
loop:
Compute $\mathbf{P}_i(\kappa)$ by (5.8)-(5.12),
Compute $\Pi_i(\kappa+1) = \Pi_i(\kappa)\mathbf{P}_i(\kappa)$,
Set $\kappa = \kappa + 1$,
if $\|\Pi_i(\kappa) - \Pi_i(\kappa-1)\| > \varsigma_0$ **then**
  goto *loop*
end if
Set $\Pi_i = \Pi_i(\kappa)$.

Lemma 11. *Given an initial value of* $\Pi_i(0)$ *and* $\varsigma_0 \to 0$, $\Pi_i(\kappa)$ *in Algorithm 5.1 would converge to the unique stationary state distribution* Π_i.

Proof. First, we construct a Markov chain (denoted M_2) where the state, denoted $\mathbf{C} \triangleq \{l_1, l_2, \cdots, l_I\}$, is the battery level of all relays. In this Markov chain, there are $(L+1)^I$ possible states. Let Φ denote the set of all possible states. If the transmission opportunity is occupied by pair i at state $\mathbf{C}$, the possible next state, denoted $\mathbf{C}' = \{l'_1, l'_2, \cdots, l'_I\}$, can be derived by

$$\mathbf{C}' = \begin{pmatrix} l_1 + l_1^h \\ l_2 + l_2^h \\ \cdots \\ l_i - l_i^r \\ \cdots \\ l_I + l_I^h \end{pmatrix}^T \tag{5.13}$$

where $l_i^r = \left\lceil \frac{P_i^r \tau_{tx}}{2\sigma} \right\rceil$.

Therefore, the transition probability from state $\mathbf{C}$ to state $\mathbf{C}'$ is derived as

$$p_{\mathbf{C}\to\mathbf{C}'} = p_c(1-p_c)^{I-1} p(R_i \geq 2\lambda^*) p_{l_i^r} \prod_{j\neq i} p_{l_j^h} \tag{5.14}$$

where $p_{l_j^h}$ and $p_{l_i^r}$ are derived as (5.15) and (5.16), respectively.

$$p_{l_j^h} = \begin{cases} p\left(|h_{ij}|^2 < \frac{2\sigma}{\beta P_s \tau_{tx}}\right), & l_j^h = 0 \\ p\left(\frac{2 l_j^h \sigma}{\beta P_s \tau_{tx}} \leq |h_{ij}|^2 < \frac{2\left(l_j^h+1\right)\sigma}{\beta P_s \tau_{tx}}\right), & 0 < l_j^h < L - l_j \\ p\left(|h_{ij}|^2 \geq \frac{2 l_j^h \sigma}{\beta P_s \tau_{tx}}\right), & l_j^h(t) = L - l_j \end{cases} \tag{5.15}$$

$$p_{l_i^r} = \begin{cases} p\left(\frac{2\left(l_i^r-1\right)\sigma}{P_s \tau_{tx}} < \frac{|h_{ii}|^2}{|g_{ii}|^2} \leq \frac{2 l_i^r \sigma}{P_s \tau_{tx}} \,|R_i \geq 2\lambda^*\right), & l_i^r < l_i \\ p\left(\frac{|h_{ii}|^2}{|g_{ii}|^2} > \frac{2\left(l_i^r-1\right)\sigma}{P_s \tau_{tx}} \,|R_i \geq 2\lambda^*\right), & l_i^r = l_i \end{cases} \tag{5.16}$$

Thus, for any state $\mathbf{C} \in \Phi$, the transition probability to any possible state $\mathbf{C}' \in \Phi$ can be calculated according to (5.13), (5.14), (5.15), and (5.16). Then, the transition probability matrix of this Markov chain, denoted $\mathbf{P_C}$, with dimension $(L+1)^I \times (L+1)^I$ can be obtained. Note that $\mathbf{P_C}$ is invariant. It is easy to know that Markov chain M_2 is aperiodic and has only one closed communicating class. Thus, a unique stationary state distribution, denoted $\Gamma_{\mathbf{C}} \triangleq \{p_{\mathbf{C}}, \forall \mathbf{C} \in \Phi\}$, of this Markov chain exists and the equation $\Gamma_{\mathbf{C}} = \Gamma_{\mathbf{C}} \mathbf{P_C}$ holds. In addition, given an initial $\Gamma_{\mathbf{C}}(0)$, the iteration $\Gamma_{\mathbf{C}}(\kappa+1) = \Gamma_{\mathbf{C}}(\kappa)\mathbf{P_C}$ for $\kappa = \{0, 1, \cdots\}$ would converge to $\Gamma_{\mathbf{C}}$, where $\Gamma_{\mathbf{C}}(\kappa) = \{p_{\mathbf{C}}(\kappa), \forall \mathbf{C} \in \Phi\}$ denotes the probability distribution of Markov chain M_2 at the κth iteration [97].

Then, we analyze the iteration $\Gamma_{\mathbf{C}}(\kappa+1) = \Gamma_{\mathbf{C}}(\kappa)\mathbf{P_C}$. After an iteration, we can obtain $\Gamma_{\mathbf{C}}(\kappa+1)$. In addition, we can also obtain the probability $p_{l_i=v}(\kappa+1)$ where $v \in \{0, 1, 2, \cdots, L\}$ in Markov chain M_2. According to the definition of $p_{l_i=v}(\kappa+1)$, we have

$$\begin{aligned}
&p_{l_i=v}(\kappa+1) \\
&= \sum_{u=0}^{L} p_{\left\{\forall \mathbf{C} \in \Phi_{l_i=u} \rightarrow \forall \mathbf{C}' \in \Phi_{l_i=v}\right\}}(\kappa) \\
&= \sum_{u=0}^{L} p_{\left\{\forall \mathbf{C} \in \Phi_{l_i=u}\right\}}(\kappa) p_{\left\{\mathbf{C} \in \Phi_{l_i=u} \rightarrow \forall \mathbf{C}' \in \Phi_{l_i=v} \middle| \forall \mathbf{C} \in \Phi_{l_i=u}\right\}}(\kappa) \\
&= \sum_{u=0}^{L} \left\{ \sum_{\mathbf{C} \in \Phi_{l_i=u}} p_{\mathbf{C}}(\kappa) \right\} p_{\left\{\mathbf{C} \in \Phi_{l_i=u} \rightarrow \forall \mathbf{C}' \in \Phi_{l_i=v} \middle| \forall \mathbf{C} \in \Phi_{l_i=u}\right\}}(\kappa).
\end{aligned} \tag{5.17}$$

For the probability $p_{\left\{\mathbf{C} \in \Phi_{l_i=u} \rightarrow \forall \mathbf{C}' \in \Phi_{l_i=v} \middle| \forall \mathbf{C} \in \Phi_{l_i=u}\right\}}(\kappa)$, we have the following cases.

1. $v > u$ and $v < L$. Then, the probability $p_{\left\{\mathbf{C} \in \Phi_{l_i=u} \rightarrow \forall \mathbf{C}' \in \Phi_{l_i=v} \middle| \forall \mathbf{C} \in \Phi_{l_i=u}\right\}}(\kappa)$ is

derived as (5.18).

$$\begin{aligned}
&p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)\\
&=\frac{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\}}(\kappa)}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}\\
&=\frac{1}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}\\
&\times\sum_{j\neq i}\sum_{d=0}^{L}\left\{p(\Xi_j(\kappa)|l_j=d)p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=u,l_j=d}\right\}}(\kappa)p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u},\Xi_j(\kappa)\right\}}(\kappa)\right\}\\
&=\frac{1}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}\sum_{j\neq i}\sum_{d=0}^{L}\{p(\Xi_j(\kappa)|l_j=d)p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)p_{\{\forall\mathbf{C}\in\Phi_{l_j=d}\}}(\kappa)\\
&\times p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u},\Xi_j(\kappa)\right\}}(\kappa)\}\\
&=\frac{1}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}\sum_{j\neq i}\left\{p(\Xi_j(\kappa))p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)p(v-u\leq l_i^h(\kappa)<v-u+1)\right\}\\
&=(I-1)p_c(1-p_c)^{I-1}p(R_j(\kappa)\geq 2\lambda^*)p\left(\frac{2\sigma(v-u)}{\beta P_s\tau_{tx}}\leq|h_{ji}(\kappa)|^2<\frac{2\sigma(v+1-u)}{\beta P_s\tau_{tx}}\right).
\end{aligned}\tag{5.18}$$

2. $v > u$ and $v = L$. Then, the probability $p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)$ is derived as (5.19).

$$\begin{aligned}
&p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)\\
&=\frac{1}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}\sum_{j\neq i}\left\{p(\Xi_j(\kappa))p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)p(l_i^h(\kappa)\geq L-u)\right\}\\
&=(I-1)p_c(1-p_c)^{I-1}p(R_j(\kappa)\geq 2\lambda^*)p\left(|h_{ji}(\kappa)|^2\geq\frac{2\sigma(L-u)}{\beta P_s\tau_{tx}}\right).
\end{aligned}\tag{5.19}$$

3. $v = u$. Then, the probability $p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)$ is derived as (5.20).

$$\begin{aligned}
&p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)\\
&=\frac{1}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}\sum_{j\neq i}\left\{p(\Xi_j(\kappa))p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)p(l_i^h(\kappa)<1)\right\}\\
&=(I-1)p_c(1-p_c)^{I-1}p(R_j(\kappa)\geq 2\lambda^*)p\left(|h_{ji}(\kappa)|^2<\frac{2\sigma}{\beta P_s\tau_{tx}}\right).
\end{aligned}\tag{5.20}$$

4. $v < u$ and $v > 0$. Then, the probability $p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)$ is derived as (5.21).

$$\begin{aligned}
&p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)\\
&=\frac{1}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}p(\Xi_i(\kappa))p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\rightarrow\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u},\Xi_i(\kappa)\right\}}(\kappa)\\
&=p_c(1-p_c)^{I-1}p\left(R_i(\kappa)\geq 2\lambda^*\right)p\left(\frac{2\sigma(u-v-1)}{\tau_{tx}P_s}<\frac{|h_{ii}(\kappa)|^2}{|g_{ii}(\kappa)|^2}\leq\frac{2\sigma(u-v)}{\tau_{tx}P_s}\,|R_i(\kappa)\geq 2\lambda^*\right).
\end{aligned}\tag{5.21}$$

5. $v < u$ and $v = 0$. Then, the probability $p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\to\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)$ is derived as (5.22).

$$\begin{aligned}&p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\to\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)\\&=\frac{1}{p_{\{\forall\mathbf{C}\in\Phi_{l_i=u}\}}(\kappa)}p(\Xi_i(\kappa))p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(\kappa)p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\to\forall\mathbf{C}'\in\Phi_{l_i=0}\middle|\forall\mathbf{C}\in\Phi_{l_i=u},\Xi_i(\kappa)\right\}}(\kappa)\\&=p_c(1-p_c)^{I-1}p\left(R_i(\kappa)\geq 2\lambda^*\right)p\left(\frac{|h_{ii}(\kappa)|^2}{|g_{ii}(\kappa)|^2}>\frac{2\sigma(u-1)}{\tau_{tx}P_s}\middle|R_i(\kappa)\geq 2\lambda^*\right).\end{aligned} \tag{5.22}$$

After that, we analyze the iteration $\Pi_i(\kappa+1)=\Pi_i(\kappa)\mathbf{P}_i(\kappa)$ in Markov chain M_1. According to the definition of $p_{l_i=v}(\kappa+1)$, we have

$$p_{l_i=v}(\kappa+1)=\sum_{u=0}^{L}p_{l_i=u}(\kappa)p_{u\to v}(\kappa) \tag{5.23}$$

where $p_{u\to v}(\kappa)$ can be obtained by (5.8)-(5.12)

Then, for Markov chain M_2, the initial state of the iteration $\Gamma_{\mathbf{C}}(\kappa+1)=\Gamma_{\mathbf{C}}(\kappa)\mathbf{P}_{\mathbf{C}}$ is set as $\Gamma_{\mathbf{C}}(0)$, which satisfies the condition $p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=d}\right\}}(0)=p_{\left\{\forall\mathbf{C}\in\Phi_{l_j=d}\right\}}(0)$ for $d\in\{0,1,2,\cdots,L\}$ and $\{i,j\}\in\{1,2,\cdots,I\}$. Accordingly, for any relay i, the probability $p_{l_i=d}(0)$ for $d\in\{0,1,2,\cdots,L\}$ can be derived as

$$\begin{aligned}p_{l_i=d}(0)&=p_{\left\{\forall\mathbf{C}\in\Phi_{l_i=d}\right\}}(0)\\&=\sum_{\mathbf{C}\in\Phi_{l_i=d}}p_{\mathbf{C}}(0)\end{aligned} \tag{5.24}$$

where $\Phi_{l_i=d}=\{\mathbf{C}\in\Phi:l_i=d\}$ and $p_{\mathbf{C}}(0)$ is the element of $\Gamma_{\mathbf{C}}(0)$. Then, given the initial state $\Gamma_{\mathbf{C}}(0)$, the next state, denoted $\Gamma_{\mathbf{C}}(1)$, can be derived by the iteration $\Gamma_{\mathbf{C}}(\kappa+1)=\Gamma_{\mathbf{C}}(\kappa)\mathbf{P}_{\mathbf{C}}$. In addition, $p_{l_i=d}(1)$ can also be derived by (5.17).

For Markov chain M_1, the initial state $\Pi_i(0)$ is set as

$$\Pi_i(0)=\{p_{l_i=0}(0),p_{l_i=1}(0),p_{l_i=2}(0),\cdots,p_{l_i=L}(0)\} \tag{5.25}$$

where $p_{l_i=d}(0)$ for $d\in\{0,1,2,\cdots,L\}$ is obtained by (5.24). Then, given the initial state $\Pi_i(0)$, $p_{l_i=d}(1)$ can be derived by (5.23) after the iteration $\Pi_i(\kappa+1)=\Pi_i(\kappa)\mathbf{P}_i(\kappa)$.

Since the value of $p_{l_i=d}(0)$ for $i\in\{1,2,\cdots,I\}$ and $d\in\{0,1,2,\cdots,L\}$ in Markov chain M_1 and Markov chain M_2 is the same, the probability $p_{u\to v}(0)$ is equivalent to the probability $p_{\left\{\mathbf{C}\in\Phi_{l_i=u}\to\forall\mathbf{C}'\in\Phi_{l_i=v}\middle|\forall\mathbf{C}\in\Phi_{l_i=u}\right\}}(0)$ for $u\in\{0,1,2,\cdots,L\}$ and $v\in\{0,1,2,\cdots,L\}$ according to (5.8)-(5.12) and (5.18)-(5.22). In addition, the probability $p_{l_i=u}(0)$ is equivalent to the probability $\sum_{\mathbf{C}\in\Phi_{l_i=u}}p_{\mathbf{C}}(0)$ according to the definition of $p_{l_i=u}(0)$. Accordingly, $p_{l_i=d}(1)$ for $d \in \{0, 1, 2, \cdots, L\}$ which is obtained by (5.17) is same with that obtained by (5.23). In other words, the value of $p_{l_i=d}(1)$ in Markov chain M_1 and Markov chain M_2 is the same. Note that $\Pi_i(1) = \{p_{l_i=0}(1), p_{l_i=1}(1), p_{l_i=2}(1), \cdots, p_{l_i=L}(1)\}$. Thus, the next state of state $\Pi_i(0)$ in Markov chain M_1 by the iteration $\Pi_i(\kappa+1) = \Pi_i(\kappa)\mathbf{P}_i(\kappa)$ can also be derived by the iteration $\Gamma_{\mathbf{C}}(\kappa+1) = \Gamma_{\mathbf{C}}(\kappa)\mathbf{P}_{\mathbf{C}}$ in Markov chain M_2.

Similarly, we can then conclude that the value of $p_{l_i=d}(2)$ in Markov chain M_1 and Markov chain M_2 is the same. Accordingly, after $\kappa \in \{0, 1, 2, \cdots\}$ iterations, the value of $p_{l_i=d}(\kappa)$ in Markov chain M_1 and Markov chain M_2 would still keep the same. Since the iteration $\Gamma_{\mathbf{C}}(\kappa+1) = \Gamma_{\mathbf{C}}(\kappa)\mathbf{P}_{\mathbf{C}}$ for $\kappa \to \infty$ converges to the unique stationary state distribution $\Gamma_{\mathbf{C}}$, $p_{l_i=d}(\kappa)$ which is derived by (5.17) in Markov chain M_2 also converges to the unique stationary probability $p_{l_i=d}$ when $\kappa \to \infty$. Therefore, for iteration $\Pi_i(\kappa+1) = \Pi_i(\kappa)\mathbf{P}_i(\kappa)$ in Markov chain M_1, $p_{l_i=d}(\kappa)$ which is derived by (5.23) also converges to the unique stationary probability $p_{l_i=d}$ when $\kappa \to \infty$. Due to $\Pi_i(\kappa) = \{p_{l_i=0}(\kappa), p_{l_i=1}(\kappa), p_{l_i=2}(\kappa), \cdots, p_{l_i=L}(\kappa)\}$, $\Pi_i(\kappa)$ in Algorithm 5.1 would converge to the unique stationary state distribution Π_i.

This completes the proof. $\square$

According to Lemma 9, Lemma 10, and Lemma 11, we propose an iterative algorithm, named Algorithm 5.2, to obtain the optimal stopping rule N^* for Problem P1.

In summary, the optimal stopping strategy of the proposed DOS scheme is stated as follows. If source i is the winner source after the channel contention, relay i calculates the achievable transmission rate R_i according to its residual energy and the channel condition. If $R_i \geq 2\lambda^*$, source i transmits its data to relay i with rate R_i, and then, relay i forwards the received data to destination i. Otherwise, source i gives up the transmission opportunity and then all I sources start a new channel contention. Note that the maximal average network throughput λ^* is calculated off-line by Algorithm 5.2. Therefore, relay i can make a fast decision. In other words, the proposed scheme is easy to implement.

Algorithm 5.2 The optimal stopping rule N^* of Problem P1

Initialize λ_0, ς_1, and $z = 0$,
loop:
Compute Π_i by Algorithm 1,
Compute $E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\mathrm{tx}}}{2}\right]$,
Compute $E\left[\sum_{n=1}^{N^*_{\lambda_z}} \tau_0^n + \tau_{\mathrm{tx}}\right]$,
Compute $\lambda_{z+1} = \frac{E\left[\frac{R_{i,N^*_{\lambda_z}}\tau_{\mathrm{tx}}}{2}\right]}{E\left[\sum_{n=1}^{N^*_{\lambda_z}} \tau_0^n + \tau_{\mathrm{tx}}\right]}$,
Set $z = z + 1$,
if $|\lambda_z - \lambda_{z-1}| > \varsigma_1$ **then**
    goto *loop*
end if
Set $\lambda^* = \lambda_z$,
The optimal stopping rule $N^* = \min\{N \geq 1 : R_{i,N} \geq 2\lambda^*\}$.

5.4 Performance Evaluation

In this section, simulation results are provided to evaluate the performance of the proposed DOS scheme. The source-to-relay link and relay-to-destination link experience i.i.d. Rayleigh fading. The number of source-destination pairs is set as 5. The channel coherence duration and the duration of an RTS/CTS packet's transmission are $\tau_{\mathrm{tx}} = 3ms$ and $\tau_{\mathrm{c}} = 30\mu s$, respectively. The channel contention probability of a source is set as 0.2. The capacity of a source's battery and the amount of energy for each level are $B_{\max} = 10^3\sigma$ and $\sigma = 10^{-3}J$, respectively. β is set as 0.7. If a source i starts a transmission, the average received signal-to-noise ratio (SNR) at relay $j(j \neq i)$ is set as $6dB$.

Firstly, we validate the optimal throughput λ^* of the proposed scheme exists. In addition, we also need to verify that λ^* of the proposed DOS scheme can be obtained by Algorithm 5.2. If source i starts a transmission, the average received SNR at relay i is set as $15dB$. We also set $\nu_{\mathrm{h}}^2 = \nu_{\mathrm{g}}^2$. In the simulation, different value of λ are tested for the proposed scheme, which means $N = \min\{N \geq 1 : R_{i,N} \geq 2\lambda\}$. The simulation results are given in Fig. 5.2. We observe that the average throughput is increasing, and then decreasing with the growth of λ. The optimal point is achieved at the point where the average throughput is approximately equal to the value of λ. Thus, the optimal value of λ is the maximal average throughput of the network. For the proposed DOS scheme, λ^* which is calculated by Algorithm 5.2 is 2.07. According to Fig. 5.2, the value of λ at the optimal point is almost the same with λ^* for the proposed DOS scheme. Thus, it validates that λ^* can be calculated by Algorithm 5.2.

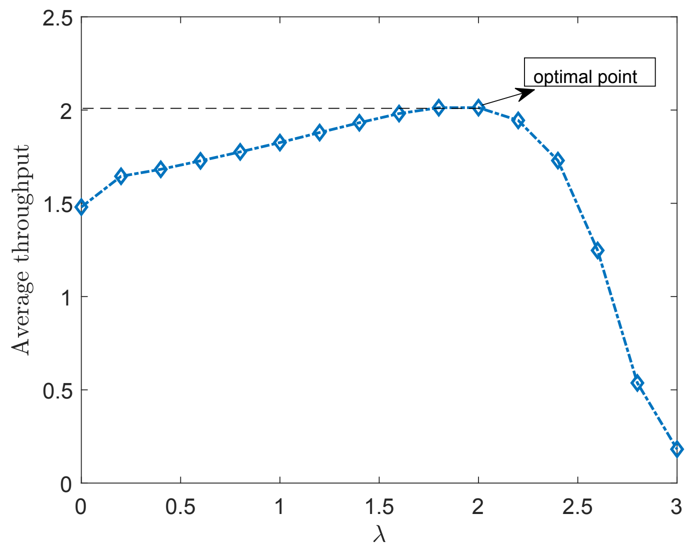

Figure 5.2: value of λ v.s. the average throughput.

To evaluate our proposed DOS scheme, we perform a comparison with two benchmark DOS schemes.

- No wait scheme: In this scheme, the winner source always transmit its data with the achievable transmission rate, and then, the relay forwards the received data without waiting.

- Two-level scheme[5]: In this scheme, let source i denote the winner source. Then, relay i probes the CSI of the source-to-relay link by the RTS/CTS scheme. Then, relay i decides to stop (i.e., start a transmission) or continue (i.e., give up the transmission opportunity). If the stop action is taken, source i starts a transmission with rate $R_{i,N}(t) = \log_2 (1 + P_{\mathrm{s}}|h_{ii,N}(t)|^2)$ over the next duration τ_{tx}. Once receiving the data, relay i probes the CSI of the relay-to-destination link by the RTS/CTS scheme. If the achievable transmission rate of this link is not less than $R_{i,N}(t)$, relay i forwards its received data to destination i. Other-

[5] A similar scheme can be found in [31]

wise, relay i waits a duration τ_{tx} and repeats the procedure (i.e., probe the CSI of the relay-to-destination link). Similar to our scheme, the optimal strategy of this scheme can be derived as $N^* = \min\{N \geq 1 : f(R_{i,N}, l_i) \geq \lambda^* \tau_{\text{tx}}\}$, where $f(R_{i,N}, l_i) = R_{i,N}\tau_{\text{tx}} - \frac{\lambda^*}{exp\left\{\frac{-(2^{R_{i,N}}-1)\tau_{\text{tx}}}{l_i \sigma v_{\text{g}}^2}\right\}}(2\tau_{\text{c}} + \tau_{\text{tx}})$, λ^* is the unique solution of $E\left[(f(R_{i,N}, l_i) - \lambda\tau_{\text{tx}})^+\right] = \lambda\tau_1$, and $\tau_1 = \left(\frac{1}{Ip_{\text{c}}(1-p_{\text{c}})^{I-1}} + 1\right)\tau_{\text{c}}$.

In the simulation, the average SNR of the source-to-relay link varies form 8dB to 30dB. The optimal stopping strategy is adopted for the proposed DOS scheme and the two-level scheme. The simulation results are given in Fig. 5.3. It can be seen that our proposed scheme achieve a larger average throughput than the no wait scheme and the two-level scheme. If the average SNR of the source-to-link takes a small value, the two-level scheme can also achieve a good performance. The reason is that this scheme can fully utilize the source-to-relay link with good condition. However, the two-level scheme has a bad performance when the average SNR of the source-to-relay link is large. It is because that it wastes lots of time to wait the relay-to-destination link to be good enough.

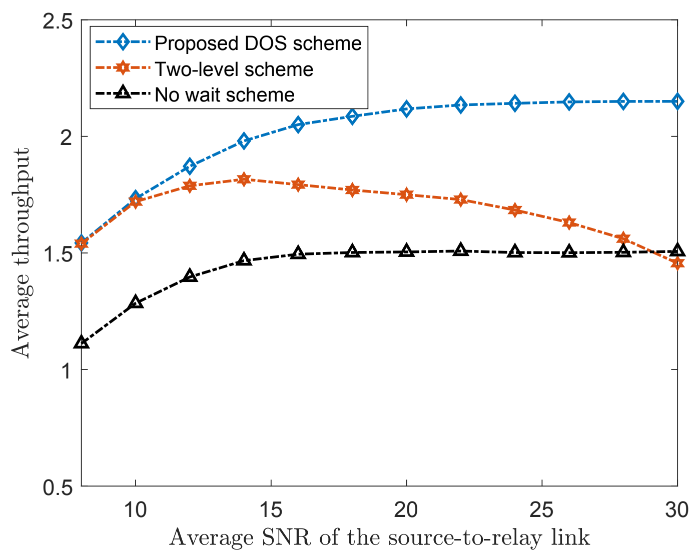

Figure 5.3: The average throughput of different DOS scheme.

5.5 Conclusion

In this chapter, we propose a DOS scheme in wireless networks with RF energy harvesting relay. To achieve the maximal network utility, the optimal stopping strategy is derived for the DOS scheme.

Chapter 6

Optimal Offloading in Fog Computing Systems with Non-orthogonal Multiple Access

Since the basic idea of non-orthogonal multiple access (NOMA) is to implement multiple access in the power domain, one of the most important methods to achieve the benefits of NOMA should be power allocation. The optimal power allocation in cellular networks with NOMA has been widely investigated. As a promising wireless technique to improve the spectrum efficiency, NOMA has also been shown important to the evolution of many types of applications or networks, e.g., vehicular *ad hoc* networks, digital TV broadcasting, terrestrial-satellite networks, fog computing, etc. [48]. Different from traditional cellular networks, the above mentioned networks with NOMA have some specific properties. Thus, NOMA power allocation schemes in traditional cellular networks cannot be directly adopted in other networks. For example, in fog computing, the computing capacity of fog nodes and the computing latency of tasks need to be considered when configuring NOMA power allocation.

Fog computing has recently become a promising method to meet the increasing computation demands from mobile applications in the Internet of Things (IoT). In fog computing, the computation tasks of an IoT device can be offloaded to fog nodes. Due to the limited computation capacity of a fog node, the IoT device may try to offload its tasks to multiple fog nodes. In this chapter, to improve the offloading efficiency, downlink non-orthogonal multiple access is applied in fog computing systems such that the IoT device can perform simultaneous offloading to multiple fog nodes. Then, to maximize the long-term average system utility, a task and power allocation problem for computation offloading is formulated, subject to task delay and energy cost constraints. By Lyapunov optimization method, the original problem is transformed to an online optimization problem in each time slot, which is non-convex. Accordingly, we propose an algorithm to solve the non-convex online optimization problem with polynomial complexity.

6.1 Introduction

In the past decade, the IoT and many new mobile applications have experienced fast growth. However, the limited capability of mobile devices cannot meet the computation demands of the applications [98]. Although traditional cloud computing may provide sufficient computation resources, the servers are normally geographically centralized. Hence, when dealing with massive computing demands, the network may be congested and the latency may be high. Users with some IoT applications, especially latency-sensitive applications, may suffer from poor quality of experience (QoE). A feasible solution to this problem is fog computing [99]. Serving as an intermediate layer between IoT devices and cloud servers, fog computing utilizes computing capabilities at the network edge, and thus, offers a new solution to meet the computing demands of lots of IoT applications [100, 101]. Computing devices of fog computing, referred to as *fog nodes*, can be traditional networking components (e.g., routers, base stations, switches, and so on) that are close to the IoT devices. Accordingly, the offloading latency can be largely reduced due to the low transmission latency. Fog computing can enhance cloud computing in many applications, e.g., latency-sensitive applications [102].

Fog computing has attracted a lot of efforts from both industry and academia. In [103], a platform that uses fog computing to improve the efficiency in industrial processes is introduced. In [104], a new type of vehicular networks, named vehicular fog computing (VFC), is proposed. The architecture and challenges of VFC are discussed. As another example, the face identification task in many applications usually needs a large amount of computation and communication capability. A new fog computing-aided face identification model is proposed in [105]. Fog nodes process the raw data of facial images, and send their processing results (feature values of the original facial images) to the cloud for further processing, thus largely reducing the traffic load to and the computation load of the cloud. The work in [106] introduces a four-layer distributed fog computing architecture in smart cities, including the top layer, the intermediate computing layer, the edge computing layer, as well as the sensing layer. The work in [107] reviews research efforts on security and resilience of fog computing.

Different from traditional cloud servers, resources in fog nodes may be very limited. Hence, allocation of computing data, i.e., computation offloading, needs to balance the cost of each fog node [108]. In [109], an effective computation offloading strategy is proposed with one IoT device and one fog node. The proportion of tasks distributed to the local computing and fog computing is derived. Moreover, the case with an energy-limited IoT device is also considered. The work in [110] considers a computation offloading scheme from a single IoT device to multiple fog nodes. The number of tasks distributed to each fog node is determined to minimize the energy cost. In [111], a computation offloading scheme is designed for multiple IoT devices and one fog node, which can achieve fairness among the IoT devices.

In fog computing, each fog node often needs to provide computing services to multiple IoT devices. Hence, the computation resources allocated for each IoT device are limited. Similarly, each IoT may have multiple fog nodes in its vicinity. To accelerate the computation of its tasks, the IoT device may try to offload its tasks to several fog nodes. For example, the feature extraction task in face identification, which is moved to fog nodes [105], still needs lots of computation resources. This task can be divided to several small tasks and executed in parallel. Thus, the IoT device tries to offload these small tasks to multiple fog nodes to accelerate the execution. As the computation offloading is usually performed by wireless channels, the orthogonal multiple access (OMA) technologies over wireless channels, e.g., time-division multiple access (TDMA), frequency-division multiple access (FDMA), and orthogonal frequency-division multiple access (OFDMA) [112], are commonly adopted to offload the tasks [109]- [111]. In other words, at a given moment, each fog node is assigned a unique wireless resource block (e.g., a time slot in TDMA or some sub-carriers in OFDMA) for computation offloading. However, the spectrum efficiency of OMA techniques may be low due to the fluctuation in channel conditions for different nodes, e.g., some resource blocks may be allocated to nodes with poor channel conditions. Therefore, the computing efficiency may be degraded as the tasks may not be delivered to fog nodes in a timely manner. In the wireless communication literature, to improve the spectrum efficiency, non-orthogonal multiple access (NOMA) is introduced [12]. In NOMA, multiple transmissions can share a single resource block simultaneously [113], and thus, the spectrum efficiency is largely improved compared to OMA [114]. NOMA has been considered as a promising radio access technology for the fifth-generation (5G) wireless communication systems [115]. The integration of NOMA with fog computing can further enhance the computation offloading performance as follows. By NOMA, an IoT device can use one single wireless resource block to offload data to multiple fog nodes. Thus, the computation offloading latency can be largely reduced, and the computing capacity of multiple fog nodes can be well exploited [116]. This feature can largely benefit latency-sensitive applications.

Although fog computing with NOMA brings profits, some design challenges are also introduced, e.g., computation offloading schemes with OMA [109]- [111] cannot be adopted to the NOMA case directly. In [117], uplink NOMA is applied to computation offloading, in which multiple mobile users offload tasks to a fog node simultaneously by uplink NOMA. To minimize the energy cost, a convex optimization problem is formulated and solved. However, this scheme cannot be adopted to the case when an IoT device offloads its tasks to multiple fog nodes. In addition, there are two issues for the work in [117]. 1) It is assumed that computation capacity of a fog node is unlimited, which may not be practical. 2) The proposed scheme in [117] cares only the profit of executing a single task, i.e., the short-term profit. However, for some applications, e.g., multi-media streaming, it is more appropriate to consider system performance over a long term [109].

To address the above issues, we propose a novel computation offloading scheme with downlink NOMA in this work. The major contributions of this work are summarized as follows.

1. A computation offloading scheme with downlink NOMA is proposed in a fog computing system with one IoT device[1] and several fog nodes. The computation

[1]Our work can be straightforwardly extended to scenarios with multiple IoT devices.

capability of each fog node which is allocated to the IoT device is limited.

2. In the proposed scheme, an optimization problem to maximize the long-term average system utility is formulated, subject to the task delay constraint and energy cost constraint. The Lyapunov optimization method is adopted to transform the formulated problem to an online optimization problem, which only involves instantaneous variables in each time slot.

3. The online optimization problem is still a non-convex optimization problem. Thus, we propose an algorithm with polynomial computation complexity to solve it.

4. Performance analysis is carried out for the proposed scheme by simulation, which shows that our proposed scheme obtains much more profits compared with traditional computation offloading schemes.

The rest of this chapter is organized as follows. Section 6.2 gives the system model and formulates the optimization problem. Section 6.3 transforms the formulated problem to an online optimization problem, and proposes a low-complexity algorithm to solve the online optimization problem. Section 6.4 shows simulation results. Finally, Section 6.5 concludes this chapter.

6.2 System Model and Problem Formulation

As shown in Fig. 6.1, we consider a fog computing system, where an IoT device running a computation-intensive application is assisted by N fog nodes. Computation offloading from the IoT device to the fog nodes is performed over the wireless channels. A slotted time structure is implemented in the system. The duration of each time slot is T.

At time slot $t \in \{0, 1, 2, \cdots\}$, the IoT device generates an amount, denoted $D_{\max}(t)$, of data that need to be computed, and it allocates a portion of the data, denoted $D(t)$, to be offloaded to fog nodes. The remaining data with amount $(D_{\max}(t) - D(t))$ can be computed at the local CPU of the IoT device or by a traditional cloud server. Let $R_i(t)$ denote the amount of data that are delivered to fog node $i \in \{1, 2, \cdots, N\}$ at the tth time slot. Then, the transmission rate between the

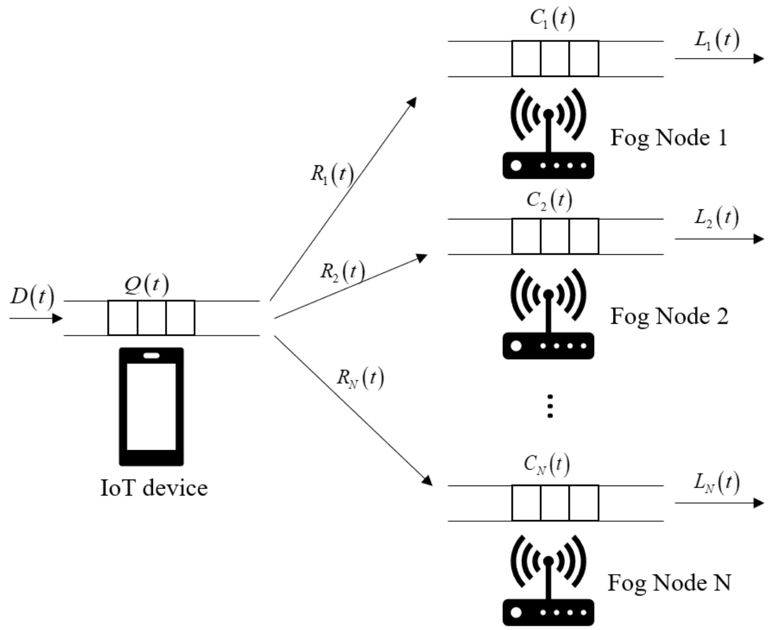

Figure 6.1: The fog computing system model.

IoT device and fog node i at the tth time slot is $r_i(t) = \frac{R_i(t)}{T}$. The IoT device has a task buffer to store the data to be offloaded to fog nodes. Let $Q(t)$ denote the total data amount in the IoT device's task buffer at the beginning of the tth time slot. Thus, we have

$$Q(t+1) = \left[Q(t) - \sum_{i=1}^{N} R_i(t) \right]^+ + D(t) \tag{6.1}$$

where $[x]^+$ means $\max\{x, 0\}$.

Each fog node has limited computing capacity. At fog node i, it maintains a task buffer to store the data from the IoT device. Let $C_i(t)$ denote the occupancy of fog node i's task buffer at the beginning of the tth time slot. Fog node i provides a service rate (CPU frequency) denoted as $f_i(t)$ to the IoT device at the tth time slot. Thus, the amount of data from the IoT device that can be computed by fog node i at the tth time slot is $L_i(t) = \frac{f_i(t)T}{\varphi}$, where φ is the number of CPU cycles that are needed to compute a unit size of the data. Thus, we have

$$C_i(t+1) = [C_i(t) - L_i(t)]^+ + R_i(t). \tag{6.2}$$

In this study, NOMA is adopted to transmit data from the IoT device to fog nodes. The wireless channels are assumed as independent and identically distributed (i.i.d.) block Rayleigh fading. In other words, the channel gain between the IoT device and fog node i, denoted as $g_i(t)$, keeps unchanged within one time slot, but varies independently from a time slot to the next one. $g_i(t)$ can be expressed as $g_i(t) = \frac{g_0}{{d_i}^{\alpha}}$, where g_0 follows an exponential distribution with unit mean, d_i is the distance between the IoT device and fog node i, and α is the path-loss exponent. Without loss of generality, for the target slot (the tth time slot), we assume $g_1(t) \geq g_2(t) \geq \cdots \geq g_N(t)$ [114]. For the IoT device's NOMA transmissions to the fog nodes at the tth time slot, let $p_i(t)$ denote the power allocated to the data offloading to fog node i. The transmission power consumption of the IoT device in each time slot should be no more than a threshold value denoted as $p_{\max}$. Thus, we have the constraint $\sum\limits_{i=1}^{N} p_i(t) \leq p_{\max}$. Moreover, the long-term average transmission power consumption of the IoT device should not be more than a threshold value denoted as $\bar{p}$. Thus, we also have the constraint $\lim\limits_{\tau \to \infty} \sum\limits_{t=0}^{\tau-1} \frac{E\left[\sum\limits_{i=1}^{N} p_i(t)\right]}{\tau} \leq \bar{p}$, where $E[\cdot]$ means expectation[2]. Based on the principle of NOMA [114], the transmission rate between the IoT device and fog node i at the tth time slot is given as

$$r_i(t) = W \log_2 \left(1 + \frac{p_i(t)\, g_i(t)}{\sum\limits_{j=1}^{i-1} p_j(t)\, g_i(t) + \upsilon} \right), \tag{6.3}$$

where W is the channel bandwidth and υ is the background noise power.

In fog computing system, the IoT device would try to offload as much data as possible to fog nodes under power constraints of the IoT device and computation capacity constraints of the fog nodes. Accordingly, we define the system utility as the amount of data that are computed by fog nodes. At the tth time slot, the actual amount of data that are computed by fog node i is $\min\{L_i(t), C_i(t)\}$. To guarantee the QoE of the IoT device, the input data to the IoT device's task buffer should be executed under finite execution delay, and thus, the amount of computed data by fog nodes is equivalent to the amount of input data to the IoT device's task buffer in a long term. Thus, the long-term system utility can be expressed as $U \triangleq \log(1 + \lim\limits_{\tau \to \infty} \sum\limits_{t=0}^{\tau-1} \frac{E[D(t)]}{\tau})$.[3] In this study, we maximize the long-term system utility

[2] In our problem formulation, we optimize $D(t), p_1(t), p_2(t), ..., p_N(t)$. The expectation is over other variables such as $Q(t)$ and $C_1(t), ..., C_N(t)$.

[3] In addition to the logarithmic function, the long-term system utility function can be other non-decreasing functions.

by optimizing the input data size $D(t)$ to the IoT device's task buffer and the power allocation vector for data transmissions $\mathbf{p}(t) \triangleq \{p_1(t), p_2(t), \cdots, p_N(t)\}$. Thus, the optimization problem (named P1) is stated as follows:

$$
\begin{aligned}
\text{P1}: \max_{D(t),\mathbf{p}(t)} \quad & U = \log\left(1 + \lim_{\tau\to\infty}\sum_{t=0}^{\tau-1}\frac{E\left[D(t)\right]}{\tau}\right) && \text{(6.4a)}\\
\text{s.t.} \quad & p_i(t) \geq 0,\ \forall i \in \{1, 2, \cdots, N\}, \\
& \qquad \forall t \in \{0, 1, 2, \cdots\} && \text{(6.4b)}\\
& \sum_{i=1}^{N} p_i(t) \leq p_{\max},\ \forall t \in \{0, 1, 2, \cdots\} && \text{(6.4c)}\\
& \lim_{\tau\to\infty}\sum_{t=0}^{\tau-1}\frac{E\left[\sum\limits_{i=1}^{N} p_i(t)\right]}{\tau} \leq \bar{p} && \text{(6.4d)}\\
& \lim_{t\to\infty}\frac{E\left[Q(t)\right]}{t} = 0 && \text{(6.4e)}\\
& \lim_{t\to\infty}\frac{E\left[C_i(t)\right]}{t} = 0,\ \forall i \in \{1, 2, \cdots, N\} && \text{(6.4f)}\\
& 0 \leq D(t) \leq D_{\max}(t),\ \forall t \in \{0, 1, 2, \cdots\} && \text{(6.4g)}
\end{aligned}
$$

where (6.4b), (6.4c), and (6.4d) are power allocation constraints, (6.4e) and (6.4f), which let the task buffers remain mean rate stable [57], are to guarantee that the data can be computed with finite execution delay, and (6.4g) is the input data size constraint to the IoT device's task buffer.

6.3 Problem Transformation and Proposed Algorithm

In this section, we will solve Problem P1.

In Problem P1, constraint (6.4d) is hard to handle, and thus, it is challenging to solve Problem P1. To address this challenging issue, we use the method of virtual queue [57] to transform constraint (6.4d) to an equivalent one, as follows.

Lemma 12. *Constraint (6.4d) in Problem P1 can be replaced by the following constraint*

$$\lim_{t\to\infty}\frac{E\left[B(t)\right]}{t} = 0, \tag{6.5}$$

where $B(t)$ is a virtual queue [57] defined as

$$B(t+1) \triangleq \left[B(t) + \sum_{i=1}^{N} p_i(t) - \bar{p} \right]^{+}, \tag{6.6}$$

with $B(0) = 0$.

Proof. According to the definition of $B(t)$, we have $B(t+1) = B(t) - b(t)$ for $t \in \{0, 1, 2, \cdots\}$, where $b(t) \triangleq \min\{\bar{p} - \sum_{i=1}^{N} p_i(t), B(t)\}$. Then, we have

$$\begin{aligned}
&B(1) - B(0) = -b(0), \\
&B(2) - B(1) = -b(1), \\
&\ldots \\
&B(\tau) - B(\tau - 1) = -b(\tau - 1).
\end{aligned} \tag{6.7}$$

Then, taking summation of the equations in (6.7), we have $B(\tau) - B(0) = -\sum_{t=0}^{\tau-1} b(t) \geq -\sum_{t=0}^{\tau-1} [\bar{p} - \sum_{i=1}^{N} p_i(t)]$. Accordingly, we have $\frac{B(\tau)}{\tau} - \frac{B(0)}{\tau} \geq -\frac{1}{\tau} \sum_{t=0}^{\tau-1} [\bar{p} - \sum_{i=1}^{N} p_i(t)]$, and by taking expectation on both sides of the inequality and taking $\tau \to \infty$, we have

$$\begin{aligned}
\lim_{\tau \to \infty} \frac{E[B(\tau)]}{\tau} &\geq \lim_{\tau \to \infty} \left\{ -\frac{1}{\tau} \sum_{t=0}^{\tau-1} \left[\bar{p} - E\left[\sum_{i=1}^{N} p_i(t) \right] \right] \right\} \\
&= -\bar{p} + \lim_{\tau \to \infty} \sum_{t=0}^{\tau-1} \frac{E\left[\sum_{i=1}^{N} p_i(t) \right]}{\tau}.
\end{aligned} \tag{6.8}$$

Thus, if $\lim_{\tau \to \infty} \frac{E[B(\tau)]}{\tau} = 0$, we have $\lim_{\tau \to \infty} \sum_{t=0}^{\tau-1} \frac{E\left[\sum_{i=1}^{N} p_i(t) \right]}{\tau} \leq \bar{p}$. It means that constraint (6.4d) in Problem P1 can be replaced by constraint (6.5). This completes the proof. $\square$

According to Lemma 12, the long-term power consumption of the IoT device can be estimated by $B(t)$.

As Problem P1 considers the long-term utility, it can be modeled as a Markov Decision Process (MDP) problem. Then, some general algorithms for MDP problems, e.g., value iteration algorithm, can be adopted. However, to find the optimal policy, it needs to simulate for a long time. Moreover, if we model Problem P1 as an MDP problem, the number of states is huge (for example, the occupancy of the IoT device's task buffer is continuous, which may have to be quantized to a large number of stages in MDP). Thus, we do not model Problem P1 using MDP.

In order to find a low-complexity algorithm to solve Problem P1, we use Lyapunov method [57]- [118] to transform Problem P1 to an online optimization problem that only involves instantaneous variables of each time slot. Firstly, the Lyapunov function is defined as

$$Y(t) \triangleq \frac{1}{2}\left[B^2(t) + Q^2(t) + \sum_{i=1}^{N} C_i^2(t)\right]. \tag{6.9}$$

Then, the conditional Lyapunov drift is given as

$$\Delta(t) = E\left[Y(t+1) - Y(t)\,|\mathcal{S}(t)\right], \tag{6.10}$$

where $\mathcal{S}(t) \triangleq \{B(t), Q(t), C_1(t), C_2(t), \cdots, C_N(t)\}$. Thus, the Lyapunov drift-plus-penalty function is expressed as

$$\Delta_V(t) = \Delta(t) - VE\left[\log(1 + D(t))\,|\mathcal{S}(t)\right], \tag{6.11}$$

where $V \geq 0$ is a parameter that can be used to achieve a balance between queue stability and system utility. An upper bound of $\Delta_V(t)$ is given by the following lemma.

Lemma 13. *For any $D(t)$ and $\mathbf{p}(t)$, $\Delta_V(t)$ is upper bounded by*

$$\begin{aligned}\Delta_V(t) \; &\leq M + E\left[B(t)\left(\sum_{i=1}^{N} p_i(t) - \bar{p}\right)\Big|\mathcal{S}(t)\right] + E\left[Q(t)\left(D(t) - \sum_{i=1}^{N} R_i(t)\right)\Big|\mathcal{S}(t)\right] \\ &\quad + E\left[\sum_{i=1}^{N}\Big(C_i(t)\left(R_i(t) - L_i(t)\right)\Big)\Big|\mathcal{S}(t)\right] - VE\left[\log(1 + D(t))\Big|\mathcal{S}(t)\right],\end{aligned} \tag{6.12}$$

where M is a constant.

Proof. We use the method in [57] to prove this lemma.

For any $x \geq 0$, $y \geq 0$, and $z \geq 0$, we have

$$\begin{aligned}&([x-y]^+ + z)^2 \\ &= ([x-y]^+)^2 + z^2 + 2[x-y]^+ z \\ &\overset{\text{(i)}}{\leq} (x-y)^2 + z^2 + 2xz \\ &= x^2 + y^2 + z^2 + 2x(z-y),\end{aligned}$$

where step (i) comes from $([x-y]^+)^2 \leq (x-y)^2$ and $[x-y]^+ \leq x$. Accordingly, we have

$$\begin{aligned}&Q^2(t+1) - Q^2(t) \\ &= \left(\left[Q(t) - \sum_{i=1}^{N} R_i(t)\right]^+ + D(t)\right)^2 - Q^2(t) \\ &\leq \left(\sum_{i=1}^{N} R_i(t)\right)^2 + D^2(t) + 2Q(t)\left(D(t) - \sum_{i=1}^{N} R_i(t)\right),\end{aligned} \tag{6.13}$$

$$
\begin{aligned}
&C_i^2(t+1) - C_i^2(t) \\
&= \left([C_i(t) - L_i(t)]^+ + R_i(t)\right)^2 - C_i^2(t) \\
&\le L_i^2(t) + R_i^2(t) + 2C_i(t)\left(R_i(t) - L_i(t)\right).
\end{aligned}
\tag{6.14}
$$

We also have

$$
\begin{aligned}
&B^2(t+1) - B^2(t) \\
&= \left(\left[B(t) + \sum_{i=1}^{N} p_i(t) - \bar{p}\right]^+\right)^2 - B^2(t) \\
&\le \left(B(t) + \sum_{i=1}^{N} p_i(t) - \bar{p}\right)^2 - B^2(t) \\
&= \left(\bar{p} - \sum_{i=1}^{N} p_i(t)\right)^2 + 2B(t)\left(\sum_{i=1}^{N} p_i(t) - \bar{p}\right).
\end{aligned}
\tag{6.15}
$$

Based on these, we have

$$
\begin{aligned}
&\Delta_V(t) \\
&= \tfrac{1}{2}E\left[B^2(t+1) + Q^2(t+1) + \sum_{i=1}^{N} C_i^2(t+1) - B^2(t) - Q^2(t) - \sum_{i=1}^{N} C_i^2(t) \,\middle|\, \mathcal{S}(t)\right] \\
&\qquad -VE\left[\log(1+D(t)) \,\middle|\, \mathcal{S}(t)\right] \\
&\le \tfrac{1}{2}E\left[\left(\bar{p} - \sum_{i=1}^{N} p_i(t)\right)^2 \,\middle|\, \mathcal{S}(t)\right] + \tfrac{1}{2}E\left[\left(\sum_{i=1}^{N} R_i(t)\right)^2 + D^2(t) \,\middle|\, \mathcal{S}(t)\right] \\
&\qquad + \tfrac{1}{2}E\left[\sum_{i=1}^{N}\left(L_i^2(t) + R_i^2(t)\right) \,\middle|\, \mathcal{S}(t)\right] + E\left[B(t)\left(\sum_{i=1}^{N} p_i(t) - \bar{p}\right) \,\middle|\, \mathcal{S}(t)\right] \\
&\qquad + E\left[Q(t)\left(D(t) - \sum_{i=1}^{N} R_i(t)\right) \,\middle|\, \mathcal{S}(t)\right] + E\left[\sum_{i=1}^{N}\Big(C_i(t)\left(R_i(t) - L_i(t)\right)\Big) \,\middle|\, \mathcal{S}(t)\right] \\
&\qquad -VE\left[\log(1+D(t)) \,\middle|\, \mathcal{S}(t)\right] \\
&\le M + E\left[B(t)\left(\sum_{i=1}^{N} p_i(t) - \bar{p}\right) \,\middle|\, \mathcal{S}(t)\right] + E\left[Q(t)\left(D(t) - \sum_{i=1}^{N} R_i(t)\right) \,\middle|\, \mathcal{S}(t)\right] \\
&\qquad + E\left[\sum_{i=1}^{N}\Big(C_i(t)\left(R_i(t) - L_i(t)\right)\Big) \,\middle|\, \mathcal{S}(t)\right] - VE\left[\log(1+D(t)) \,\middle|\, \mathcal{S}(t)\right].
\end{aligned}
\tag{6.16}
$$

where M is the maximal possible value of the expression $\frac{1}{2}E\left[\left(\bar{p} - \sum_{i=1}^{N} p_i(t)\right)^2 \middle| \mathcal{S}(t)\right] + \frac{1}{2}E\left[\left(\sum_{i=1}^{N} R_i(t)\right)^2 + D^2(t) \middle| \mathcal{S}(t)\right] + \frac{1}{2}E\left[\sum_{i=1}^{N}\left(L_i^2(t) + R_i^2(t)\right) \middle| \mathcal{S}(t)\right]$. Since $\sum_{i=1}^{N} p_i(t)$, $\sum_{i=1}^{N} R_i(t)$, $D(t)$, and $\sum_{i=1}^{N} L_i(t)$ have fixed minimum values and fixed maximal values, M is a constant. This completes the proof. □

Then, in order to maintain the amount of data in the task buffers (of the IoT device and fog nodes) at a low level and maximize the system utility, we use the

Lyapunov optimization method to transform Problem P1 to Problem P2 given as (6.17), which minimizes the upper bound of the drift-plus-penalty function.

$$\begin{aligned}
\text{P2}: \min_{D(t),\mathbf{p}(t)} \quad & B(t)\left(\sum_{i=1}^{N} p_i(t) - \bar{p}\right) + Q(t)\left(D(t) - \sum_{i=1}^{N} R_i(t)\right) \\
& + \sum_{i=1}^{N}\Big(C_i(t)\left(R_i(t) - L_i(t)\right)\Big) - V\log\left(1 + D(t)\right) && (6.17a) \\
\text{s.t.} \quad & \text{constraints (6.4b), (6.4c), and (6.4g).} && (6.17b)
\end{aligned}$$

For Problem P2, it can be divided to the following two sub-problems, P2.1 and P2.2, which do not have coupled constraints. Note that the terms $B(t)\bar{p}$ and $C_i(t)L_i(t)$ are irrelevant to the variables $D(t)$ and $\mathbf{p}(t)$ at the tth time slot, and thus, these two terms are ignored in the sub-problems.

$$\begin{aligned}
\text{P2.1}: \min_{D(t)} \quad & G(D(t)) \triangleq Q(t)D(t) - V\log(1 + D(t)) && (6.18a) \\
\text{s.t.} \quad & 0 \le D(t) \le D_{\max}(t). && (6.18b)
\end{aligned}$$

$$\begin{aligned}
\text{P2.2}: \min_{\mathbf{p}(t)} \quad & O(\mathbf{p}(t)) \triangleq B(t)\sum_{i=1}^{N} p_i(t) - Q(t)\sum_{i=1}^{N} R_i(t) + \sum_{i=1}^{N} C_i(t)R_i(t) && (6.19a) \\
\text{s.t.} \quad & p_i(t) \ge 0,\ \forall i \in \{1, 2, \cdots, N\} && (6.19b) \\
& \sum_{i=1}^{N} p_i(t) \le p_{\max}. && (6.19c)
\end{aligned}$$

In the following, we focus on solving the two sub-problems.

6.3.1 Optimal Solution of Problem P2.1

For the optimal solution of Problem P2.1, the following lemma is given.

Lemma 14. *The optimal solution of Problem P2.1 is given as*

$$D^*(t) = \min\left\{\max\left\{\frac{V}{Q(t)} - 1, 0\right\}, D_{\max}(t)\right\}. \quad (6.20)$$

Proof. This proof is mainly based on the theory of convex optimization. Taking the first-order derivative of the objective function of Problem P2.1 with respect to $D(t)$, we have $\frac{dG(D(t))}{dD(t)} = Q(t) - \frac{V}{1+D(t)}$. Accordingly, we have $\frac{dG(D(t))}{dD(t)} = 0$ when

$D(t) = \frac{V}{Q(t)} - 1$. Moreover, the objective function of Problem P2.1 is convex because $\frac{d^2G(D(t))}{dD(t)^2} = \frac{V}{(1+D(t))^2} > 0$ for $D(t) \geq 0$. Thus, the objective function, $G(D(t))$, decreases monotonically when $0 \leq D(t) \leq \frac{V}{Q(t)} - 1$ and increases monotonically when $D(t) > \frac{V}{Q(t)} - 1$. Then, taking constraint (6.18b) into account, the optimal solution of Problem P2.1 is shown in (6.20). This completes the proof. $\square$

6.3.2 Optimal Solution of Problem P2.2

According to $R_i(t) = r_i(t) \cdot T$ and (6.3), the objective function of Problem P2.2 can be expressed as

$$O(\mathbf{p}(t)) = B(t) \sum_{i=1}^{N} p_i(t) - T \sum_{i=1}^{N} W \log_2 \left(1 + \frac{p_i(t) g_i(t)}{\sum\limits_{j=1}^{i-1} p_j(t) g_i(t) + \upsilon} \right) (Q(t) - C_i(t)). \tag{6.21}$$

It is easy to find that Problem P2.2 is a non-convex optimization problem. Thus, it is hard to solve this problem by some standard methods. Then, we introduce the following lemma for Problem P2.2.

Lemma 15. *If $\mathbf{p}^*(t) \triangleq \{p_1^*(t), p_2^*(t), \cdots, p_N^*(t)\}$ is the optimal solution of Problem P2.2, $\sum\limits_{j=1}^{i} p_j^*(t)$ ($\forall i \in \{1, 2, \cdots, N\}$) should take one of the following values*

$$\begin{cases} 0; \\ \frac{\left(X_{i_2}(t) g_{i_2}(t) - X_{i_1}(t) g_{i_1}(t)\right)\upsilon}{g_{i_1}(t) g_{i_2}(t) \left(X_{i_1}(t) - X_{i_2}(t)\right)}, & 1 \leq i_1 < i_2 \leq N; \\ \frac{X_{i_1}(t)}{B(t)} - \frac{\upsilon}{g_{i_1}(t)}, & 1 \leq i_1 \leq N; \\ p_{\max} \end{cases} \tag{6.22}$$

where $X_i(t) = \frac{TW(Q(t) - C_i(t))}{\log 2}$.

Proof. In Problem P2.2, the two constraints are linear constraints, and the constraints' gradients are linearly independent. According to [119, Prop. 3.3.1], the Karush-Kuhn-Tucker (KKT) condition is a necessary condition for optimal solution of Problem P2.2. Thus, $\mathbf{p}^*(t)$ satisfies the KKT condition.

The Lagrangian of Problem P2.2 is given as $\mathcal{L}(\{p_i(t)\}, \{\mu_i\}, \lambda) = B(t) \sum\limits_{i=1}^{N} p_i(t) - T \sum\limits_{i=1}^{N} W \log_2 \left(1 + \frac{p_i(t) g_i(t)}{\sum\limits_{j=1}^{i-1} p_j(t) g_i(t) + \upsilon} \right) (Q(t) - C_i(t)) + \sum\limits_{i=1}^{N} \mu_i p_i(t) - \lambda \left(\sum\limits_{i=1}^{N} p_i(t) - p_{\max} \right)$, where λ, $\mu_1, \mu_2, ..., \mu_N$ are the Lagrange multipliers. Then, the KKT condition can be listed as follows.

$$\frac{d\mathcal{L}\left(\{p_i\left(t\right)\},\{\mu_i\},\lambda\right)}{dp_i\left(t\right)} = Z_i\left(t\right) + \mu_i - \lambda = 0, \forall i \in \{1,2,\cdots,N\}, \quad (6.23)$$

$$\mu_i p_i\left(t\right) = 0, \forall i \in \{1,2,\cdots,N\}, \quad (6.24)$$

$$\lambda\left(\sum_{i=1}^{N} p_i\left(t\right) - p_{\max}\right) = 0, \quad (6.25)$$

where

$$\begin{aligned} Z_i(t) \triangleq\ & B\left(t\right) - X_i(t)\frac{g_i(t)}{\sum\limits_{j=1}^{i} p_j(t)g_i(t)+\upsilon} \\ & + X_{i+1}(t)\left(\frac{g_{i+1}(t)}{\sum\limits_{j=1}^{i} p_j(t)g_{i+1}(t)+\upsilon} - \frac{g_{i+1}(t)}{\sum\limits_{j=1}^{i+1} p_j(t)g_{i+1}(t)+\upsilon}\right) \\ & + \cdots \\ & + X_N(t)\left(\frac{g_N(t)}{\sum\limits_{j=1}^{N-1} p_j(t)g_N(t)+\upsilon} - \frac{g_N(t)}{\sum\limits_{j=1}^{N} p_j(t)g_N(t)+\upsilon}\right). \end{aligned}$$

Then, we have the following two cases.

Case 1: All $p_i^*(t)$'s are zeros. Then, we have $\sum\limits_{j=1}^{i} p_j^*\left(t\right) = 0$.

Case 2: A number, denoted K, of $p_i^*(t)$'s are positive. Denote those positive power allocations as $p_{l_1}^*(t), p_{l_2}^*(t), ..., p_{l_K}^*(t)$ with $l_1 < l_2 < ... < l_K$. Denote $F(i) = \sum\limits_{j=1}^{i} p_j^*\left(t\right)$. Then we only need to prove that $F(i)$ $(i = l_1, l_2, ..., l_K)$ takes one value in (6.22).

From (6.24) we have $\mu_{l_1} = \mu_{l_2} = ... = \mu_{l_K} = 0$, and from (6.23) we have $Z_{l_1}(t) = ... = Z_{l_K}(t) = \lambda$. Thus, for $m = 1, 2, ..., K-1$, from $Z_{l_m}(t) = Z_{l_{m+1}}(t)$ we have

$$F(l_m) = \frac{\left(X_{l_{m+1}}\left(t\right)g_{l_{m+1}}\left(t\right) - X_{l_m}\left(t\right)g_{l_m}\left(t\right)\right)\upsilon}{g_{l_m}\left(t\right)g_{l_{m+1}}\left(t\right)\left(X_{l_m}\left(t\right) - X_{l_{m+1}}\left(t\right)\right)},$$

which is one value in (6.22).

Next we show $F(l_K)$ also takes one value in (6.22). If $\lambda \neq 0$, from (6.25) we have $F(l_K) = F(N) = p_{\max}$. If $\lambda = 0$, from (6.23) we have $Z_{l_K}\left(t\right) = B\left(t\right) - X_{l_K}\left(t\right)\frac{g_{l_K}(t)}{F(l_K)g_{l_K}(t)+\upsilon} = 0$. Accordingly, we have $F(l_K) = \frac{X_{l_K}(t)}{B(t)} - \frac{\upsilon}{g_{l_K}(t)}$, which is a value in (6.22).

This completes the proof. $\square$

6.3.2.1 Algorithm 6.1 to solve Problem P2.2

Based on Lemma 15, an algorithm, named Algorithm 6.1, can be developed to solve Problem P2.2, as follows.

The value of $p_i(t)$ for $\forall i \in \{1, 2, \cdots, N\}$ has two cases: $p_i(t) = 0$ or $p_i(t) > 0$. Therefore, there are 2^N cases for $\mathbf{p}(t)$. For each case, the values of $p_i(t)$'s can be derived by Lemma 15, as follows.

In the case, assume that the set with $p_i(t) > 0$ is $\{p_{l_1}(t), p_{l_2}(t), \cdots, p_{l_K}(t)\}$ with $l_1 < l_2 < ... < l_K$. Then from the proof of Lemma 15, we have

$$p_{l_1}(t) = \frac{\left(X_{l_2}(t)\, g_{l_2}(t) - X_{l_1}(t)\, g_{l_1}(t)\right) \upsilon}{g_{l_1}(t)\, g_{l_2}(t) \left(X_{l_1}(t) - X_{l_2}(t)\right)}, \tag{6.26}$$

and

$$p_{l_k}(t) = \frac{\left(X_{l_{k+1}}(t)\, g_{l_{k+1}}(t) - X_{l_k}(t)\, g_{l_k}(t)\right) \upsilon}{g_{l_k}(t)\, g_{l_{k+1}}(t) \left(X_{l_k}(t) - X_{l_{k+1}}(t)\right)} - \frac{\left(X_{l_k}(t)\, g_{l_k}(t) - X_{l_{k-1}}(t)\, g_{l_{k-1}}(t)\right) \upsilon}{g_{l_{k-1}}(t)\, g_{l_k}(t) \left(X_{l_{k-1}}(t) - X_{l_k}(t)\right)} \tag{6.27}$$

for $k = 2, 3, ..., K-1$.

We have two possible values for $p_{l_K}(t)$:

$$p_{l_K}(t) = p_{\max} - \frac{\left(X_{l_K}(t)\, g_{l_K}(t) - X_{l_{K-1}}(t)\, g_{l_{K-1}}(t)\right) \upsilon}{g_{l_{K-1}}(t)\, g_{l_K}(t) \left(X_{l_{K-1}}(t) - X_{l_K}(t)\right)} \tag{6.28}$$

and

$$p_{l_K}(t) = \frac{X_{l_K}(t)}{B(t)} - \frac{\upsilon}{g_{l_K}(t)} - \frac{\left(X_{l_K}(t)\, g_{l_K}(t) - X_{l_{K-1}}(t)\, g_{l_{K-1}}(t)\right) \upsilon}{g_{l_{K-1}}(t)\, g_{l_K}(t) \left(X_{l_{K-1}}(t) - X_{l_K}(t)\right)}. \tag{6.29}$$

Thus, for each case (among the 2^N cases) for $\mathbf{p}(t)$, we can get two possible objective function values (6.21) for Problem P2.2. Totally we can have 2×2^N possible objective function values[4]. Among the 2×2^N possible objective function values, the minimal value is the optimal objective function value of Problem P2.2, and the corresponding power allocation values in (6.26)-(6.29) are the optimal solution of Problem P2.2.

Although Problem P2.2 can be solved by the above Algorithm 6.1, the time and space complexity of this algorithm is $\mathbf{O}\left(2^N\right)$, in which $\mathbf{O}(\cdot)$ means big O notation. Thus, the time and space cost are still unacceptable when N is large. To solve this challenge, next we develop another algorithm to solve Problem P2.2.

[4] It is possible that one or more power allocation values in (6.26), (6.27), (6.28), and (6.29) are negative or more than $p_{\max}$. If this happens, the corresponding objective function values (6.21) should be set to ∞.

6.3.2.2 Algorithm 6.2 to solve Problem P2.2

We have the following lemma for Problem P2.2.

Lemma 16. *Denote* $\mathbf{p}_I^*(t) \triangleq \{p_1^*(t), p_2^*(t), \cdots, p_I^*(t)\}$ $(1 < I \leq N)$ *as the optimal solution of the following problem*

$$\min_{\mathbf{p}_I(t)} \quad O(\mathbf{p}_I(t)) = B(t)\sum_{i=1}^{I} p_i(t) - Q(t)\sum_{i=1}^{I} R_i(t) + \sum_{i=1}^{I} C_i(t) R_i(t) \tag{6.30a}$$

$$s.t. \quad p_i(t) \geq 0, \ \forall i \in \{1, 2, \cdots, I\} \tag{6.30b}$$

$$\sum_{i=1}^{I} p_i(t) \leq p_{\max}. \tag{6.30c}$$

Then, $\mathbf{p}_{I-1}^*(t) \triangleq \left\{p_1^*(t), p_2^*(t), \cdots, p_{I-1}^*(t)\right\}$ *is the optimal solution for the following problem:*

$$\min_{\mathbf{p}_{I-1}(t)} \quad O(\mathbf{p}_{I-1}(t)) = B(t)\sum_{i=1}^{I-1} p_i(t) - Q(t)\sum_{i=1}^{I-1} R_i(t) + \sum_{i=1}^{I-1} C_i(t) R_i(t) \tag{6.31a}$$

$$s.t. \quad p_i(t) \geq 0, \ \forall i \in \{1, 2, \cdots, I-1\} \tag{6.31b}$$

$$\sum_{i=1}^{I-1} p_i(t) = \sum_{i=1}^{I-1} p_i^*(t). \tag{6.31c}$$

Proof. We use proof by contradiction.

We define $\mathbf{p}_J(t) \triangleq \{p_1(t), p_2(t), \cdots, p_J(t)\}, \forall J \in \{1, 2, ..., N\}$. For problem (6.31), assume $\mathbf{p}_{I-1}^*(t)$ is not the optimal solution. Accordingly, we let $\mathbf{p}_{I-1}^\dagger(t)$ which is defined as $\mathbf{p}_{I-1}^\dagger(t) \triangleq \left\{p_1^\dagger(t), p_2^\dagger(t), \cdots, p_{I-1}^\dagger(t)\right\}$ denote the optimal solution of problem (6.31), we have $O\left(\mathbf{p}_{I-1}^\dagger(t)\right) < O\left(\mathbf{p}_{I-1}^*(t)\right)$.

Denote $p_I^\dagger(t)$ as the optimal solution for the following problem

$$\min_{p_I(t)} \quad O(p_I(t)) = B(t)\, p_I(t) - R_I(t)(Q(t) - C_I(t)) \tag{6.32a}$$

$$\text{s.t.} \quad p_I(t) \geq 0 \tag{6.32b}$$

$$p_I(t) \leq p_{\max} - \sum_{i=1}^{I-1} p_i^*(t). \tag{6.32c}$$

Then, $\mathbf{p}_I^\dagger(t) = \{p_1^\dagger(t), p_2^\dagger(t), \cdots, p_I^\dagger(t)\}$ is a feasible solution for problem (6.30).

Thus, we have

$$\begin{aligned} O\left(\mathbf{p}_I^{\dagger}(t)\right) &= O\left(\mathbf{p}_{I-1}^{\dagger}(t)\right) + O\left(p_I^{\dagger}(t)\right) \\ &< O\left(\mathbf{p}_{I-1}^{*}(t)\right) + O\left(p_I^{\dagger}(t)\right) \\ &\le O\left(\mathbf{p}_{I-1}^{*}(t)\right) + O\left(p_I^{*}(t)\right) \\ &= O\left(\mathbf{p}_I^{*}(t)\right), \end{aligned}$$

which contradicts the fact that $\mathbf{p}_I^*(t)$ is the optimal solution of problem (6.30). Therefore, it can be concluded that $\mathbf{p}_{I-1}^*(t)$ is the optimal solution of problem (6.31). This completes the proof. □

When $I = N$, problem (6.30) is identical to Problem P2.2. According to Lemma 15, if $\mathbf{p}_I^*(t)$ is the optimal solution of problem (6.30), we have $\sum_{i=1}^{I} p_i^*(t)$ is a value in χ and χ denotes the set of all values in (6.22). According to Lemma 16, $\mathbf{p}_{I-1}^*(t)$ is also the optimal solution of problem (6.31). Therefore, $\sum_{i=1}^{I-1} p_i^*(t) \in \chi$. Based on these observations, we have the following remark.

Remark 1: Denote all values in (6.22) in ascending order as $\kappa_1, \kappa_2, \kappa_3, ..., \kappa_{|\chi|}$. Let $\mathbf{p}_{I-1}^{*,j}(t)$ denote the optimal solution of problem (6.31) with constraint (6.31c) replaced by $\sum_{i=1}^{I-1} p_i(t) = \kappa_j$. Then, for any given $\kappa_m \in \chi$, the optimal solution of problem (6.30) with adding the constraint $\sum_{i=1}^{I} p_i(t) = \kappa_m$ is given as $\mathbf{p}_I^{*,m}(t) = \left\{\mathbf{p}_{I-1}^{*,n}(t), \kappa_m - \kappa_n\right\}$ where $n = \arg\min_{j=1,2,\cdots,m}\left\{O\left(\mathbf{p}_{I-1}^{*,j}(t)\right) + O\left(\kappa_m - \kappa_j\right)\right\}$.

According to Remark 1, we propose another algorithm, named Algorithm 6.2, to obtain the optimal solution of Problem P2.2.

In the algorithm, Steps 1–11 are to find all values in χ. Steps 12–17 are to use a recursive method (based on Remark 1) to find the optimal solution of problem (6.30) with adding the constraint $\sum_{i=1}^{I} p_i(t) = \kappa_m$ with $I = 2, ..., N$ and $m = 1, 2, ..., |\chi|$. Steps 18–20 deal with the case $I = N$, and search all possible values in χ to find the optimal solution that minimizes the objective function of Problem P2.2.

According to Lemma 15, the size of χ is $\mathbf{O}(N^2)$. Thus, the complexity in Step 13, Step 14, and Step 15 in Algorithm 6.2 is $\mathbf{O}(N)$, $\mathbf{O}(N^2)$, and $\mathbf{O}(N^2)$, respectively. Accordingly, the time and space complexity of Algorithm 1 is $\mathbf{O}(N^5)$, which is acceptable.

Note that the solution by Algorithm 6.2 (and Algorithm 6.1) is the optimal solution of Problem P2.2. Thus, the optimal solution of Problem P2 can be obtained by the

Algorithm 6.2 The proposed algorithm to obtain the optimal solution of Problem P2.2

$\chi \leftarrow \{0, p_{\max}\}$
for $i \leftarrow 1$ to N **do**
  if $0 < \frac{X_i(t)}{B(t)} - \frac{v}{g_i(t)} < p_{\max}$ **then**
    $\chi \leftarrow \chi \cup \left\{ \frac{X_i(t)}{B(t)} - \frac{v}{g_i(t)} \right\}$
  end if
  for $j \leftarrow i+1$ to N **do**
    if $0 < \frac{(X_j(t)g_j(t) - X_i(t)g_i(t))v}{g_i(t)g_j(t)(X_i(t)-X_j(t))} < p_{\max}$ **then**
      $\chi \leftarrow \chi \cup \left\{ \frac{(X_j(t)g_j(t) - X_i(t)g_i(t))v}{g_i(t)g_j(t)(X_i(t)-X_j(t))} \right\}$
    end if
  end for
end for
Set $\mathbf{p}_1^{*,j}(t) \leftarrow \kappa_j$ for any $j = 1, 2, \cdots, |\chi|$
for $I \leftarrow 2$ to N **do**
  for $m \leftarrow 1$ to $|\chi|$ **do**
    $\mathbf{p}_I^{*,m}(t) = \left\{ \mathbf{p}_{I-1}^{*,n}(t), \kappa_m - \kappa_n \right\}$ where $n = \arg\min_{j=1,2,\cdots,m} \left\{ O\left(\mathbf{p}_{I-1}^{*,j}(t)\right) + O\left(\kappa_m - \kappa_j\right) \right\}$
  end for
end for
$z = \arg\min_{j=1,2,\cdots,|\chi|} \left\{ O\left(\mathbf{p}_N^{*,j}(t)\right) \right\}$
$\mathbf{p}_N^{*}(t) \leftarrow \mathbf{p}_N^{*,z}(t)$
return $\mathbf{p}_N^{*}(t)$

closed-form optimal solution (6.20) for Problem P2.1 and by using Algorithm 6.2 (or Algorithm 6.1) to solve Problem P2.2.

6.3.3 Relationship between Problems P1 and Problem P2

Recall that our original optimization problem is Problem P1. Next we will discuss the relationship between Problem P1 and Problem P2.

Let Ω_1 and Ω_2 denote the optimal policy for Problems P1 and P2, respectively. Then, the value of the objective function U in Problem P1 based on Ω_1 and Ω_2 are denoted as U_{Ω_1} and U_{Ω_2}, respectively. The gap between U_{Ω_1} and U_{Ω_2} is given in the following lemma.

Lemma 17. *The gap between U_{Ω_1} and U_{Ω_2} is*

$$U_{\Omega_1} - U_{\Omega_2} \leq \frac{M}{V}. \tag{6.33}$$

Proof. We use the method in [57] to prove this lemma. For any $\sigma > 0$, there is a policy Ω which meets all constraints of Problem P2 (i.e., Ω is a feasible solution of

Problem P2) and satisfies the following inequalities [57]:

$$E^{\Omega}\left[\sum_{i=1}^{N} p_i\left(t\right)-\bar{p}\middle|\mathcal{S}(t)\right]\leq\sigma, \tag{6.34}$$

$$E^{\Omega}\left[D\left(t\right)-\sum_{i=1}^{N} R_i\left(t\right)\middle|\mathcal{S}(t)\right]\leq\sigma, \tag{6.35}$$

$$E^{\Omega}\left[R_i\left(t\right)-L_i\left(t\right)\middle|\mathcal{S}(t)\right]\leq\sigma, \tag{6.36}$$

$$-E^{\Omega}\left[\log\left(1+D\left(t\right)\right)\middle|\mathcal{S}(t)\right]\leq -U_{\Omega_1}+\sigma, \tag{6.37}$$

where $E^{\Omega}[\cdot]$ means expectation when policy Ω is applied.

When policy Ω is applied, from (6.12), we have the following for the Lyapunov drift-plus-penalty function $\Delta_V(t)$ under the policy Ω, denoted as $\Delta_V^{\Omega}\left(t\right)$:

$$\begin{aligned}\Delta_V^{\Omega}\left(t\right) &\leq M+E^{\Omega}\left[B\left(t\right)\left(\sum_{i=1}^{N} p_i\left(t\right)-\bar{p}\right)\middle|\mathcal{S}\left(t\right)\right]\\ &\quad+E^{\Omega}\left[Q\left(t\right)\left(D\left(t\right)-\sum_{i=1}^{N} R_i\left(t\right)\right)\middle|\mathcal{S}\left(t\right)\right]\\ &\quad+E^{\Omega}\left[\sum_{i=1}^{N}\left(C_i\left(t\right)\left(R_i\left(t\right)-L_i\left(t\right)\right)\right)\middle|\mathcal{S}\left(t\right)\right]\\ &\quad-VE^{\Omega}\left[\log\left(1+D\left(t\right)\right)\middle|\mathcal{S}\left(t\right)\right]\\ &\leq M-VU_{\Omega_1}+\sigma\cdot A,\end{aligned} \tag{6.38}$$

in which A is the maximal possible value of $B(t)+Q(t)+\sum_{i=1}^{N} C_i(t)+V$.

Since the policy Ω_2 is the optimal solution of Problem P2 (i.e., it minimizes the upper bound of $\Delta_V(t)$), we have

$$\Delta_V^{\Omega_2}\left(t\right)\leq\Delta_V^{\Omega}\left(t\right), \tag{6.39}$$

in which $\Delta_V^{\Omega_2}\left(t\right)$ is the Lyapunov drift-plus-penalty function $\Delta_V(t)$ under the policy Ω_2.

Let $\sigma\to 0$. Then from (6.38) and (6.39) we have

$$\Delta_V^{\Omega_2}\left(t\right)\leq M-VU_{\Omega_1}. \tag{6.40}$$

Thus, from (6.10) and (6.11) we have

$$E^{\Omega_2}\left[Y\left(t+1\right)-Y\left(t\right)\right]-VE^{\Omega_2}\left[\log\left(1+D\left(t\right)\right)\right]\leq M-VU_{\Omega_1}. \tag{6.41}$$

Then, taking summation of the inequalities in (6.41) for $t = 0, 1, \cdots, \tau - 1$, we get

$$E^{\Omega_2}\left[Y(\tau)\right] - E^{\Omega_2}\left[Y(0)\right] - V\sum_{t=0}^{\tau-1} E^{\Omega_2}\left[\log\left(1 + D(t)\right)\right] \leq M\tau - VU_{\Omega_1}\tau. \tag{6.42}$$

All buffers are set as empty at $t = 0$. Thus, $E^{\Omega_2}\left[Y(\tau)\right] - E^{\Omega_2}\left[Y(0)\right] \geq 0$. Letting $\tau \to \infty$, we have the following inequality

$$\lim_{\tau\to\infty} \frac{\sum_{t=0}^{\tau-1} E^{\Omega_2}\left[\log\left(1 + D(t)\right)\right]}{\tau} \geq U_{\Omega_1} - \frac{M}{V}. \tag{6.43}$$

Note that based on the Jensen's inequality, the left-hand side of (6.43) is actually not more than the objective function of Problem P1 when policy Ω_2 is applied. Accordingly, we have $U_{\Omega_1} - U_{\Omega_2} \leq \frac{M}{V}$. This completes the proof. □

Similarly, we have the following inequality for summation of the average queue length of the task buffers of the IoT device and fog nodes:

$$E\left[\sum_{i=1}^{N} C_i(t) + Q(t)\right] \leq \frac{M}{H} + \mathbf{O}(V) \tag{6.44}$$

where H is a constant. The proof is similar to that in [118], and thus, is omitted here. From (6.33) and (6.44), we have the following observation for the tradeoff between the system utility and the average length of task buffers. Note that the average execution delay of the data is determined by the average amount of data buffered in the fog computing system. If a larger amount of data are buffered in the fog computing system, the data have to wait more time to be computed, and thus, a larger execution delay is obtained. Thus, if V takes a large value, it benefits the system utility, at the cost of possible large average delay. If V takes a small value, it tends to reduce the average delay, at the cost of possible small system utility.

6.4 Performance Evaluation

In this section, simulation results, which are obtained by using Matlab software, are provided to evaluate the performance of our proposed scheme. The parameters used in the simulation are given in Table 6.1. Similar settings have been widely considered in existing works, such as [110], [111] and [120].

Table 6.1: The parameters in the simulation

Parameters	Value
T	30 ms
N	5
W	15 MHz
$f_i(t)$	uniform in [0.4 GHz, 0.5 GHz]
φ	30 cycles/bit
υ	-80 dBm
$\{d_1, d_2, d_3, d_4, d_5\}$	$\{50, 70, 90, 110, 130\}$ meters
α	4
$\bar{p}$	1.5 Watt
$p_{\max}$	3 Watt

Firstly, we verify the correctness of Algorithm 6.1 and Algorithm 6.2. Accordingly, an exhaustive search (ES) scheme is introduced. In ES scheme, to solve Problem P2, we search all feasible values of $D(t)$ and $\mathbf{p}(t)$ to minimize the objective function. Thus, the ES scheme can obtain the optimal $D(t)$ and $\mathbf{p}(t)$ for Problem P2 at each time slot. Since the amount of data in task buffers and the channel gains vary with time, the ES scheme needs to be performed at each time slot. Thus, the computation complexity is $\mathbf{O}(\tau \Xi_{D(t)} \prod_{i=1}^{N} \Xi_{p_i(t)})$, where τ is the number of time slots, $\Xi_{D(t)}$ is the number of feasible values for $D(t)$ after quantization, and $\Xi_{p_i(t)}$ is the number of feasible values for $p_i(t)$ after quantization. The simulation statistics are collected over 10,000 time slots. The simulation result is given in Fig. 6.2. In our simulation results, "system utility" means the amount of data which are computed at fog nodes. It can be seen that Algorithm 6.1 and Algorithm 6.2 achieve the same system utility as that achieved by the ES scheme, thus verifying that Algorithm 6.1 and Algorithm 6.2 provide optimal solution to Problem P2.2 with much less complexity than that of the ES scheme.

Then, we investigate how V affects the system utility and the average length of task buffers (q) in our proposed scheme, where the average length of task buffers q is defined as $q = \frac{\sum_{t=0}^{\tau-1} \left(\sum_{i=1}^{N} C_i(t) + Q(t) \right)}{\tau}$ with τ being the number of time slots. The simulation result is given in Fig. 6.3. The system utility of our proposed scheme grows with the increase of V. Similarly, increasing V raises the average length of task buffers of our proposed scheme. Fig. 6.3 also shows the average execution delay of the tasks. It can be seen that the average execution delay has the same trend as the

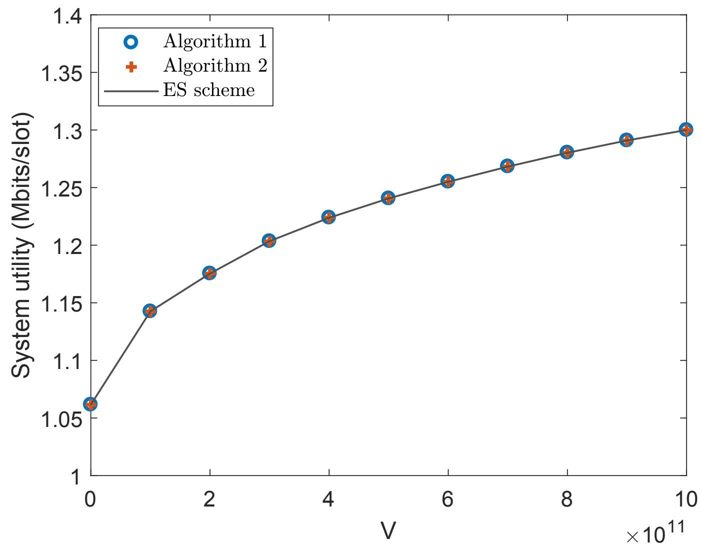

Figure 6.2: System utility of proposed algorithms and the ES scheme verse the parameter V.

average length of task buffers. Accordingly, there is a tradeoff between the system utility and the average execution delay. The system utility tends to keep stable when V keeps increasing beyond a large value. The reason is that the computing capacity of fog nodes is limited (in other words, the system utility is bounded by the computation capacity of the fog nodes).

We also investigate how the computation capacity of the fog nodes affects the system utility. The computation capacity of each fog node is identical to those of other fog nodes. The average computation capacity of each fog node, denoted $\bar{f}$, varies in the simulation. The simulation result is given in Fig. 6.4. The system utility grows with the increase of $\bar{f}$. It confirms our intuitive understanding that, with higher computation capacity of fog nodes, more data can be computed at fog nodes, which means that more system utility can be achieved. However, the system utility keeps stable when $\bar{f}$ keeps increasing beyond a large value. It is because the amount of data that can be transmitted to fog nodes is limited due to the constraints of the transmission power (in other words, the system utility is also bounded by the transmission power of the IoT device).

In order to evaluate the performance of our proposed scheme, we compare with the following benchmark schemes.

1. NOMA equal power (NOMA-EP) scheme: In this scheme, the IoT device also

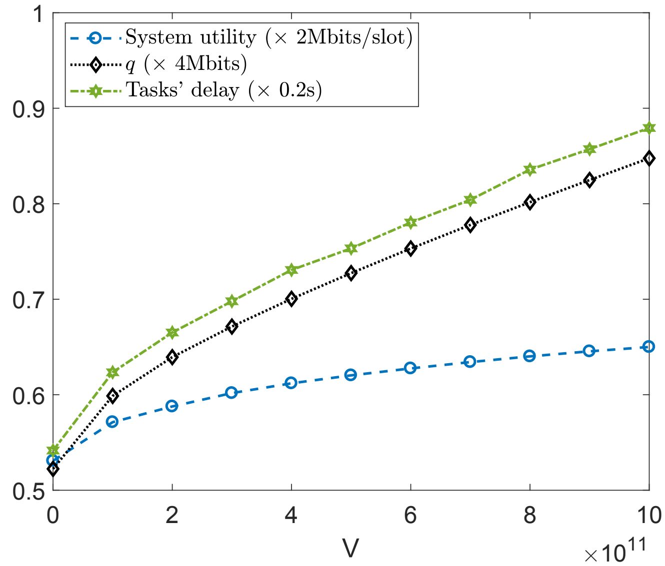

Figure 6.3: Tradeoff among system utility, average length of task buffers, and average execution delay of tasks.

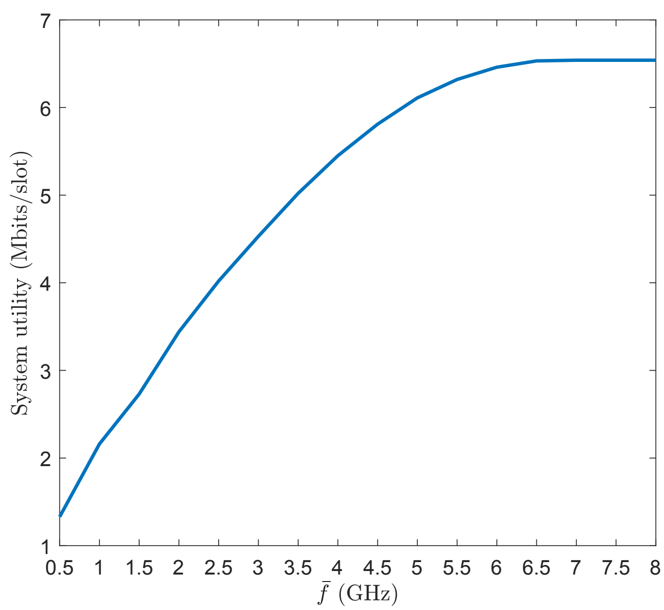

Figure 6.4: System utility with different computation capacity f.

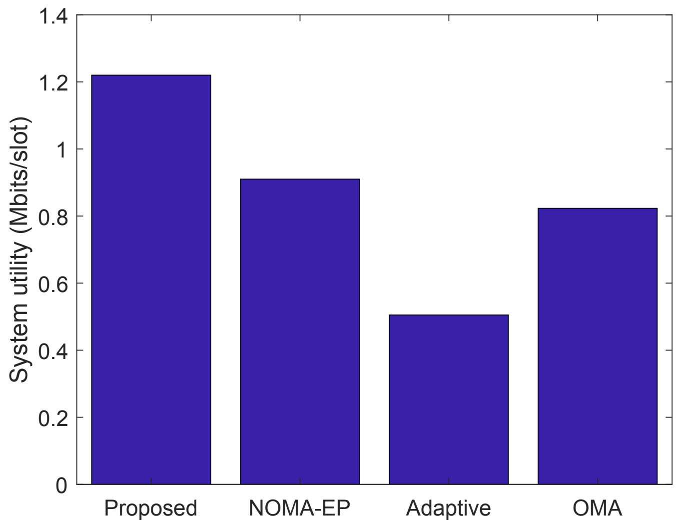

Figure 6.5: System utility with different schemes.

uses NOMA for computation offloading. The power that is allocated for the transmission to each fog node is the same. Thus, $p_i(t) = \frac{\bar{p}}{N}$.

2. Adaptive scheme: In this scheme, in current tth time slot, the IoT device only offloads to one fog node, which is the fog node that has the smallest $C_i(t)$. The transmission power is set as $\bar{p}$.

3. OMA scheme [110]: During existing works, only OMA is adopted in the fog computing system for offloading to multiple fog nodes. In [110], a computation offload scheme with OMA is proposed to minimize the latency and the energy consumption, which is used here for comparison.

We fix the value of V as 7×10^{11}.[5] Fig. 6.5 and Fig. 6.6 show the simulation results of the system utility and the average length of task buffers, respectively, with different schemes. The simulation results demonstrate that the performance of our proposed scheme is better than other schemes. To be specific, our proposed scheme has the largest system utility and smallest backlog in the task buffers. In adaptive scheme, it transmits data to the fog node that has the lowest $C_i(t)$. Thus, for the fog node which has the longest distance to the IoT device, i.e., fog node 5, it has the worst channel, and thus, the amount of data that can be sent to fog node 5 over

[5]We just take this value as an example. Similar results can be obtained for different values of V.

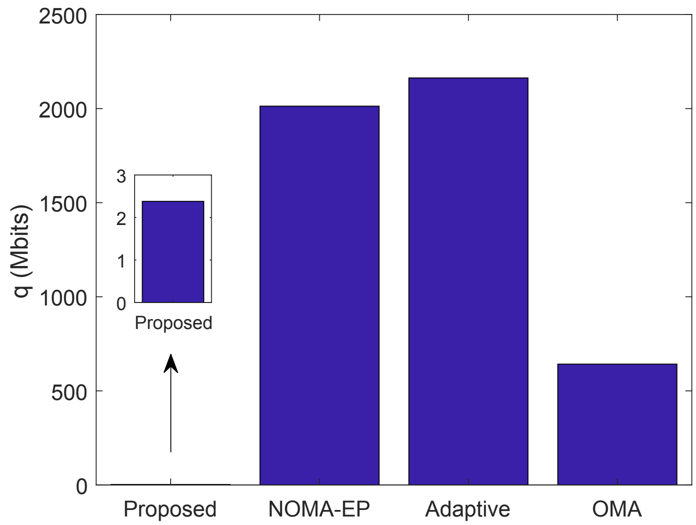

Figure 6.6: Average length of task buffers with different schemes.

wireless channel is small. So fog node 5 has a high chance to have the lowest $C_i(t)$, and accordingly, has a high chance to be selected by the IoT device in each time slot. Therefore, the system utility of the adaptive scheme is the worst, as shown in Fig. 6.5.

Fig. 6.7 shows the length of virtual buffer B with different V. It can be seen that B increases with the growth of V. When V keeps increasing beyond a large value, the virtual buffer length keeps stable due to constraint (6.4c) and the limited computing capacity.

Fig. 6.8 shows the average length of each fog node's task buffer $C_i(t)$ with different schemes. Thus, the load balance performance across multiple fog nodes can be shown by the simulation results. It is observed that our proposed scheme well utilizes the computation capacity of all fog nodes. Thus, our scheme achieves a good load balance across multiple fog nodes, which can guarantee the fairness of each fog node. The adaptive scheme and the OMA scheme can also balance the computation tasks of all fog nodes. However, according to Fig. 6.5 and Fog. 6.6, these two schemes achieve a smaller system utility and a larger backlog compared with our propose scheme. In NOMA-EP scheme, the backlog of fog node 1's task buffer is much larger than backlog of other fog nodes' task buffers. The reason is as follows. Fog node 1 has higher average channel gain than those of other fog nodes. When equal power allocation is adopted in NOMA-EP, higher transmission rate can be achieved for fog node 1 than

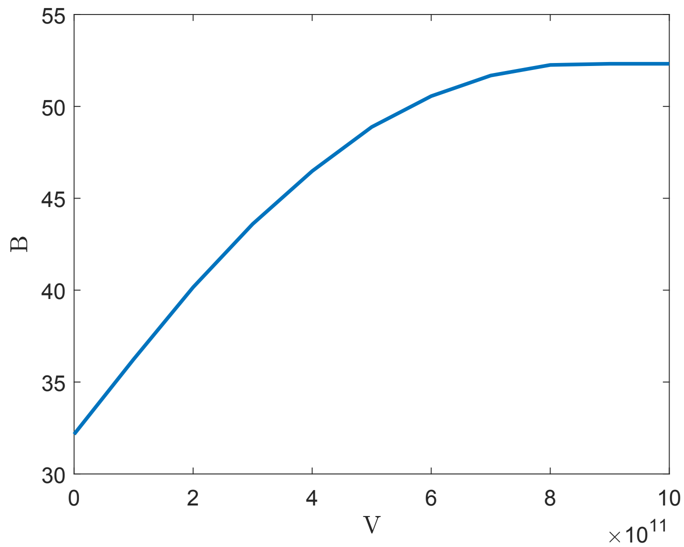

Figure 6.7: Length of virtual buffer B versus the parameter V.

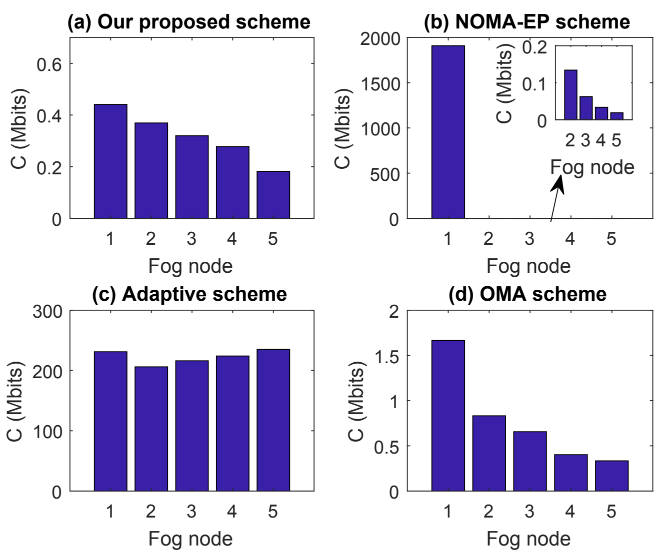

Figure 6.8: Average length of task buffer $C_i(t)$ with different schemes.

those for any other fog node, and thus, more data are sent to fog node 1 than those sent to any other fog node.

6.5 Conclusion

In this chapter, we propose an optimal computation offloading scheme with downlink NOMA for a fog computing system. To achieve the maximal system utility, the input data size to the IoT device's task buffer and the transmit power to the fog nodes are optimized. By Lyapunov method, the problem is transformed to an online optimization problem that only involves instantaneous variables of the current time slot. To solve the non-convex online optimization problem, an algorithm with polynomial computation complexity is proposed.

Chapter 7

Conclusions and Future Research

7.1 Conclusions

This thesis focuses on spectrum efficiency enhancement in wireless communication networks. Cognitive radio, opportunistic scheduling, and NOMA are promising techniques which can largely improve the spectrum efficiency. However, some challenges exist in deploying them in practical wireless networks, and thus, we aim at QoS provisioning of networks by solving these challenges.

In Chapter 3, a slot length configuration scheme is proposed. The scheme tells how to determine the length of a time slot after obtaining the channel state by spectrum sensing. By introducing some interesting properties of the research problem, an algorithm is proposed to find the optimal slot length.

In Chapter 4, the opportunistic scheduling problem is modeled as an SMDP which reduces the implementation complexity. Then, a model-based scheduling method and a model-free scheduling method are proposed to derive the optimal scheduling policy for fully explored networks and partially explored networks, respectively. Further, in Chapter 5, distributed opportunistic channel access in energy-limited wireless cooperative networks is investigated. A DOS scheme is proposed to maximize the average throughput of the network. Then, an optimal stopping strategy, which has threshold-based structure, is derived to guide the user to decide whether to utilize the channel access opportunity.

In Chapter 6, the power allocation problem of NOMA is investigated in computation offloading, which is a major part in fog computing systems. By exploring the original problem, some properties are derived, and thus, the original problem which is a non-convex problem is solved by an algorithm with polynomial complexity. Accordingly, the IoT device can decide how to offload its tasks to fog nodes.

7.2 Future Research

In Chapter 3, the slot length configuration problem in cognitive radio networks is investigated. In this work, the spectrum sensing duration is decided by the expected $(P_\mathrm{d}, P_\mathrm{f})$ pair. If no such pair is provided, the following question arises, how to jointly determine the spectrum sensing length and the slot length. Therefore, how to derive an efficient method to find the optimal sensing duration and the slot length will be investigated in the future.

Opportunistic scheduling and NOMA are two important techniques to improve the spectrum efficiency. Therefore, the combination of them, which is how to realize opportunistic scheduling in wireless networks with NOMA, is very meaningful. Due to the effects of NOMA, the traditional opportunistic scheduling strategy may not be suitable to the networks with NOMA. Accordingly, how to derive an optimal strategy is a challenging and meaningful problem, and will be investigated in the future.

The computation offloading problem in fog computing system with NOMA is investigated in Chapter 6. Due to the limited computation capacity, the task which cannot be computed by the fog nodes will be computed by the IoT device or the cloud server. Therefore, future works may consider the IoT device's and cloud server's computation capacity, and optimally distribute the IoT device's tasks to its local CPU, the cloud server, and the fog nodes

Bibliography

[1] C. V. N. Index, "Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021 white paper," *White paper, CISCO*, 2017.

[2] J. Cheon and H.-S. Cho, "Power allocation scheme for non-orthogonal multiple access in underwater acoustic communications," *Sensors*, vol. 17, no. 11, p. 2465, Oct. 2017.

[3] A. Cocchia, "Smart and digital city: A systematic literature review," in *Smart city*. Springer, 2014, pp. 13–43.

[4] G. Varrall, *5G Spectrum and Standards*. Artech House, 2016.

[5] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016.

[6] NSF, "2015 NSF ears workshop final report," https://earsworkshop.wireless.vt.edu/pdfs/EARS_2015_workshop_report_final_ver.pdf, 2015.

[7] A. Dobaria and S. Sodhatar, "A literature survey on efficient spectrum utilization: cognitive radio technology," *International Journal of Innovative and Emerging Research in Engineering*, vol. 2, no. 1, pp. 72–75, 2015.

[8] P. W. Dent, "TDMA/FDMA/CDMA hybrid radio access methods," Jul. 23 1996, U.S. Patent no. 5,539,730.

[9] M. Morelli, C.-C. J. Kuo, and M.-O. Pun, "Synchronization techniques for Orthogonal Frequency Division Multiple Access (OFDMA): A tutorial review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394–1427, Aug. 2007.

[10] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “A survey on spectrum management in cognitive radio networks,” *IEEE Communications magazine*, vol. 46, no. 4, pp. 40–48, Apr. 2008.

[11] X. Liu and N. S. Shankar, “Sensing-based opportunistic channel access,” *Mobile Networks and Applications*, vol. 11, no. 4, pp. 577–591, Aug. 2006.

[12] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, “Non-orthogonal multiple access (NOMA) for cellular future radio access,” in *Proceedings of IEEE Vehicular Technology Conference (VTC Spring)*, Jun. 2013, pp. 1–5.

[13] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[14] L. Yang and G. B. Giannakis, “Ultra-wideband communications: An idea whose time has come,” *IEEE Signal Processing Magazine*, vol. 21, no. 6, pp. 26–54, Nov. 2004.

[15] P. Wang, Y. Li, L. Song, and B. Vucetic, “Multi-gigabit millimeter wave wireless communications for 5G: from fixed access to cellular networks,” *IEEE Communications Magazine*, vol. 53, no. 1, pp. 168–178, Jan. 2015.

[16] I. 802.22.2-2012(TM), “Standard for recommended practice for installation and deployment of IEEE 802.22 systems,” 2012.

[17] J. Zou, H. Xiong, D. Wang, and C. W. Chen, “Optimal power allocation for hybrid overlay/underlay spectrum sharing in multiband cognitive radio networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1827–1837, May 2013.

[18] Y.-C. Liang, Y. Zeng, E. C. Peh, and A. T. Hoang, “Sensing-throughput tradeoff for cognitive radio networks,” *IEEE transactions on Wireless Communications*, vol. 7, no. 4, pp. 1326–1337, Apr. 2008.

[19] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE communications surveys & tutorials*, vol. 11, no. 1, pp. 116–130, 1st Quart., 2009.

[20] F. Salahdine, H. El Ghazi, N. Kaabouch, and W.F. Fihri, "Matched filter detection with dynamic threshold for cognitive radio networks," in *Proceedings of IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM)*. Oct. 2015, pp. 1–6.

[21] L. Claudino and T. Abrão, "Spectrum sensing methods for cognitive radio networks: A review," *Wireless Personal Communications*, vol. 95, no. 4, pp. 5003–5037, Aug. 2017.

[22] P. D. Sutton, K. E. Nolan, and L. E. Doyle, "Cyclostationary signatures in practical cognitive radio applications," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 13–24, Jan. 2008.

[23] P. Aparna and M. Jayasheela, "Cyclostationary feature detection in cognitive radio using different modulation schemes," *International Journal of Computer Applications*, vol. 47, no. 21, pp. 12–16, Jun. 2012.

[24] S. Atapattu, C. Tellambura, and H. Jiang, "Analysis of area under the ROC curve of energy detection," *IEEE Transactions on Wireless Communications*, vol. 9, no. 3, pp. 1216–1225, Mar. 2010.

[25] X. Ge, H. Jin, J. Zhu, J. Cheng, and V. C. M. Leung, "Exploiting opportunistic scheduling in uplink wiretap networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4886–4897, Oct. 2016.

[26] T. Paul and O. Tokunbdo, "Wireless LAN comes of age: Understanding the IEEE 802.11n amendment," *IEEE Circuits and Systems Magazine*, vol. 8, no. 1, pp. 28–54, Apr. 2008.

[27] N. U. Hassan and M. Assaad, "Low complexity margin adaptive resource allocation in downlink MIMO-OFDMA system," *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp. 3365–3371, Aug. 2009.

[28] E. Yaacoub, A. M. El-Hajj, and Z. Dawy, "Weighted ergodic sum-rate maximisation in uplink orthogonal frequency division multiple access and its achievable rate region," *IET Communications*, vol. 4, no. 18, pp. 2217–2229, Jan. 2011.

[29] Z. Zhang, *Optimal Opportunistic Channel Access in Wireless Communication Networks*. PhD Thesis of the University of Alberta (Canada), 2013.

[30] D. Zheng, W. Ge, and J. Zhang, "Distributed opportunistic scheduling for Ad Hoc networks with random access: An optimal stopping approach," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 205–222, Dec. 2008.

[31] Z. Zhang, S. Zhou, H. Jiang, and L. Dong, "Opportunistic cooperative channel access in distributed wireless networks with Decode-and-Forward relays," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1778–1781, Jul. 2015.

[32] L. Dong, Y. Wang, H. Jiang, Z. Zhang, and Shuai Zhou, "Two-level distributed opportunistic scheduling in DF relay networks," *IEEE Wireless Communications Letters*, vol. 4, no. 5, pp. 477–480, Jun. 2015.

[33] J. N. Laneman, D. N. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information theory*, vol. 50, no. 12, pp. 3062–3080, Jan. 2005.

[34] Y. Zhao, R. Adve, and T. J. Lim, "Improving Amplify-and-Forward relay networks: optimal power allocation versus selection," in *Proceedings of IEEE International Symposium on Information Theory*, Jul. 2006, pp. 1234–1238.

[35] T. Wang, A. Cano, G. B. Giannakis, and J. N. Laneman, "High-performance cooperative demodulation with Decode-and-Forward relays," *IEEE Transactions on Communications*, vol. 55, no. 7, pp. 1427–1438, Aug. 2007.

[36] S. S. Ikki and M. H. Ahmed, "Performance analysis of incremental-relaying cooperative-diversity networks over Rayleigh fading channels," *IET Communications*, vol. 5, no. 3, pp. 337–349, 2011.

[37] 3GPP, "Justification for NOMA in new study on enhanced multi-user transmission and network assisted interference cancellation for LTE," RP-141936, Dec. 2014.

[38] I. SM, M. Zeng, and D. OA, "NOMA in 5G systems: Exciting possibilities for enhancing spectral efficiency," *arXiv preprint arXiv:1706.08215*, 2017.

[39] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, "A survey of non-orthogonal multiple access for 5G," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2294–2323, 3rd Quart., 2018.

[40] S. R. Islam, N. Avazov, O. A. Dobre, and K. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 721–742, 2nd Quart., 2016.

[41] K. Higuchi and A. Benjebbour, "Non-orthogonal multiple access (NOMA) with successive interference cancellation for future radio access," *IEICE Transactions on Communications*, vol. 98, no. 3, pp. 403–414, Mar. 2015.

[42] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 533–545, Mar. 2008.

[43] Z. Chen, L. X. Cai, Y. Cheng, and H. Shan, "Sustainable cooperative communication in wireless powered networks with energy harvesting relay," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8175–8189, Oct. 2017.

[44] T. J. Kazmierski and S. Beeby, *Energy Harvesting Systems*. Springer, 2014.

[45] M.-L. Ku, W. Li, Y. Chen, and K.J. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2641–2657, Dec. 2015.

[46] K.-H. Liu and T.-L. Kung, "Performance improvement for RF energy-harvesting relays via relay selection," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8482–8494, Apr. 2017.

[47] H. Ding, D. B. da Costa, X. Wang, U. S. Dias, R. T. S. Junior, and J. Ge, "On the effects of LOS path and opportunistic scheduling in energy harvesting relay

systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8506–8524, Oct. 2016.

[48] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, C.L. I, and H. V. Poor, "Application of non-orthogonal multiple access in LTE and 5G networks," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 185–191, Feb. 2017.

[49] T. S. Ferguson, *Optimal Stopping and Applications*, 2012.

[50] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, 2014.

[51] M. Petrik and S. Zilberstein, "Average-reward decentralized markov decision processes." in *Proceedings of International Joint Conferences on Artificial Intelligence*, Jan. 2007, pp. 1997–2002.

[52] R. Bellman, *Dynamic Programming.* Courier Corporation, 2013.

[53] Q. Hu and W. Yue, *Markov Decision Processes With Their Applications.* Springer Science & Business Media, 2007.

[54] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 2011.

[55] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[56] M. J. Neely, "Stability and probability 1 convergence for queueing networks via Lyapunov optimization," *Journal of Applied Mathematics*, vol. 2012, p. 831909, Apr. 2012.

[57] 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.

[58] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, pp. 1–144, 2006.

[59] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems," *IEEE transactions on wireless communications*, vol. 15, no. 11, pp. 7244–7257, Aug. 2016.

[60] Y. Gao, B. Xia, K. Xiao, Z. Chen, X. Li, and S. Zhang, "Theoretical analysis of the dynamic decode ordering SIC receiver for uplink NOMA systems," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2246–2249, Jun. 2017.

[61] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2181–2195, Jul. 2017.

[62] D. Zhang, Z. Chen, J. Ren, N. Zhang, M. K. Awad, H. Zhou, and X. Shen, "Energy-harvesting-aided spectrum sensing and data transmission in heterogeneous cognitive radio sensor network," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 831–843, Apr. 2016.

[63] Y. Zhang, J. Zheng, and H.-H. Chen, *Cognitive Radio Networks: Architectures, Protocols, and Standards.* CRC press, 2016.

[64] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1277–1304, 2nd Quart., 2017.

[65] W. Chung, S. Park, S. Lim, and D. Hong, "Spectrum sensing optimization for energy-harvesting cognitive radio systems," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2601–2613, May 2014.

[66] C.-H. Liu, A. Azarfar, J.-F. Frigon, B. Sanso, and D. Cabric, "Robust cooperative spectrum sensing scheduling optimization in multi-channel dynamic spectrum access networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2094–2108, Sep. 2015.

[67] H. Zhang, Y. Nie, J. Cheng, V. C. M. Leung, and A. Nallanathan, "Sensing time optimization and power control for energy efficient cognitive small cell with

imperfect hybrid spectrum sensing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 730–743, Nov. 2016.

[68] C. Jiang, N. C. Beaulieu, and C. Jiang, "A novel asynchronous cooperative spectrum sensing scheme," in *Proceedings of IEEE International Conference on Communications (ICC)*, Jun. 2013, pp. 2606–2611.

[69] R. Fan, J. An, H. Jiang, and X. Bu, "Adaptive channel selection and slot length configuration in cognitive radio," *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2636–2648, Jul. 2016.

[70] J. Liu, M. Sheng, L. Liu, and J. Li, "Effect of densification on cellular network performance with bounded pathloss model." *IEEE Communications Letters*, vol. 21, no. 1, pp. 346–349, Feb. 2017.

[71] X. Huang, T. Han, and N. Ansari, "On green-energy-powered cognitive radio networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 827–842, 2nd Quart., 2015.

[72] D. V. Widder, *Laplace Transform (PMS-6).* Princeton University Press, 2015.

[73] X. Chen, L. Guo, X. Li, C. Dong, J. Lin, and P. T. Mathiopoulos, "Secrecy rate optimization for cooperative cognitive radio networks aided by a wireless energy harvesting jammer," *IEEE Access*, vol. 6, pp. 34127–34134, Jul. 2018.

[74] B. L. Lee and S. K. Fan, "A two-dimensional search algorithm for dose-finding trials of two agents," *Journal of Biopharmaceutical Statistics*, vol. 22, no. 4, pp. 802–818, Jul. 2012.

[75] Q. Wang, K. Jaffrès-Runser, J.-L. Scharbarg, C. Fraboul, Y. Sun, J. Li, and Z. Li, "A thorough analysis of the performance of delay distribution models for IEEE 802.11 DCF," *Ad Hoc Networks*, vol. 24, pp. 21–33, 2015.

[76] Y. Lu and A. Duel-Hallen, "Channel-aware spectrum sensing and access for mobile cognitive radio ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2471–2480, Apr. 2016.

[77] N. T. Do, D. B. da Costa, T. Q. Duong, V. N. Q. Bao, and B. An, "Exploiting direct links in multiuser multirelay SWIPT cooperative networks with opportunistic scheduling," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5410–5427, Aug. 2017.

[78] A. Garcia-Saavedra, A. Banchs, P. Serrano, and J. Widmer, "Adaptive mechanism for distributed opportunistic scheduling," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3494–3508, Jun. 2015.

[79] S. Bulusu, N. B. Mehta, and S. Kalyanasundaram, "Rate adaptation, scheduling, and mode selection in D2D systems with partial channel knowledge," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1053–1065, Feb. 2018.

[80] K. Wang, "Optimally myopic scheduling policy for downlink channels with imperfect state observation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5856–5867, Jul. 2018.

[81] P. Nguyen and B. Rao, "Optimal fair opportunistic scheduling for wireless systems via classification framework," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 2, pp. 185–199, Jun. 2015.

[82] J. Zhu, Y. Song, D. Jiang. and H. Song, "A new deep-Q-learning-based transmission scheduling mechanism for the cognitive internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2375–2385, Aug. 2018.

[83] S. Liu, X. Hu, and W. Wang, "Deep reinforcement learning based dynamic channel allocation algorithm in multibeam satellite systems," *IEEE Access*, vol. 6, pp. 15 733–15 742, 2018.

[84] B. Liang, *Mobile Edge Computing*. Cambridge University Press, 2017.

[85] K. Wu, C. Tellambura, and H. Jiang, "Optimal transmission policy in energy harvesting wireless communications: A learning approach," in *Proceedings of IEEE International Conference on Communications (ICC),* Jun. 2017, pp. 1–6.

[86] T. O. Kim, C. N. Devanarayana, A. S. Alfa, and B. D. Choi, "An optimal admission control protocol for heterogeneous multicast streaming services," *IEEE Transactions on Communications*, vol. 63, no. 6, pp. 2346–2359, Jun. 2015.

[87] Q. Li, L. Zhao, J. Gao, H. Liang, L. Zhao, and X. Tang, "SMDP-based coordinated virtual machine allocations in cloud-fog computing systems," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1977–1988, Jun. 2018.

[88] A. Abdrabou and W. Zhuang, "Service time approximation in IEEE 802.11 single-hop ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 305–313, Jan. 2008.

[89] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, "Hethetnets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2252–2265, Oct. 2015.

[90] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 178–192, 2009.

[91] M. Laghate, P. Urriza, and D. Cabric, "Channel access method classification for cognitive radio applications," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 70–73, Feb. 2018.

[92] W. Mao, X. Wang, and S. Wu, "Distributed opportunistic scheduling with QoS constraints for wireless networks with hybrid links," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8511–8527, Oct. 2016.

[93] H. Li, C. Huang, P. Zhang, S. Cui, and J. Zhang, "Distributed opportunistic scheduling for energy harvesting based wireless networks: A two-stage probing approach," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1618–1631, Jun. 2016.

[94] D. Zhou, M. Sheng, R. Liu, Y. Wang, and J. Li, "Channel-aware mission scheduling in broadband data relay satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 1052–1064, May 2018.

[95] Z. Zhang and H. Jiang, "Distributed opportunistic channel access in wireless relay networks," *arXiv preprint arXiv:1102.5461*, 2011.

[96] H. Liu, K. J. Kim, K. S. Kwak, and H. V. Poor, "QoS-constrained relay control for full-duplex relaying with SWIPT," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2936–2949, May 2017.

[97] R. G. Gallager, *Discrete Stochastic Processes*. Springer Science & Business Media, 2012.

[98] L. Yang, H. Zhang, M. Li, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[99] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. ACM, Aug. 2012, pp. 13–16.

[100] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, Mar. 2014, pp. 169–186.

[101] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint arXiv:1502.01815*, 2015.

[102] J. Fan, X. Wei, T. Wang, T. Lan, and S. Subramaniam, "Deadline-aware task scheduling in a tiered IoT infrastructure," in *Proceedings of IEEE Global Communications Conference (GLOBECOM),* Dec. 2017, pp. 1–7.

[103] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao, "Fog computing based efficient IoT scheme for the industry 4.0," in *Proceedings of IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM),* May 2017, pp. 1–6.

[104] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[105] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing based face identification and resolution scheme in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1910–1920, Aug. 2017.

[106] B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei. H. He, and Q. Yang, "Incorporating intelligence in fog computing for big data analysis in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2140–2150, Oct. 2017.

[107] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586–2595, Nov. 2017.

[108] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*. Springer, 2018, pp. 103–130.

[109] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[110] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[111] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[112] L. Feng, J. Yang, and H. Zhang, "RT-notification: A novel real-time notification protocol for wireless control in fog computing," *China Communications*, vol. 14, no. 11, pp. 17–28, Nov. 2017.

[113] L. Lv, J. Chen, Q. Ni, and Z. Ding, "Design of cooperative non-orthogonal multicast cognitive multiple access for 5G systems: User scheduling and performance analysis," *IEEE Transactions on Communications*, vol. 65, no. 6, pp. 2641–2656, Jun. 2017.

[114] Z. Ding, P. Fan, and H. V. Poor, "Impact of user pairing on 5G nonorthogonal multiple-access downlink transmissions," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6010–6023, Aug. 2016.

[115] L. Dai, B. Wang, Y. Yuan, S. Han, C.L. I, and Z. Wang, “Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, Sep. 2015.

[116] M. Peng, S. Yan, K. Zhang, and C. Wang, “Fog-computing-based radio access networks: Issues and challenges,” *IEEE Network*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.

[117] F. Wang, J. Xu, and Z. Ding, “Optimized multiuser computation offloading with multi-antenna NOMA,” *arXiv preprint arXiv:1707.02486*, 2017.

[118] X. Wei and M. J. Neely, “Power-aware wireless file downloading: A Lyapunov indexing approach to a constrained restless bandit problem,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2264–2277, Aug. 2016.

[119] D. P. Bertsekas, *Nonlinear Programming.* Athena scientific Belmont, 1999.

[120] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, “The impact of power allocation on cooperative non-orthogonal multiple access networks with SWIPT,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4332–4343, Jul. 2017.