COGNITIVE SPECTRUM ACCESS FOR D2D COMMUNICATIONS IN EMERGING NETWORKS

by

Xingya Liu

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2017

Approved by:

Dr. Jiang (Linda) Xie

Dr. Asis Nasipuri

Dr. Yu Wang

Dr. Haitao Zhang

ABSTRACT

XINGYA LIU. Cognitive Spectrum Access for D2D Communications in Emerging Networks. (Under the direction of DR. JIANG (LINDA) XIE)

Device to device (D2D) Communication is essential in emerging networks, such as Internet of things (IoT). Cognitive radio (CR)-enabled device is a promising technology to address D2D Communications. However, existing work on CR networks suffers from impractical protocol designs which lead CR users to long spectrum access delay (SAD) with high energy consumption (EC).

In this research, a practical adaptive framework for fast and energy-efficient spectrum access for D2D Communications is explored. The design challenges are addressed from two perspectives. First, new protocols on the medium access control (MAC) layer are established in order to deal with the unique issues in the MAC, such as the rendezvous failure due to channel status change, the handshake failure due to asynchronous time slots, the unaware deadlock due to the handshake message collision, the transmission failure due to the restless handoff, and the network congestion due to the packet congestion in each CR device. Another reason causing the long SAD with high EC is the long restless operation of the rendezvous process itself, since existing channel hopping schemes for rendezvous mainly focus on the hopping sequence design without considering the node conditions and communication environments. Therefore, cross-layer rendezvous design jointly considering channel selection, channel hopping, and transmit power control is studied.

The proposed practical adaptive framework in this research is endowed with the learning and mining abilities, which makes unlicensed users truly cognitive. Therefore, this research will provide important insights on the designs of future D2D Communications.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the direction of my advisor, advice from my dissertation committee members, assistance from my colleagues, help from my friends, and encouragement from my wife and family.

First and foremost, I wish to express my sincere gratitude to my advisor, Dr. Jiang (Linda) Xie, for giving me a good opportunity and excellent research atmosphere to work in her group at UNC-Charlotte. She has trained my technical skills, writing, and thinking skills throughout my research work during the past four years. I am very grateful for her excellent guidance, inspiration, and patience. I am very honored to be her student. Her instructions will always inspire me to continue improving myself in the future.

I would like to thank my dissertation committee members, Dr. Asis Nasipuri, Dr. Yu Wang, and Dr. Haitao Zhang, for their great help and invaluable advice along this dissertation work.

I would also like to thank my colleagues and friends: Yi Song, Ji Li, Wahida Nasrin, Md. Rajib and many others. I am so thankful and appreciative to have been able to receive help from them.

Thanks also to my parents and my parents-in-law, who have over the years supported and encouraged me with their best wishes.

Finally, I would like to thank my wife, Yiqian Li. This dissertation would not have been possible without her support, patience, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACS Available Channel Set

BER Bit Error Rate

CCC Common Control Channel

CD2DN Cognitive Device to Device Network

CH Channel Hopping

CRD Cognitive Radio Device

CRN Cognitive Radio Network

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CTS Clear to Send

CW Contention Window

D2D Device to Device

EC Energy Consumption

EHF Extremely High Frequency

ETTR Expected Time to Rendezvous

IEEE Institute of Electrical and Electronics Engineers

IoT Internet of Things

MAC Medium Access Control

MTTR Maximum Time to Rendezvous

MWB Multiple Wide Band

PU Primary User

RTS Request to Send

SAD Spectrum Access Delay

STTR Stopping Time to Rendezvous

SU Secondary User

TDMA Time Division Multiple Access

TTH Time to Handshake

TTR Time to Rendezvous

VHF Very High Frequency

CHAPTER 1: INTRODUCTION

1.1 Introduction to Cognitive D2D Communications

Device-to-device (D2D) communications enable networked devices to exchange information among each other without human control and therefore create what is known as the Internet-of-Things (IoT). The definition of IoT given by Global Standard Initiative in 2013 is *the internetworking of physical devices, vehicles, buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data.* In other words, D2D communication technique can enable things in the physical world the ability of communication, including communicating both with human and with each other. Therefore, D2D communication is essential for next generation wireless networks which aim to make the wireless nodes more like the human being during communications. D2D communications have already shown a bright future and importance as promising solutions for various modern applications such as personal asset (vehicles, pets, kids, etc.) monitoring, building management, smart homes and cities, and cognitive health systems.

However, a large number of connected devices will create challenges regarding spectrum scarcity and significant control overhead. From the forecast by Cisco, as shown in Figure 1.1, by the year 2020, the number of wirelessly connected devices will be five times than human's population. That is, devices will be the main user in the next generation network ecosystem. On the other hand, since wireless communication is based on radio propagation on certain spectrum band, here comes an inevitable problem: is the spectrum sufficient for such an increasing number of devices? According to Federal Communications Commission (FCC) [1], almost all the spectrum

has already been allocated to current wireless services. On the other hand, up to 85% of the allocated spectrum is underutilized. Studies in [2] show that even in a crowded place like a city, there are still "spectrum holes" across time and location.

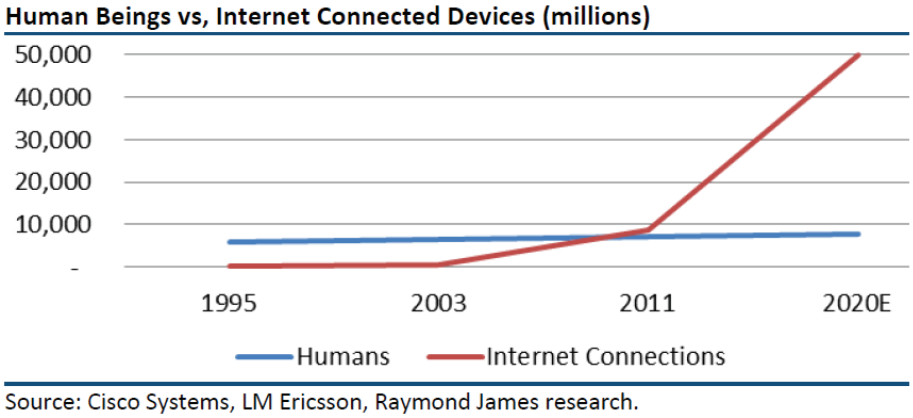


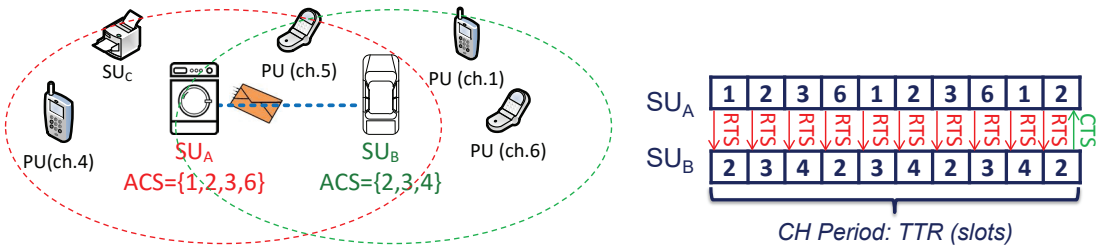
Figure 1.1: Wireless device population v.s. human population forecast by Cisco https://caseprep.wordpress.com/2014/03/03/tech-industry-insights-the-internet-of-things/.

Therefore, in order to balance the increase in the spectrum demand and the inefficiency in the spectrum usage, realizing the D2D communications falls into the solution domain of cognitive radio networks (CRNs) [3]. It is expected that devices equipped with cognitive radios can dynamically configure their operating parameters to seek "spectrum holes", or, available channels, and communicate with each other on these channels without causing harmful interference to licensed users. The devices will be secondary users (SUs) in such a network which can opportunistically access the spectrum which is temporarily unused by licensed users, or, primary users (PUs). We call this type of network cognitive D2D networks (CD2DNs).

1.2 Spectrum Access in CD2DNs

Next, we introduce the unique spectrum access process in such a network. Figure 1.2(a) shows an example of a D2D communication between a smart laundry machine and a smart car. Each device has some PUs nearby. And the channel shown in the brackets is the channel the primary user is currently using. The circles represent the

interfering range of the secondary user. Suppose there are totally 6 channels in the system. Then, the laundry machine cannot use channel 4 and channel 5, those left channels are the available channels for it. Similarly, the smart vehicle also has its unique available channel set (ACS). Moreover, they both do not know the available channels of each other. From the starting point when the laundry machine wants to establish a link to the car, till the moment they finally set up a link on some channel, is the process called the spectrum access for a cognitive D2D communication.



(a) A 6-channel CD2DN. (b) The CH process.

Figure 1.2: An illustration of the CH process in an $N = 6$ CRN.

Due to PUs' distribution and activities in the network, the ACS of a SU may change with its location and time. Thus, unlike traditional wireless (ad hoc) networks, it is impractical to assume a common control channel (CCC) in CD2DNs since it may not exist or cannot last for a long time. This requires SUs to work in a decentralized way to communicate with each other. Under such scenarios, a fundamental issue before information exchange and data transmission between a pair of SUs who want to communicate with each other is: how they can find each other on a common available channel without the knowledge of each other's available channels. This challenging process is called *blind rendezvous*, which is a unique operation in CD2DNs.

With blind rendezvous (in short, rendezvous), the spectrum access for a SU consists of the following steps: 1) rendezvous with its destination SU on a common available channel; 2) handshake with its destination SU to set up a temporary communication link; and 3) data transmission on this link. Then, the main issues impacting spectrum access delay and energy consumption, as shown in Figure 1.3, can be summarized as

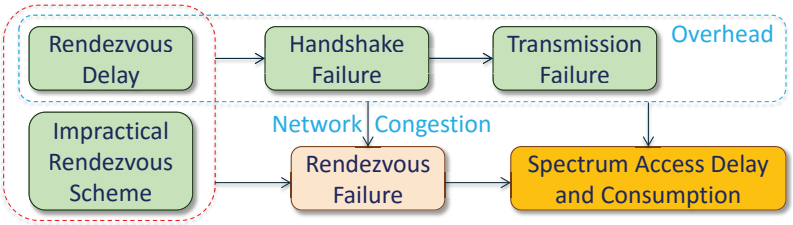


Figure 1.3: Spectrum access issues in CD2DNs.

follows:

- Rendezvous failure: in CRNs, the channel status may change during the rendezvous process, which contradicts with the SU's sensing result and is unknown to the SU. This problem usually causes unsuccessful rendezvous or long rendezvous delay, both of which delay the SU's spectrum access.

- Handshake failure: even a SU pair already rendezvous on the same channel, they may still miss each other due to the handshake failure. There are different reasons causing the handshake failure under slot-asynchronous and slot-synchronous scenarios. The fatal problem of the handshake failure is that it is unknown to SUs and will reoccur again and again whenever the SU pair rendezvous. In this way, it will generate an unaware deadlock in the network and prohibit more and more SUs from spectrum access.

- Transmission failure: after the communication link is established, the transmission may still fail due to the unavailability of the current channel, i.e., PUs reoccupy the channel used by SUs. Then, the SU pair needs to handoff to another common available channel to resume the transmission, which may lengthen the spectrum access time.

- Network congestion: all previous factors together usually result in an extraordinarily long time for each SU to set up and maintain its transmission, which is a high overhead for other SUs in the network to access the spectrum. Consequently, the

network may get congested eventually and each SU has to spend longer and longer time to access the spectrum.

- Impractical rendezvous scheme: even if all the above processes are performed successfully, the rendezvous itself may still take a long time and restless operations based on existing rendezvous schemes, especially in the worst-case situation, which causes the long spectrum access delay with high energy consumption.

1.2.1 Rendezvous Failure in CRNs

Currently, there are quite a few studies addressing the rendezvous issue in CRNs [4–21]. The technique that guides SUs to rendezvous is called channel hopping (CH). In a fully-distributed CH approach, both the source SU (who has data to send) and listening SU (who is ready for receiving potential transmission) sense all channels (N) in the network to generate their own ACS which includes the channels not used by PUs within one's transmission range, as illustrated in Figure 1.2(a).

Then, each SU divides the time evenly into time slots and hops onto these channels one by one following a predefined sequence. Figure 1.2(b) shows the CH process between $\mathrm{SU_A}$ and $\mathrm{SU_B}$ with their ACSs, where the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in IEEE 802.11 is adopted, and the hopping sequence is based on the simplest natural-order sequence. For the source SU, the sender broadcasts the request-to-send (RTS) message on each channel it hops on during each time slot until receiving the correct clear-to-send (CTS) message. For the listening SU, it keeps listening on each channel it hops on during each time slot until receiving the correct RTS.

A successful RTS and CTS exchange is called handshake. Suppose the handshake can be performed successfully. Then, two SUs can rendezvous if they hop on the same channel in the same time slot. The number of hopped slots before a source SU meets its destination SU is called time to rendezvous (TTR), e.g., $TTR = 10$ in

Figure 1.2(b).

Note that the TTR achieved by those state-of-the-art CH designs [19–21] is only the theoretical result when the rendezvous pair is within an ideal environment, i.e., the channel status in their ACS is stable. However, since the ACS varies with time and location, the previously sensed one may be out-of-date during the CH process. Hopping onto those status-changed channels may cause interference to other nodes or the rendezvous delay.

For example, suppose the rendezvous pair in Figure 1.2 is about to rendezvous on channel 3 which is suddenly reoccupied by a neighboring PU/SU of SU_A. If SU_A the sender still sends a message on this channel, it will interfere its neighboring PU/SU. Otherwise, if SU_A realizes that the channel status is changed via certain way like CSMA and keeps silent on this channel, they will miss the rendezvous in this time slot.

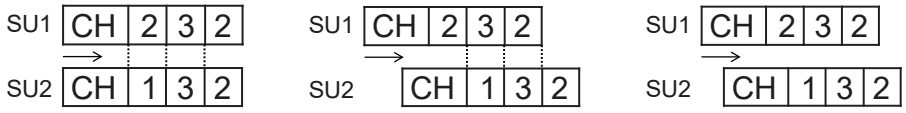
Actually, four factors can result in channel status change during CH, which are further analyzed in Chapter 3.2.1.

1.2.2 Handshake Failure in CRNs

In practical scenarios, successful hopping on the same channel does not necessarily lead to a successful handshake which can be affected by many factors. Existing CH algorithms need to work under appropriate MAC protocols to guarantee successful handshake in CRNs.

In such a MAC protocol, the key feature to support the CH is to maintain per-unit-length the same in all sequences, i.e., each SU's sojourn duration on each channel should be the same. This feature accords with the time-slotted system where the staying time on each channel can be treated as one time slot. A time slot should be long enough for two SUs to complete a handshake process. Furthermore, the time slot of each SU might need to be synchronized in order to ensure that two SUs can hop on the same channel at the same time. Although some CH papers claim that

their sequence design can work under the asynchronous scenario, it differs from the asynchronous case in this research. For example, Figure 1.4(a) shows a synchronous CH case where SU_1 and SU_2 start to hop at the same time; Figure 1.4(b) shows the asynchronous CH case where two SUs start to hop at different time slots; and in Figure 1.4(c), the slots are unsynchronized. We name the latter two cases as user-asynchronous case and slot-asynchronous case, respectively. In this research, the asynchronous scenario only means the slot-asynchronous case.



(a) synchronous (b) user-asynchronous (c) slot-asynchronous

Figure 1.4: Synchronous and asynchronous scenarios in this research.

In synchronous scenarios, analytical models [11,22] show that existing CH schemes perform well when the role of each SU is predefined: for a given time, a source/destination SU will always be a source/destination SU, or, SUs are always in the same pairs and the pairing information is known. However, in a more practical scenario, the role of a SU is not fixed: a source SU may be a destination SU of other SUs, and a destination SU may be a source SU at the same time if it has data to send. In addition, a SU's pairing SU may change with time. One main problem under such cases is the *rendezvous deadlock*. As illustrated in Figure 1.5, SU_1 can rendezvous with its destination user SU_2 on channel 2 if SU_2 acts as an idle listening user. However, when SU_2 is also a source SU to another SU, it cannot hear from SU_1 even on their rendezvous channel. We call this case the *both-shouting* scenario. Among all both-shouting cases, the worst one is that SU_2's destination user is SU_1 coincidentally, which is a rendezvous deadlock. What is worse, the number of SUs in a deadlock can be more than two in CRNs.

The rendezvous deadlock is a **unique** phenomenon in CRNs without CCC. In existing multi-channel MACs [23–25] for CRNs, the status/role of other users can be informed by the CCC. Thus, deadlock will not happen with the existence of a CCC.

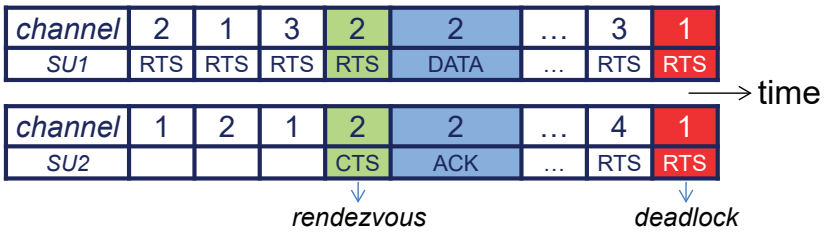


Figure 1.5: The deadlock during blind rendezvous without role pre-assignment.

Meanwhile, in single-channel networks, since all users operate on the same channel, one can infer the RTS collision if no corresponding CTS is received. Hence, a deadlock can be easily detected and will not last for a long time. However, in CRNs, before rendezvous, a SU is ignorant of other SUs' role. As a result, there is always a chance to generate the deadlock. On the other hand, since the pair of deadlock users are blind of each other's current operating channel, the deadlock cannot be noticed by the suffering SUs. For example, in Figure 1.5, after SU_1 sends an RTS message on one channel, the expected CTS may not show up due to several reasons. One is the absence of the destination SU on the current channel, such as in SU_1's first three time slots. Another reason is the both-shouting issue, as what happens in the last time slot in Figure 1.5. Therefore, even if SU_1 and SU_2 are a deadlock pair, they have no clear evidence to confirm the situation and have to keep CH.

In addition, the rendezvous deadlock cannot be well solved by existing MAC mechanisms. We cannot directly apply the existing CSMA/CA mechanism in IEEE 802.11, which theoretically requires the time slot for each hopping channel to be infinite long due to its retransmission mechanism.

Moreover, the length of a time slot is a tradeoff parameter in both scenarios. When a time slot is long, the probability of having a successful handshake is high, but it takes a long time for two SUs to hop on the same channel. On the other hand, if a time slot is short, the handshake process cannot be guaranteed, which may cost more time to get a successful rendezvous.

1.2.3 Transmission Failure in CRNs

In the traditional wireless MAC, a long packet is usually fragmented into small frames in order to achieve a satisfactory bit-error-rate (BER) for the PHY layer transmission in a dedicated channel. However, compared with BER, the main factor contributing to the retransmission in CRNs is the PU's interference, since a SU can always find a channel with a relatively better quality (low BER) but cannot avoid PUs' reoccurrence. Meanwhile, the retransmission delay is also different from the traditional one without spectrum handoffs, which affects the overhead analysis of the frame size design. Thus, packet fragmentation in CRNs should be specially designed, which, however, is less investigated.

Moreover, the desired protocol should satisfy the following three design guidelines for practical packet fragmentation simultaneously. First, as mentioned previously, the protocol should capture the unique features of the new CR functions. Currently, the ability of multi-channel handoffs is only considered in [26]. Second, it is impractical to derive a universally optimal frame size and implement it to all packets with different lengths. For example, consider a packet which is 1.5 times longer than the optimal frame size. Is it still optimal to fragment this packet into two frames and only one of them has the optimal size? Only [27] mentioned that each given packet may have its own optimal frame size to fragment. Last, due to the time-varying mobility and activity of both PUs and SUs in CRNs, even for the same packet size, the optimal fragmentation changes with the network environment. Only [28] aims to dynamically acquire the optimal packet size.

1.2.4 Network Congestion in CRNs

Some rendezvous-guaranteed CH schemes ensure that, by following their CH sequences, any two SUs with at least one common available channel can rendezvous with each other within a finite number of slots called maximum TTR (MTTR).

Without a proper MAC framework for CRNs where traditional MAC cannot be used due to the unique features mentioned above, directly applying existing CH schemes into CRNs will further influence the secondary network in three ways, progressing in sequence:

1. If a SU uses the theoretical MTTR as the length of its spectrum access window, the rendezvous delay and handshake failure explained above may cause the access failed when the total delay is longer than the window size. Obviously, a longer MTTR is needed for a satisfactory spectrum access rate in CRNs.

2. A longer MTTR results in a longer Expected TTR (ETTR). However, since the ACS of a SU varies from time to time due to PU activities and its location, CH has to be performed before every transmission session in the secondary network. Based on queuing theory [29], a long ETTR then increases the service time for each session and increases the congestion probability. Thus, the throughput of a SU may be decreased due to the congestion especially when the source SU has a high traffic rate.

3. Furthermore, a SU with a long ETTR spends more time on its sender mode while encumbers other SUs who want to communicate with it. Since SUs are blind of each other's status, other SUs have to spend more time on CH. By parity of reasoning, more and more SUs' ETTR will be increased.

In this way, the spectrum access delay of an individual will be amplified by the blind features to the entire secondary network.

1.2.5 Impractical Rendezvous Scheme in CRNs

The drawbacks of current CH-based rendezvous algorithms are threefold. Firstly, these algorithms raise an awkward scenario: their TTR increases with the number of available channels. If the total number of channels in the network is N, the state-of-the-art CH design [19,20,30,31] can achieve ETTR and MTTR on the order of $O(N)$

and $O(N^2)$, respectively. This contradicts the original intention of cognitive radios which should perform better when there are ample unoccupied channels in primary networks. A method to downsize the large ACS of SUs during the rendezvous process is proposed in [32], but the TTR is still proportional to the new size of ACS. Only in [11], instead of N, the TTR is proportional to the number of SUs, but every SU is preassigned a role (source SU or listening SU) in each slot. In addition, neither the design proposed in [32] nor [11] is rendezvous-guaranteed.

Secondly, despite the relationship with N, the TTR itself in existing effort is not short enough on networking operations at higher layers. To get an acceptable successful handshake rate, it is proved in Chapter 3 that the staying time on each hopping channel (i.e., one time slot) should be at least $4 \sim 6$ times longer than the $(RTS+CTS)$ transmission duration, $t_{RTS+CTS}$. Consequently, the actual rendezvous time is more than $(4 \times t_{RTS+CTS} \times TTR)$. Even if the fastest CH algorithm is adopted, this long rendezvous time can easily result in network congestion even under moderate traffic conditions as explained in Chapter 3.

Last but not the least, in existing CH schemes, the energy consumption of a SU is high since it keeps hopping from one channel to another. Even if a SU does not have any packet to send, in order to provide the communication chance for potential source SUs, it still needs to keep hopping. Most existing rendezvous schemes ask every SU to follow the same CH algorithm no matter it is an active source node or a passive listening node. In [32], different CH algorithms are designed for the source SU and the listening SU, but the latter still has to hop on different available channels frequently. From this point of view, there is no "idle" user in a secondary network when using current CH methods. The restless operation makes the CR technique unsuitable to nowadays smart devices, which usually have restricted battery life.

To eliminate the above CH problems, we consider shortening the TTR from the perspective of constructing desirable ACS for the rendezvous-pair. We propose that

the ACS of a listening SU should be a subset of a source SU's ACS. We use the following simple CH strategy to explain the salient feature of such rendezvous pair no matter how the subset relationship of their ACS is formed. Assume that the listening SU keeps staying on an arbitrary channel in its ACS. The source SU keeps hopping on to every channel one by one in its ACS. Due to the subset relationship, the source SU can always meet the listening SU on one of its available channels eventually and thus rendezvous is guaranteed. As we can see, even in such a rough scheme, the operation of the listening SU is minimized since it only needs to keep listening on one channel and the order of MTTR is $O(N)$, which is already a breakthrough achievement in CH design with respect to $O(N^2)$.

Motivated by the above feature, we investigate the issue of available channel selection in order to realize such ACS pair. In fact, as explained in [33], the concept of available channels is imprecisely mentioned in many CH-based rendezvous papers as the channels that are not occupied by PUs. In this research, we regard an available channel as the channel that can be used without generating unacceptable interference to PUs. By this definition, whether a channel is available or not for a SU depends on the SU's *interfering range* and the locations of PUs on the same channel. Since the interfering range is variable under different transmission power of a SU [34] and the locations of PUs can be inferred from the sensing period (see details in Chapter 4), the number of available channels for a SU is controllable using appropriate transmission power.

However, it is still very challenging to form the ACS subset relationship for a pair of rendezvous SUs in practical CRNs. For example, one challenge exists in the impact of the one-hop distance. In order to include the listening SU's ACS, the source SU should limit its transmission power low enough to interfere less number of PUs and thus make more channels available. However, this may be a problem for the source SU's transmission to reach its destination SU when their distance is large. The

protocol called SUBSET addressing all these issues is proposed in Chapter 4.

1.3 Overview of the Proposed Research

Figure 1.6 shows the overview of the proposed research. The research objective is to design a framework for fast, energy-efficient, collision avoidance, contention free, and high data rate D2D communications for emerging networks. As introduced previously, the MAC-layer failures will cause severe problems during spectrum access, which need to be addressed at the very beginning. Therefore, corresponding MAC-layer protocols are proposed in sequence to address or at least mitigate the impact of these practical issues on the spectrum access in CD2DNs. This part makes the idea of CD2DNs applicable. Next, in order to make it more attractive, PHY layer protocols and CH algorithms were further designed. The integration of these two can largely improve the network's performance. Furthermore, with such a framework, particular functions are designed to fulfill the potential applications of CD2DNs, including a cross-layer fast neighbor discovery protocol, a 2D heterogeneous rendezvous protocol for D2D communications in multi-wideband (MWB) scenarios, and a priority-based spectrum access scheme for CD2DNs in IoT/5G scenarios.

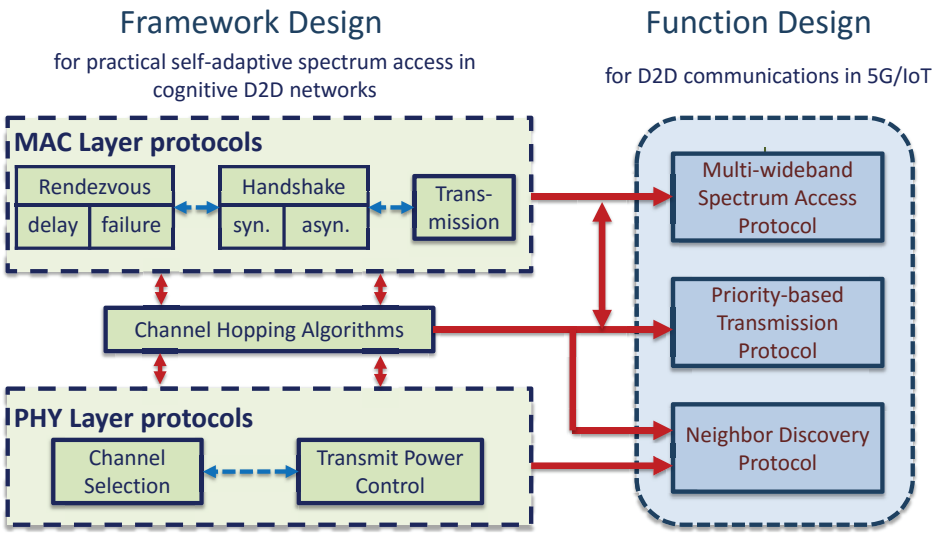


Figure 1.6: The overview of the proposed research.

For the framework design, fist of all, we propose a rendezvous control protocol to

deal with the rendezvous delay and network congestion. The factors that influence the channel status change are first studied in detail. Then, analytical models are established for deriving the relationship between the rendezvous performance and the system parameters. Based on the analysis, we develop an efficient reasoning-reaction mechanism to quickly discover the changed channel and compensate the rendezvous delay timely. Furthermore, we name the rendezvous window size as stopping TTR (STTR), after which a source SU gives up the sender status and reports a rendezvous failure to higher layers. The optimal STTR in terms of minimizing the spectrum access delay of the whole network can be dynamically derived and performed in each distributed SU.

Then, two protocols to increase the successful handshake rate are proposed. For the asynchronous scenario, we propose a novel RTS/CTS handshake mechanism to mitigate the effects caused by asynchronous time slots. This mechanism can achieve successful handshake with a high probability when the sender and the receiver arrive on the same channel at different moments due to their asynchronous time slots. In addition, we also derive the optimal length of a time slot for our design.

For the synchronous scenarios, the CSMA/CA mechanism is revised to adapt to the nature of CH schemes, which can be integrated into the time slotted system. The contention window (CW) concept is kept but adding some new mechanisms to address deadlock scenarios. In the proposed design, the size of the CW is fixed in order to maintain the equal length of each time slot. A novel probabilistic model is established for deriving the optimal CW size in terms of maximizing the network throughput.

Furthermore, a packet fragmentation protocol is proposed to control the total transmission time with several transmission failures. We first mathematically model the retransmission probability of a frame as a function of the given length. Our model takes into account related CR operations such as spectrum sensing, SU-contention,

and spectrum handoff. The derivation employs parameters that a SU can either know or learn by itself as much as possible. Then, based on the average retransmission rate counted for different packet lengths in a short-term transmission history, the unknown (PU-related) parameters can be estimated using our model. In addition, the optimal frame size in terms of maximizing the throughput is derived with both known and estimated parameters. Finally, we propose a self-adaptive optimal fragmentation (SAOF) protocol for SUs in CRNs. With SAOF, each SU can individually derive the optimal frame size for a given packet based on its latest environment.

Last but not the least, a cross-layer rendezvous protocol is proposed to accelerate the rendezvous speed from the fundamental level. First, the relationships of several important parameters are analyzed such as the transmission power of a SU, the *interfering range* of a SU transmitter, the distance between the rendezvous pair, and the *interference range* of a SU receiver (a different concept from the interfering range of the source SU. See details in Chapter 4). These relationships exist in all CRNs but are not well utilized by other CH schemes. Besides, practical assumptions and worst-case scenarios are also considered including the sensing resolution of SUs, role-exchange problem, the interference among SUs, and PU location derivation. Then, based on these relationships and practical constraints, we propose a mechanism to construct the ACS of any rendezvous SU pair. Moreover, two highly efficient CH algorithms are developed based on such ACS pairs to deal with different one-hop distances. Finally, we propose analytical and application models to evaluate the TTR of the proposed protocol.

For the function design, besides the above cross-layer rendezvous protocol which can perfectly fulfill the neighbor discovery function, we further propose two basic yet less investigated functions for CD2DNs in different scenarios. The first one is the heterogeneous rendezvous design in MWB scenarios. Ideally, CR users are capable of sensing and exploiting any potential transmission opportunities in the available

spectrum band ranging from 30 KHz to 300 GHz. With the multiple-diverse-band spectrum, the network can provide more radio resources and capacity to a large number of CR users. However, the multiband scenario (e.g., TV band + 2/3G band + 4/5G band) also introduces significant challenges in channel rendezvous. Existing studies on channel rendezvous suffer from unacceptable long delay and high energy consumption when applied to such scenarios. In this research, we propose a two-dimensional heterogeneous rendezvous (2D-HR) protocol which can support MWB-CD2DNs with a significantly reduced rendezvous delay and energy consumption for various rendezvous scenarios, such as the pair-wise rendezvous, any-wise rendezvous, and multi-wise rendezvous. The proposed design also performs better than existing efforts even when dealing with traditional single-band rendezvous.

The other function is to achieve priority-based spectrum access in IoT/5G scenarios since D2D communication is of the essential in IoT/5G. However, the unique CH process may impact or even prevent general communication services in IoT/5G. Among all these functions, priority-based spectrum access is less investigated yet urgently desired in IoT/5G. Unfortunately, existing CH methods cannot help CR users establish such capacity due to its shortcomings. In this research, we propose PCH, a priority-based spectrum access protocol, which can be integrated with any existing CH algorithm. PCH can support priority transmissions with a significantly reduced CH delay compared with nonpriority transmissions. More importantly, PCH can work under practical scenarios such as the D2D both with priority packets, the D2D handoff, and the overhead/energy constraint IoT.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, related work on spectrum access solutions and MAC designs in CRNs are surveyed. In Chapter 3, four MAC-layer protocols to address the practical issues in the CD2DN spectrum access are presented. In Chapter 4, a cross-layer rendezvous protocol for fast neighbor discovery

in CD2DNs is proposed. In Chapter 5, a 2D heterogeneous rendezvous design is proposed in multi-wideband CD2DNs. In Chapter 6, a priority-based spectrum access scheme is proposed in CD2DNs. In Chapter 7, this dissertation is concluded and future works are considered.

CHAPTER 2: RELATED WORK

Although the spectrum access issue has been extensively investigated in both traditional single channel and multi-channel networks, there are unique challenges that are unexplored in the spectrum access design in CD2DNs. Unfortunately, the direct solutions to address these issues considered above are ignored in existing works.

2.1 Existing Rendezvous Protocols in CRNs

Many existing CH schemes are based on impractical assumptions: *(i)* the symmetric assumption [4, 9, 13], which requires that all SUs have the same available channel set; *(ii)* the time-synchronous assumption [5, 7, 11], which asks all SUs to start hopping at the same global time; *(iii)* the role preassignment assumption [11], i.e., every node pair is pre-assigned a role as either a sender or a receiver; and *(iv)* each occupied channel is assumed to be used by one PU. In other words, each channel's availability is associated with one PU's availability. This assumption has been widely adopted by probabilistic-based CH schemes [6, 10] since the activity of PUs usually shows similar patterns along the time. However, in realistic PU networks like 2G/3G, each PU may be randomly assigned a channel on each transmission or several PUs may share one channel when using mechanisms like time division multiple access (TDMA). Therefore, the available channel sets of SUs are unpredictable and time-varying.

2.2 Existing Handshake MAC Protocols in CRNs

Not all MAC protocols in CRNs can be used for blind rendezvous, such as the protocols under the single channel [35] and the overlay model [36, 37]. Other designs [38, 39] either directly use the synchronized time slot assumption [40–43] or achieve synchronization by impractical methods, including employing a CCC [44–48], using

multiple transceivers [6, 7, 43, 44, 48], or broadcasting beacons on all channels before rendezvous [49], which is less efficient due to high overhead and collision. The protocol proposed in [50] is claimed to be robust enough under asynchronous slots. However, it does not fully consider the potential problems in such a scenario and lacks details. In summary, all the existing MAC designs consider time synchronization as a necessary condition for blind rendezvous and achieve it in different impractical ways.

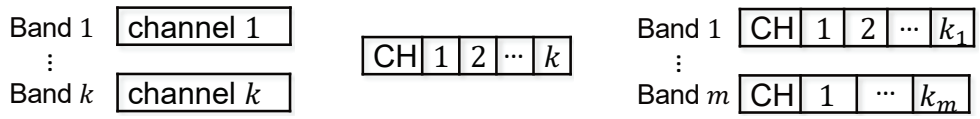
2.3 Existing Packet Fragmentation Protocols in CRNs

There are several remotely related papers studying the optimal packet size in different CR networks with various limitations: i) the optimal SU packet size from the perspective of system design is designed in [26, 51], which is not appropriate to be embedded in a distributed SU MAC layer due to the lack of ample system information; ii) the optimal packet size for physical layer in terms of energy saving and BER control is considered in [28, 52, 53], but they neither fully adapt with the CR environment (the underlay mode in [53]) nor establish practical channel models (a common control channel in [28] which are difficult to maintain in CRAHNs and overly simplified assumption in [28, 52] where each channel is only associated with one dedicated PU) ; and iii) the optimal frame size for packet fragmentation is proposed in [27, 54, 55], but also modeled as the underlay mode in a single channel network.

2.4 Related Work on Heterogeneous Rendezvous in MWB Scenarios

Ideally, multiple frequency bands promise significant enhancements to CR networks (CRNs)' throughput by providing more radio resources and capacity to a large number of CR users [56, 57]. Quite a few research papers have addressed various issues in multiband CRNs, including spectrum sensing [57, 58], power allocation [59, 60], and multiband antenna design [61]. However, another basic yet important operation, *channel rendezvous*, is less investigated in MWB-CRNs. Particularly, this rendezvous process faces significant challenges in MWB-CRNs on the following three aspects.

First of all, the MWB-CRN considered in this research is a more realistic multi-band network where PUs may be on different spectrum bands with different service providers. For example, SUs trying to utilize the spectrum band near 800 MHz need to access the channels defined by the 2G service providers in order to co-exist with their PUs. Hence, *the system model in our considered MWB-CRNs is totally different from that in existing studies.* As shown in Figure 2.1(a), the MWB-CRNs considered in some existing papers refer to multiple narrowband channels [57] where k spectrum bands are represented by k channels. On the other hand, related research on wide-band CRNs deals with multiple channels in a single wideband [58], as shown in Figure 2.1(b). However, since SUs on each band should follow the channels defined in each particular band, the system model in our considered CRNs is like a combination of the previous two scenarios, as shown in Figure 2.1(c): from the very high frequency (VHF) band to the extremely high frequency (EHF) band, the number and band-width of channels may be different in each corresponding primary network. Note that so far there is no corresponding channel rendezvous effort for MWB-CRNs.



(a) multiband (b) wideband (c) multi-wideband

Figure 2.1: The system models in different CRNs.

Unfortunately, it is not feasible to apply existing rendezvous approaches to MWB-CRNs without significant performance degradation due to the unacceptable rendezvous delay and high energy consumption. The current state-of-the-art rendezvous designs [19–21] can guarantee rendezvous in a single band with the expected rendezvous delay and the worst-case rendezvous delay on the order of $O(N)$ and $O(N^2)$, respectively, where N is the total number of channels in the band. Apparently, these designs assume that there are only a small number of channels (at most tens of channels) in the network and SUs will utilize all the channels to design the rendezvous

algorithm. These algorithms have several drawbacks when applied to MWB-CRNs: 1) SUs have to consider all the channels in every spectrum band together as in a single wideband, which will suffer from $O(MN)$ expected rendezvous delay and $O(M^2N^2)$ worst-case rendezvous delay, where M is the number of the spectrum bands; 2) according to existing rendezvous algorithms, SUs have to keep hopping among all these channels to achieve a possible rendezvous, which makes the energy consumption for this process also significantly increased in the MWB scenario; and 3) most existing rendezvous designs ask every SU to follow the same algorithm no matter it is an active source node or a passive listening node. Even if a SU does not have any packet to send, in order to provide the communication opportunity for other potential source SUs, it still needs to keep hopping from one channel to another. From this point of view, there is no idle time for a SU. The restless operation especially in the MWB scenario is unsuitable to the smart devices nowadays which have a limited battery life.

Other existing rendezvous schemes without the above drawbacks have other limitations as well. In [11], instead of N, the rendezvous delay is proportional to the number of SUs, but the rendezvous cannot be guaranteed. The same shortcoming applies to the design proposed in [32], where different algorithms are designed for the source SU and the listening SU. In [62], a fast wideband rendezvous design is proposed under which the delay is impacted by the number of channels and SUs. However, it cannot work in MWB-CRNs since its proposed channel grouping method cannot be reproduced in multiple bands.

Last but not the least, in MWB-CRNs with a large number of channels and users, there should be more than one type of rendezvous to better serve the SUs with different status in different network phases. Motivated by this, we define three types of rendezvous in MWB-CRNs: 1) pair-wise rendezvous: the most common rendezvous scenario where a source SU tries to rendezvous with a particular destination SU for

communications. Actually, almost all rendezvous designs in existing efforts refer to the pair-wise rendezvous; 2) any-wise rendezvous: after the network setup phase, a newly joined SU or a re-activated SU with no updated information about the network needs to rendezvous with any other SU in the network to obtain the latest network control information. This type of rendezvous has no specific destination SU, but has to be done as soon as possible for the sake of low overhead. A similar definition of rendezvous is mentioned in [18, 62], but the solution is not suitable for large-scale CRNs; and 3) multi-wise rendezvous: a source SU may have different data packets for multiple destination SUs at the same time. The source SU has to rendezvous with each destination SU one by one sequentially using the traditional rendezvous method causing long total rendezvous delay, not to mention the largely increased rendezvous delay for each destination SU, as explained before. Therefore, a multi-wise rendezvous should be designed to be able to rendezvous with multiple destination SUs faster. This type of rendezvous is never investigated in CRNs. Such heterogeneous rendezvous (pair-wise, any-wise, and multi-wise) is highly desirable in MWB-CRNs, but also complicates the design especially the any-wise rendezvous and multi-wise rendezvous cases.

2.5 Related Work on Priority-based Spectrum Access for IoT/5G

Although CH enables the spectrum access for CD2DNs, the obligated CH process prevents a potential function, *priority-based spectrum access*. Though there is hardly any existing work focusing on the priority-based communication in such networks, it is highly desired in IoT/5G. As the fast growth of IoT applications [63], CR devices (CRD) may deal with different types of data or data with different time constraints. Some data type such as emails has a higher delay tolerance than the real-time video stream. On the other hand, data of an urgent event need to have a higher priority to send through the network. However, all these high priority packets may lose their priority in the CD2DN due to the lack of the control channel.

To be specific, consider two CRDs with the same destination CRD. The one with higher priority data cannot connect to the destination device quicker by existing CH methods. In order to differentiate the CH delay for packets with different priority, one possible way is to use different CH algorithm to serve different packet, i.e., the faster algorithm for the higher priority packet. However, only both the source CRD and the destination CRD use the same algorithm, can the CH process lead to the rendezvous. Unfortunately, the recipient in such networks does not know the priority of a packet before CH. Then, the algorithms used by the D2D pair may be different and thus even cannot guarantee the rendezvous. Another way is to design some downgraded versions of a CH algorithm for low priority packets. In other words, the purpose of distinguishing is achieved by the cost of manually slowing down regular D2D connections.

Besides, the D2D pair may need further action after rendezvous to satisfy with the priority-based spectrum access. Without loss of generality, the D2D pair prefers to transmit data with higher priority on the channel with faster speed, especially in upcoming networks where users can access several different bands [64]. However, in existing CH efforts, maximizing rendezvous channel diversity [5, 30] is one of the primary design goals. In most cases, the rendezvous channel is not the desired fast channel. Then, the D2D pair needs to handoff to the corresponding channel in order to transmit priority data, which adds additional handoff delay.

Moreover, some practical scenario and constraint also need to be considered for the design. For example, if any priority-based method is adopted in the network, for the D2D pair where both are source devices with priority packets to send, can they still connect with each other? And furthermore, rendezvous no later than the case when one of the destination SU with non-priority packet? Also, will the design take additional actions with extra overhead and energy consumption? Will this fact influence the overall performance?

CHAPTER 3: PROPOSED PRACTICAL CD2DN-MAC PROTOCOLS

In this chapter, we first analyze all practical issues leading to MAC layer failures during spectrum access in CD2DNs. Next, we propose corresponding MAC protocols to address these issues or mitigate their impact on the spectrum access process.

3.1 System Model

Network Environment: the system considered in this chapter consists of finite number of PUs and SUs which can operate over a set of orthogonal channels denoted by $\boldsymbol{C} = \{c_1, c_2, c_3, ..., c_N\}$ where N is the number of total channels in the network. Their locations are randomly chosen but are able to maintain the network connectivity, i.e., each node is within at least one of other nodes's transmission range.

Primary networks: each time when a PU has a packet to transmit, a channel c_i is randomly selected from the system. Note that one channel can be assigned to multiple PUs simultaneously, which accords with realistic communication systems.

Secondary networks: each time when a new packet is generated by a SU, it becomes a *source SU* and randomly chooses a *neighboring SU* (within its transmission range) as its *destination SU*. Assume that each SU is equipped with one cognitive radio working in the half-duplex mode due to the cost limit. Consequently, the destination SU should be either an *listening SU* who is in the receiver mode or a *busy SU* who is in the sender mode.

Communication Steps: since the scanning time of the whole band ($\boldsymbol{C}$) is much longer than the time slot in CH process (i.e., it has been proposed in the IEEE 802.22 standard [65] that in the order of milliseconds per channel should be spent by PHY-layer sensing so as to achieve the desirable level of PU-detection quality, while the

transmission time of an RTS and a CTS is usually far less than 1 ms), each SU usually only performs spectrum sensing once to get the ACS at the beginning of the CH, and then devoutly follows the predefined CH sequence before the rendezvous success. The scanning is done periodically to keep ACS up-to-date to avoid channel status change due to PU activities.

After a SU pair successfully rendezvous, they enter the data transmission period. However, if somehow they fail to rendezvous before STTR, the source SU has to backoff or drop the current packet and start a new CH period to serve other packets left in the queue. If no new traffic generates, it turns to a listening SU. In this chapter, we mainly focus on the CH period since the operation for protecting data transmission is well-studied [66, 67].

Performance Metrics: for the part of rendezvous design, although ETTR and MTTR are regard as two crucial metrics in existing CH efforts, they are calculated in ideal CRNs where interference during the CH is not considered and the role of the rendezvous pair is preassigned. However, in CRNs, the MTTR may not exist since the rendezvous cannot be guaranteed due to many practical factors analyzed later. Then, ETTR lost its meaning as a measurement of the rendezvous speed. Therefore, we slightly revise the definition of the fundamental metric, TTR, in CRNs: instead of time to rendezvous, it represents *the number of time slots a source SU spends in CH period*. For example, the TTR for SU_A in Figure 1.2b is 10. However, if SU_A fails to rendezvous within STTR, then the TTR for SU_A is STTR.

In this way, MTTR actually means STTR and ETTR indicates the average time a source SU spent in CH period. Meanwhile, compared with above metrics, the normalized throughput, becomes a more important metric which can reflect the the average rendezvous successful rate in CRNs. Thus, the main performance metrics considered in this chapter are $ETTR$ and the normalized throughput.

For the part of handshake design, only after a successful handshake, can two SUs

truly establish data communications. Thus, we define time to handshake (TTH) as our performance metric in this issue. Meanwhile, since the rendezvous delay and network congestion caused by the deadlock will be eventually reflected on the network throughput, we define the normalized throughput of the network as our performance metric in this issue.

3.2 The Proposed Rendezvous-control MAC Protocol

In this section, we analyze and evaluate each possible factor which may affect the rendezvous performance, including practical factors inherent with the CRN environment and the drawbacks of existing CH schemes. Then, based on these analysis, we propose a practical self-adaptive rendezvous framework guiding current CH algorithms to gain maximum throughput in CRNs by dynamically learning and changing corresponding parameters. Simulation results validate our analysis and demonstrate the merits of PSA against above issues for different CH algorithms under various scenarios.

3.2.1 Factor Analysis for Rendezvous Delay

Consider the same example in Figure 1.2a, the ideal rendezvous process is already illustrated in Figure 1.2b. However, the real scenario for the rendezvous pair in CRNs could be more like what included in Figure 3.1. There are mainly two reasons that may result in a rendezvous delay or failure which are not considered before.

First, the channel status in the ACS changes before its updating. For instance, if c_2 in Figure 3.1 becomes unavailable during the broadcast while not known by $\mathrm{SU_A}$, the two SUs have to keep hopping and rendezvous on c_3 in the next time slot. In fact, there are mainly four factors which can affect the status of a channel during the CH.

1. A channel will become unavailable to a SU if a neighboring PU re-occupies the channel during its rendezvous. For example, the PU in Figure 3.1 reuses c_6 after $\mathrm{SU_A}$'s CH. $\mathrm{SU_A}$ has no way to know this information since at that time

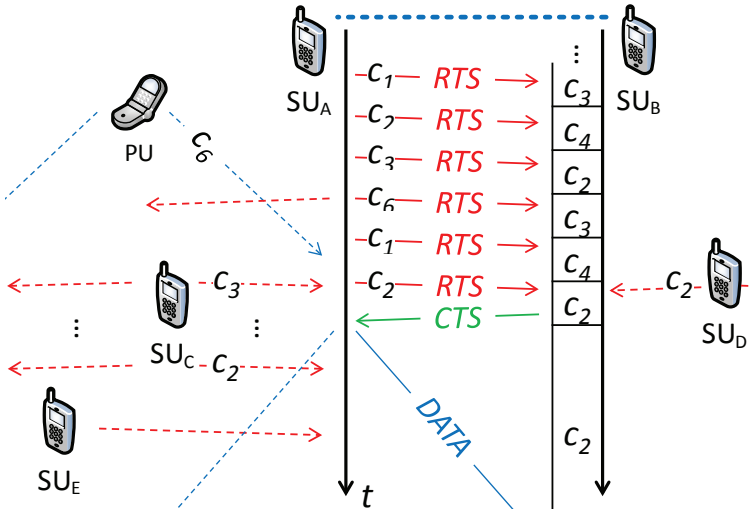


Figure 3.1: An illustration of the rendezvous in a 6-channel CRN.

it is on another channel (c_2) with sender mode. When SU_A hops on to c_6: i) if it directly sends the RTS and the PU is working in the duplex mode, the message will interfere PU's receiving; ii) if the PU is working in sender mode, SU_A cannot receive the potential CTS by the interference; and iii) if CSMA is adopted in each time slot, SU_A has to keep silent on this channel and thus a rendezvous may be missed if SU_B is also on this channel. Also, if the PU is a neighbor of SU_B, the rendezvous has no chance to happen on this channel.

2. A channel can becomes unavailable to a SU if its neighboring SU rendezvous on this channel ahead of it. Consider SU_C in Figure 3.1, when it turns to a source SU, c_2 is still available to it. However, SU_A rendezvous with SU_B on c_2 during SU_C's hopping and keeps transmitting data on this channel. When SU_C hops on to c_2, the channel is already unavailable. No matter CSMA is adopted or not, SU_C can either not send RTS for its potential destination SU on this channel or be interfered by SU_A when receiving potential CTS.

3. The RTS collision at the receiver side is an unavoidable issue which also exists in CRNs. As SU_D in Figure 3.1 sends its RTS on c_2, whoever its destination SU is, the RTS colliding with the one from SU_A prohibits the potential rendezvous

for SU_A and SU_B in this time slot. Besides, in CRNs, if SU_D is a neighbor of SU_A and CSMA is adopted, one of the them has to give up the broadcast in this slot.

4. A channel will become unavailable to a SU during the CH if the SU keeps moving and encounters new neighboring nodes (PUs and SUs) which are currently using an available channel of the SU. For instance, if SU_E moving towards the PU and SU_A during its CH, c_6 and c_2 will become unavailable to it if they are in SU_E's ACS previously.

Another reason is called the invisible-role problem. The role of the destination SU in ideal CRNs is assumed to be an listening SU. However, as we mentioned in the handshake problem, it may also be on a sender mode. Suppose the destination SU of SU_C and SU_E in Figure 3.1 is just SU_A. They have no idea whether SU_A is in the sender mode (CH or data transmission) or not unless they heard the CTS to SU_A, which, requires them to show on the same channel in the same time slot while within SU_B's transmission range. Hence, *the role of their destination SU is invisible to them.* In this case, they can only keep CH and count on that SU_A is idle or become idle soon before their STTR expired. Most rendezvous failure is attributed to this reason.

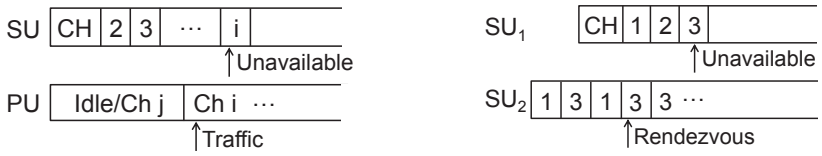
Table 3.1: Notations used in the rendezvous control protocol

M	The number of available channels of a SU
K_P/K_S	The number of neighboring PUs/SUs of a SU
$T_R/T_E/T_S$	TTR/ETTR/STTR in the unit of seconds
T_P	The time for one data packet transmission
P_R	The rendezvous probability within STTR
λ_P/λ_S	Average packet arrival rate of PUs/SUs

Next, we analyze the features of these factors in order to design the practical rendezvous framework. Except the parameters introduced in the network model, other parameters will be used in our analysis are listed in Table 3.1. Note that the

values of the parameters used in the examples in the following analysis are referred from IEEE 802.11-based protocols [68–71].

Neighboring PU Return: a channel will become unavailable to a SU if a neighboring PU re-occupies the channel during rendezvous, as illustrated in Figure 3.2a.



(a) The PU's reoccurrence (b) The SU's transmission

Figure 3.2: Packet-level factors affecting the channel status during the CH.

We model each PU as a general $G/G/1$ system. The probability that a PU is busy is $\rho_P = \lambda_P T_P$ by *Little's Law* [29]. Let $N(t)$ be the number of data packet arrivals of the PU in time t. Then, the probability of n packet arrivals in time t is $Pr[N(t) = n]$, which is associated with the distribution of packets arrivals. For example, if the traffic of this PU follows Poisson distribution, $Pr[N(t) = n] = \frac{(\lambda_P t)^n}{n!} e^{-\lambda_P t}$.

First, we consider the one-PU-return scenario, which means that only one neighboring PU reuses a channel in the ACS during CH. This scenario happens in two cases:

1. The PU is idle $(1 - \rho_P)$ at the beginning of the source SU's broadcast. Then it starts traffic during the CH $(1 - Pr[N(T_E) = 0])$. In addition, the new channel assigned to the PU (c_i) is a channel in the SU's ACS and going to be used by the SU in the remaining CH time. Since the return may happen at any time during the SU's CH period with equal probability, the expected returning moment is at the half of the SU's TTR. Thus, the probability that c_i is reused by the PU is $\binom{N}{1}/\binom{M/2}{1}$.

2. The PU is busy (ρ_P) at the beginning of the broadcast (c_j), but it finishes its transmission within the SU's CH period $(\frac{T_E}{T_P})$ and immediately starts another

transmission since it still has packets in the queue waiting for the service (ρ_P). The new selected channel should also satisfy the requirement in case 1.

Hence, the occurrence probability of one-PU-return within a CH period is

$$Pr[one\ PU\ return] = (1-\rho_P)\left(1 - Pr[N(T_E)=0]\right)\frac{M}{2N} + \rho_P \frac{T_E}{T_P}\rho_P \frac{M}{2N}. \quad (3.1)$$

Then, in the multiple-neighboring-PU scenario, the occurrence probability of PUs' return can be derived as

$$Pr[PUs'\ return] = 1 - \left(1 - Pr[one\ PU\ return]\right)^{K_P}. \quad (3.2)$$

The correctness of the probabilistic model is validated through the numerical results shown in Figure 3.3 where Random CH is adopted. From the left figure, when in a moderate CRN, the occurrence probability of this factor is less than 3%. However, as the number and traffic rate of neighboring PUs increase, the probability goes considerably high. To avoid the interference to PUs, *CSMA-like* mechanism is adopted in each time slot during the CH, with which, the rendezvous delay caused by this factor is little as illustrated in the right figure.

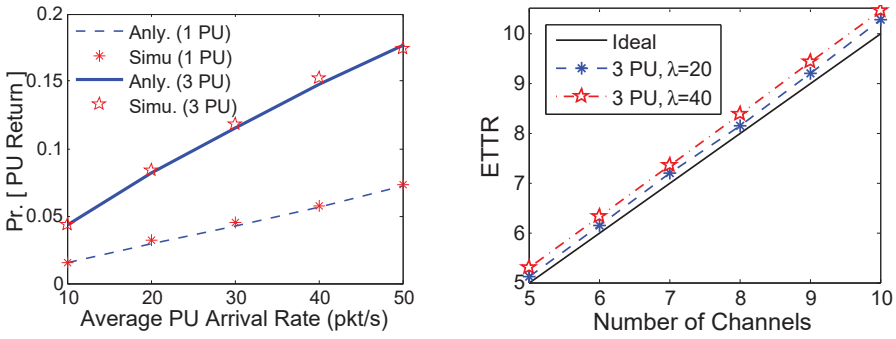


Figure 3.3: The validation for the analysis of PU return.

Neighboring SU Transmission: using a similar method, we first consider the one-neighboring-SU scenario. In Figure 3.2b, SU_2'rendezvous during SU_1'CH period

($\rho_S \frac{T_E}{T_E+T_P}$). Similarly as the analysis in PU case, the expected rendezvous moment is at the half of the SU_1's TTR. Based on the study in [11, 72], the rendezvous channel of SU_2 has a close-to-1 probability that it is also in the ACS of SU_1. SU_1 has half chance to employ this channel in the remaining CH time. Therefore, the occurrence probability of the channel status change caused by one-neighboring-SU transmission is:

$$Pr[one\ SU\ trans.] = \rho_S \frac{T_E}{2(T_E + T_P)} = \frac{\lambda_S T_E}{2}. \tag{3.3}$$

Finally, the probability of neighboring SUs' transmissions is:

$$Pr[SUs'\ trans.] = 1 - (1 - Pr[one\ SU\ trans.])^{K_S}. \tag{3.4}$$

Since it only related with λ_S, T_E, and K_S, all of which are secondary network parameters, it is a unique factor inherent with the system causing the channel status change during CH. In other words, no matter how other system parameters changes, like the link speed and the packet size, the occurrence probability of this factor is still the same.

SU Mobility: very few rendezvous papers consider the influence of nodes' mobility. In this chapter we also consider PUs and SUs in CRNs as static nodes during the CH. However, we give a proof to justify this assumption.

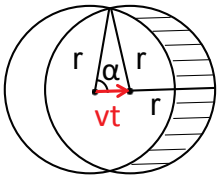


Figure 3.4: SU's Mobility.

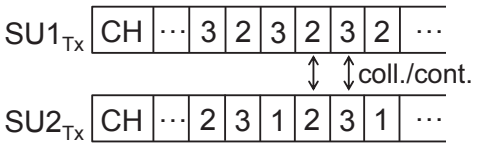


Figure 3.5: The RTS collision/contention.

In Figure 3.4, the circle represents the sensing range of a SU with a radius r. We assume that the speed of the SU is $\boldsymbol{v}$ which is a relative speed compared to surrounding nodes. The shadow part is the additional sensing area during the SU's moving. Note

that the moving time $t = T_R$ and $\alpha = arccos\frac{vt}{2r}$.

If we assume that PUs are evenly distributed in the network, the ratio of K'_P to K_P, i.e., the number of new PUs in the SU's new sensing area to the number of original neighboring PUs of the SU, is the same as the ratio of the shadow area size to the original circular area size, then

$$K'_P = \frac{2\left(\pi r^2 \frac{180-\alpha}{360} - (\pi r^2 \frac{\alpha}{360} - \frac{vt}{2} r sin\alpha)\right)}{\pi r^2} K_P.$$

We can also get the number of new encountered SUs as

$$K'_S = \frac{2\left(\pi r^2 \frac{180-\alpha}{360} - (\pi r^2 \frac{\alpha}{360} - \frac{vt}{2} r sin\alpha)\right)}{\pi r^2} K_S.$$

We consider the probability that a new encountered neighboring PU/SU is busy on one of the SU's available channels as $Pr[busy|PU'] = \rho_P \frac{M}{N}$ or $Pr[busy|SU'] = \rho_S \frac{T_P}{T_P+T_E}$. Therefore, the probability of the available channel status changing due to SU's mobility during the CH is

$$Pr(Mobi.) = 1 - \left(1 - \rho_p \frac{M}{N}\right)^{K'_P} \left(1 - \rho_S \frac{T_P}{T_E + T_P}\gamma\right)^{K'_S}. \tag{3.5}$$

Even in a dense-node high-traffic CRN where $\rho_P = \rho_S = 0.5$, $K_P = K_S = 5$, and $ETTR = 15$, $Pr(Mobility) = 0.99\%$ if $\boldsymbol{v} = 10m/s$, $r = 10m$. Even if we increase the speed $\boldsymbol{v}$ to $30m/s$ which is equivalent to the highway vehicular scenario, $Pr(Mobility) = 2.95\%$ which is still negligible. Thanks to the high link speed maintaining T_E in a small level. However, if the ratio of the node speed to its sensing range increases, the above probability will become considerable.

RTS Collision/Contention: the RTS collision at the receiver side is an unavoidable issue in real networks. It also exists in CRNs. As explained in the beginning of this section, the RTS collision from two hidden SUs will result in the correspond-

ing channel temporarily unavailable for a receiver in their overlapping transmission ranges, or, the RTS contention from two exposed SUs will make the corresponding channel unavailable to one of them. Both cases sacrifice at least one SU's broadcast opportunity and thus lead to the potential rendezvous delay.

In the following we demonstrate the probabilistic model of the RTS collision/contention. As illustrated in Figure 3.5, two SU senders hop on to a same channel at the same time. We have the lemma:

Lemma 1. *The probability that a source SU rendezvous with a listening SU is the same as the probability that its RTS collides/contends with another SU who is also in the CH period.*

Proof. If a sender and a receiver hop to a same channel at the same time, it is a successful rendezvous. On the other hand, if the receiver is replaced with a sender, it becomes an RTS collision/contention case. □

If SU_1 and SU_2 are neighboring SUs and SU_1 starts the broadcast during SU_2's CH period ($\rho_S \frac{T_E}{T_E+T_P}$), the expected overlapping CH duration between the two SUs is the half time of the ETTR, which makes the contention probability to be ($\frac{T_E/2}{T_E}$). Symmetrically, if SU_2 begins the broadcast during SU_1's CH period, the case has the same chance to happen. Therefore, the RTS contention probability between the two SUs is

$$Pr[RTS\ contention|2SUs] = \rho_S \frac{T_E}{T_E + T_P}. \tag{3.6}$$

If SU_1 and SU_2 are hidden terminals, when the collision happens, any common neighboring SU in the listening state $(1 - \rho_S)$ can hear the collision when it hops on to the occasional channel in the same time slot ($\frac{1}{M}\frac{M}{N}$). Therefore, the RTS collision probability between the two SUs is

$$Pr[RTS\ collision|2\ SUs] = \rho_S(1 - \rho_S)\frac{T_E}{T_E + T_P}\frac{1}{N}. \tag{3.7}$$

At last, the total RTS collision/contention probability caused by a source SU with K_S neighboring SUs or hidden SUs during its broadcast is

$$Pr[RTS\ co.] = 1 - (1 - Pr[RTS\ co.|2\ SUs])^{K_S}\ . \tag{3.8}$$

Compared with equations (3.3) and (3.4), the RTS collision/contention is also a unique factor inherent with the system causing the channel status change during the broadcast. Thus, their validations are shown together in Figure 3.6. Only RTS contention is given since the collision case is relatively negligible comparing Eq.(3.7) to Eq.(3.6).

From the results in Figure 3.6, the occurrence probability of the RTS contention is approximately twice as that of the neighbor transmission factor, which accords with our probabilistic model reflected in Eq.(3.6) and Eq.(3.3). If we suppose the busy rate of the PU is similar as that of the SU and regard the occurrence rate of the PU factor as level 1, then the occurrence rate of the two factors in Figure 3.6 can be treated as level 1 and level 2, respectively. The latter two inherent factors in secondary networks is the main reason leading the rendezvous delay. For example, in a moderate secondary network where $\rho_S = 0.3$, $N = 14$, and $K_S = 3$, as illustrated in Figure 3.6, a source SU has half chance to experience the rendezvous delay.

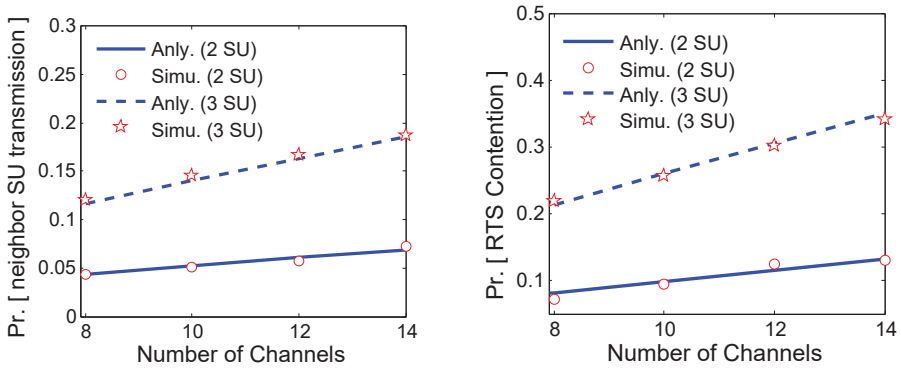


Figure 3.6: The validation for the analysis of the factors between SUs.

In summary, the above four factors usually affect the channel status simultaneously,

which may generate tremendous influence to rendezvous delay especially in a dense-node high-traffic CRN. An ACS adjustment mechanism is highly desired that can *detect the channel status change during the CH period and take reaction to compensate the rendezvous delay*, which is called *C/R mechanism* given later.

3.2.2 Factor Analysis for Rendezvous Failure

A rendezvous failure means that a source SU can not achieve a successful handshake before its STTR expired. For a source SU, there are two factors that may lead the failure from its own aspect and the outside (system) aspect, respectively.

Short STTR: a relatively short STTR easily results in rendezvous failure:

1. If the STTR is shorter than MTTR, obviously, it cannot guarantee the rendezvous even in ideal scenarios.

2. If the STTR equals to the MTTR, the rendezvous is still not guaranteed in CRNs since some TTR may exceed MTTR due to the rendezvous delay. In such cases, a rendezvous delay turns to be a rendezvous failure since the STTR expired before the delay.

3. Even if the STTR of a source SU is long enough for the possible rendezvous delay, the source SU may still fail to handshake when its invisible destination SU is not a listening SU.

Invisible Busy Role: as we discussed in Figure 3.1, a source SU cannot expect the CTS from an invisible destination SU which is also operating on the sender-mode. In this case, the source SU easily fails the rendezvous because: *i)* the sender-mode may last a long time until all queuing packets are served; *ii)* even the sender-mode for one packet usually takes longer time than the STTR (T_R+T_P or T_S); and *iii)* even the destination SU finishes its sender-mode within the source SU's STTR, the remaining time may less than T_R. Hence, the rendezvous failure probability approaches to the encounter rate of a busy role, which is ρ_S.

The occurring rate of this factor is only related with the busy rate of other SUs, which is another system factor and cannot be eliminated. Using the same example in Figure 3.6, the rendezvous failure probability due to this factor should be 30%, which is similar with the probability of the RTS factor. Follow the level setting from previous factors, the occurring rate of this factor should be level 2.

3.2.3 Factor Analysis for Network Congestion

Without loss of generality, we define a packet which is generated from a source SU being successfully transmitted to its destination receiver as one throughput. Since the failure or interference during the data transmission period is well studied in existing works [66, 67] and thus out of the scope in this chapter, we count one throughput as long as the packet sender rendezvous with its target receiver. From the global perspective of view, the normalized throughput in CRNs equals to the average rendezvous-successful-rate in the secondary network, namely P_R. The throughput process is modeled in Figure 3.7.

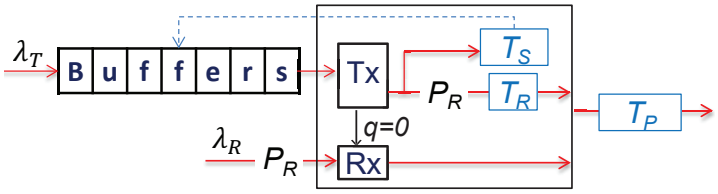


Figure 3.7: The throughput model for a SU in CRNs.

Long STTR for Self-Throughput: consider the process from the sender side. From previous analysis, the STTR should be as long as possible in order to avoid the rendezvous failure. Thus, the self-throughput, P_R^s, is positively correlated with T_S. Meanwhile, note that T_E is also positively correlated with T_S since:

$$ETTR = \sum_{j=1}^{STTR} j(1-P_1)^{j-1}P_1, \tag{3.9}$$

where P_1 represents the rendezvous probability in one time slot.

On the other hand, the service time of a packet should be properly controlled to avoid the throughput congestion. In this model, suppose λ_T is the average packets generating rate from this SU. It requires

$$\rho_T = \lambda_T \left[P_R^s(T_E + T_P) + (1 - P_R^s)T_S\right] \leq 1. \tag{3.10}$$

When T_S increases, ρ increases. Therefore, from the sender's perspective, a higher throughput requires a long STTR but within a limitation to avoid its throughput congestion.

Long STTR for Others-Throughput: consider the process from the receiver side. As illustrated in Figure 3.7, other SUs should expect this destination SU is in the receiver mode. However, only when the queue is empty ($q = 0$ in the figure), can the SU get rid of a busy role with $1 - \rho_T$. Meanwhile, if we specify P_1 in Eq. (3.9) as a theoretical parameter only counting the ideal cases (i.e., the destination SU is always in the receiver mode), P_1 turns to a preknown parameter inherent with the CH algorithm adopted. Then, the ETTR should be expressed as

$$ETTR = P_R^s \sum_{j=1}^{STTR} j(1 - P_1)^{j-1} P_1 + (1 - P_R^s)STTR, \tag{3.11}$$

and the throughput of other packets with $STTR'$ aiming at this receiver, P_R^o, can be further derived:

$$P_R^o = \left(1 - (1 - P_1)^{STTR'}\right)(1 - \rho_T). \tag{3.12}$$

Thus, a longer STTR decreases others'throughput since ρ_T increases.

In summary, a longer STTR of a source SU may increase its own throughput but may also lead to throughput congestion, while definitely decreases other SUs'throughput. Therefore, there should be a STTR generating mechanism working in the distributed manner which helps each source SU choose its own optimal STTR in order to gain

the maximum throughput of the secondary network. The method is given in the following section.

3.2.4 Practical Self-Adaptive Schemes

We propose a practical self-adaptive (PSA) rendezvous framework according to the analysis above. The features of those analyzed factors are concluded in Table 3.2, which also shown the corresponding schemes and their influence. Explanation is given as follows.

Table 3.2: Practical factors analysis

Factors	PU return	SU trans.	RTS	Role
Perceivable	Yes	Yes	Yes	No
Probability	lv. 1	lv. 1	lv. 2	lv. 2
function of	λ_P, K_P, t_E	λ_S, K_S, t_E	λ_S, K_S, t_E	λ_S, t_E
leads to	rendezvous delay			failure
Methods	C/R		optimal STTR	
self-th.	no change		depends	
others-th.	increase		depends	
Throughput	increase		maximum	

***CSMA-like* Scheme:** the CSMA-like scheme requires that in each time slot during the broadcast, the source SU sense the channel first. In this way, the first three factors in Table 3.2 are *perceivable* due to different signs during the sensing: *i)* if the PU return factor or the neighboring-SU-transmission factor takes place, the status-change channel will be sensed noisy throughout the entire time slot; *ii)* if the RTS contention factor happens, the RTS sent by the neighboring SU (unrelated RTS) will be heard; and *iii)* if the RTS collision from two hidden SUs occurred, their common neighboring SUs in the same channel will hear a noise lasting around 1∼2 RTS length (two RTS with overlapping time). Emphasized by [73], Software Defined Radio (SDR) gives packets an opportunity to process beyond physical layer. Thus, at least based on the noise duration length, these signs can be notified to the controller in MAC layer. Note that the occurrence of these factors can also be detected by listening SUs since they are listening on every hopping channel all the time. The

concrete scheme is given in Algorithm 1 (the reaction part is released in Algorithm 3 later).

Algorithm 1: The *CSMA-like* protocol in one time slot

1: Sensing the current channel;
2: **if** (*idle*) $sign = 0$;
3: **if** ($t_{busy} \geq 2t_{RTS}$) $sign = 1$;
4: **if** ($t_{RTS} < t_{busy} \leq 2t_{RTS}$ **or** *unrelated RTS*) go to 1;
5: **if** (*related RTS*) $sign = 2$;
6: **if** (*unrelated CTS*) $sign = 3$;
7: **return** *sign*;

The difference with the traditional CSMA mainly falls on two aspects. One is the retransmission mechanism. When a channel in the ACS is detected busy during CH, it is unnecessary to backup and retransmit the RTS in the current time slot: *i)* if the busy is caused by PU/SU transmission, the busy status will last through the entire time slot; *ii)* if the busy is caused by RTS contention/collision, the remaining idle time in the time slot is not enough for the RTS/CTS handshake since the size of a time slot is usually less than the length of two RTS/CTS exchange for the sake of quick rendezvous [74]; and *iii)* the destination SU may not on the same channel in this time slot. Another difference is particularly designed to increase the rendezvous chance in the invisible-busy-role cases: when a broadcasting SU occasionally hears a RTS message from another SU indicating itself as the destination SU (related RTS), it should backup the current packet and send the CTS to prepare receiving the packet from that SU.

The benefits of this function are obvious. First, the potential interference caused by a source SU to its neighboring PUs and SUs during the broadcast period is prohibited. Then, the potential RTS collision from neighboring SUs is turned to the RTS contention case, which increases half chance of each SU to broadcast RTS without interference depending on who sends first. At last, the enabled listening ability helps the busy destination SU still has the chance to hear the RTS from its source SU in

the CH period, which mitigates the invisible role problem.

Nevertheless, the status-change channel still cannot be used and the role of one's neighboring SU is still hard to know. In other words, the CSMA-like scheme alone cannot solve the rendezvous delay and rendezvous failure. The methods help address the problem are given in the following subsections.

C/R Scheme: as mentioned previously, only avoiding interference on the status-changed channel cannot mitigate the impact of the rendezvous delay. Thus, we further design a reaction scheme once a SU realizes the channel status change, which can fasten the rendezvous speed and compensate the rendezvous delay. The main idea is to cut or replace (C/R) those status-changed channel during CH.

theory: if the ACS of each SU is downsized while their common available channels are maintained, a shorter rendezvous time can be achieved, which is mathematically proved in [19, 75]. Motivated by this theory, a source/listening SU can cut its unavailable channels to fasten the rendezvous. In PSA, when the unavailable channel is detected during the CH process, two reactions are offered according to different CH algorithms. For the memoryless algorithm like RCH, it removes the unavailable channel from their ACS then starts the new CH with a downsized ACS. For those sequence-based algorithms such as EJS and QoS, in order to maintain the sequence, they can replace the unavailable channel with an available channel randomly chosen from its remaining ACS. Compared with the unavailable channel, hopping on the substituted channel has certain chance to rendezvous. Thus, both reactions help SUs compensate the rendezvous delay.

example: we conduct simulations to justify our idea. We use the same parameters and CH sequence as in [32] and assume that a SU knows which channel becomes unavailable during the CH. The TTR and rendezvous successful rate of no adjustment and cut-channel reaction are compared in Table 3.3.

All the numbers in Table 3.3 without brackets are the average values and their

Table 3.3: No adjustment vs. channel removal

Metrics	A common available channel changes		An uncommon available channel changes	
	No Act	Cut	No Act	Cut
ETTR	4.2 [1.0]	2.2 [0.7]	2.7 [0.8]	2.1 [0.7]
MTTR	9.1 [3.0]	3.9 [1.9]	5.5 [2.3]	3.8 [1.9]
Suc.	80.15%		100%	

standard deviation is shown in brackets. From the table we can conclude: *(i)* cutting a common available channel cannot save those lost successful rate, but can mitigate rendezvous delay; and *(ii)* if an uncommon available channel has been cut, the successful rate is not affected, but the rendezvous delay is significantly reduced.

algorithm: the use of C/R scheme is shown in Algorithm 3. In short, the C/R scheme can only be triggered when the status of a channel is confirmed to be unavailable for a long time even beyond the CH period. We conclude the triggering cases as follows.

First, C/R is used for the first two factors in Table 3.2, since PU return and neighboring SU transmission long-time employ a channel in the ACS. However, C/R cannot be used for the third factor though it is perceivable. The RTS collision/contention does not help the occurring channel become the rendezvous/transmission channel for other SUs. Therefore, the collision/contention channel may still be available on the next hopping time.

Another scenario to use C/R is when receiving an unrelated CTS. In traditional CSMA protocol, user prepares a NAV time period to avoid transmitting on the channel. In PSA, an unrelated CTS indicates a successful handshake of other SUs. It also means the channel will be occupied for a long time for data transmission, which satisfies with the C/R triggering condition.

The last case to perform C/R is only for listening SUs who overhear the CTS collision, as illustrated in Figure 3.8. The CTS collision indicates that at least two receivers (R1 and R2) have successfully received the RTS from their senders (S1 and

S2) and the CTS collision happens at their common listening neighbor, R3. Since both R1 and R2 will receive data on this channel from the next time slot, in a long time R3 should avoid handshake on this channel for the sake of others' interference-free transmission.

(a) The RTS period (b) The CTS period

Figure 3.8: An illustration of the CTS collision.

Note that *only listening SUs have the chance to overhear the CTS collision.* For example, if R3 is a source SU, R1 and R2 will experience the RTS collision during the first period in Figure 3.8. Then, no CTS can be sent in the following period. Based on this rule, the CTS collision is also perceivable since it only takes place in the later period in a time slot.

Although C/R scheme does not increase the rendezvous probability, the compensated rendezvous delay fastens the service time of packets and thus increases the rendezvous chance to receive packets from other SUs. Overall, it increases the network throughput, which is concluded in Table 3.2.

STTR Scheme: from Table 3.2, a lower T_E can mitigate the occurrence rate of all factors, especially the invisible busy role, to increase the rendezvous probability of other SUs. Meanwhile, T_E can be decreased by a shorter STTR from Eq. (3.11). However, a short STTR is the factor that leads to self-rendezvous failure. Thus, an optimal STTR to balance them and gain the maximum network throughput is highly desired. As for the self-throughput and the others-throughput, their change depend on which factor dominates the result with the optimal STTR.

Derivation: it is difficult to have a precise probabilistic model for the network throughput for CRNs due to its unpredictable features and the practical factors.

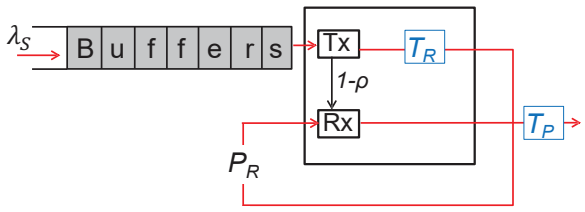


Figure 3.9: The optimization model for congestion.

However, from a macroscopic view, since all the packets generated from all SUs go to all the SUs in the network, we can treat the throughput of the whole network as that of a single SU scenario. Thus, we modify the throughput model in Figure 3.7 to better represent both a SU's status and its neighbors'as shown in Figure 3.9.

A successful throughput begins from the buffer in the sender. Once a packet is taken from the queue, the sender has to rendezvous within STTR, to avoid throughput congestion. The packet has a probability P_R to be successfully received by the target receiver. Finally, after a successful rendezvous, this packet can be counted as one throughput. Therefore, P_R is the normalized throughput we want to maximize and the optimal STTR should well balance the SU'busy rate rho. Such a model performed in each individual helps to enhance the local traffic balance when combining with the existing congestion control protocol in the transport layer. In other words, the potential traffic rate from outside of a SU should be similar to the SU's internal traffic rate, when the local network is controlled in a stable status by each SU updating its optimal STTR.

Consequently, a SU's optimal STTR can be derived by:

$$\begin{aligned} &\underset{STTR}{\text{Maximize}} \quad P_R \\ &\text{subject to} \quad (10),\ (11),\ (12),\ P_R^o = P_R^s, \\ &\qquad\qquad STTR = STTR',\ \text{and}\ \lambda_T = \lambda_R = \lambda_S. \end{aligned} \tag{3.13}$$

Finally, P_R is a function of P_1, T_P, λ_T, and $STTR$. The first two parameters are theoretical value preknown and λ_T can be easily obtained by simply observing a SU's

own history. The optimal STTR is thus derived by simply solving Eq.(13).

Note that the throughput under the optimal STTR derived from this model may be not accord with the real value, since the ETTR is also influenced by the rendezvous delay factors which are not considered. However, the STTR derived from this model can still hold the optimality. This is because that the pattern of the throughput is mainly affected by the STTR.

Application: during the CH period, on one hand, SU updates its optimal STTR dynamically to adapt with the traffic change. On the other hand, in order to further increase the throughput, SU can also dynamically adjust the size of its ACS to adapt with the optimal STTR under some certain circumstances. The comprehensive STTR scheme is given in Algorithm 2, followed by the entire PSA protocol in Algorithm 3.

Algorithm 2: The optimal STTR updating protocol

Require: ACS, CH algorithm and self-history database;
1: Get P_1 and $MTTR$ from the adopted algorithm;
2: Get T_P, λ_S, and γ from the database;
3: Derive STTR from Eq(3.13);
4: **if** $STTR \leq MTTR$ ***and*** $MTTR \propto ACS_{size}$ **then**
    if $\lfloor\sqrt{STTR}\rfloor \geq \gamma$ **then**
        Downsize ACS to $\lfloor\sqrt{STTR}\rfloor$;
5: Go to 1 when database changes;

For some CH algorithms, their theoretical MTTR is mainly related with the size of the ACS, M. For example, the MTTR of QoS is M^2. When the STTR derived is less than MTTR, we can further downsize the ACS size to $\lfloor\sqrt{STTR}\rfloor$ (e.g., keep the first $\lfloor\sqrt{STTR}\rfloor$ available channels and cut the remaining channels). If the new ACS remains at least one common available channel, the theoretical rendezvous rate within STTR will be increased to 100%. Such downsize method is encouraged to use unless $\lfloor\sqrt{STTR}\rfloor$ is too small to possibly include one common available channel. The minimum ACS size γ can be set depending on the acceptable rendezvous rate in corresponding environment.

Algorithm 3: The rendezvous control protocol

```
1: Calculate STTR from Algorithm 2;
2: CH by updated ACS and the algorithm adopted;
3: Perform CSMA-like scheme by Algorithm 1;
4: if sign = 1 or sign = 3 then
       C/R; go to 2;
5: if sign = 2 then
       Backup packet; send CTS; go to 7;
6: if sign = 0 then
       if q = 0 then                                          /* listening SU */
           Listening to the channel;
           if (related RTS)  send CTS; go to 7;
           if noise then                                      /* CTS collision */
               C/R; go to 2;
           if (nothing received)  go to 2;
       if q ≠ 0 then                                          /* source SU */
           if within STTR then
               Send RTS; listening;
               if (CTS received)  go to 7;
               else  go to 2;
           else  Drop/backup packet; go to 2;
7: Data Transmission;
```

So far, the content and the size of a ACS are self-adaptive with the occurrence of practical factors and the optimal STTR, respectively. Besides the optimal STTR is also self-adaptive with the local traffic change. Thus, the proposed rendezvous control framework is practical and self-adaptive.

3.2.5 Performance Evaluation

Table 3.4: Simulation parameters for PSA

Number of PUs/SUs	30
PU/SU packet size	50 slots
Simulation time	100000 slots
Simulation area	5×5
SU sensing radius	1
Channel data rate	2 Mbps
The size of RTS/CTS	160/112 bits
The size of one time slot	600 bits

In our simulation, we do not impose any impractical assumptions explained in Section 1 which are widely adopted in most of other existing CRN papers. Meanwhile, in order to satisfy the features of CRNs, 1) PUs and SUs are randomly distributed in the simulation area, so the number of neighboring PUs/SUs is different and unpredictable for each SU; 2) Each PU is randomly assigned a channel when a new packet needs to be transmitted, so the available channels are unpredictable; and 3) The destination SU of each packet is also randomly chosen, so the rendezvous pair changes dynamically. To be specified, packet arrivals follow the Poisson distribution. The parameters used in our simulation are listed in Table 3.4.

3.2.5.1 Optimization Validation

Figure 3.10 illustrates the normalized throughput under different STTR in various CRNs. The simulation results match with our analysis: the real throughput is lower than the analytical result but the optimal STTR derived from the analytical model holds the optimality in real cases. The pattern shape of the throughput is mainly affected by STTR since the rendezvous delay has limited impact on the throughput.

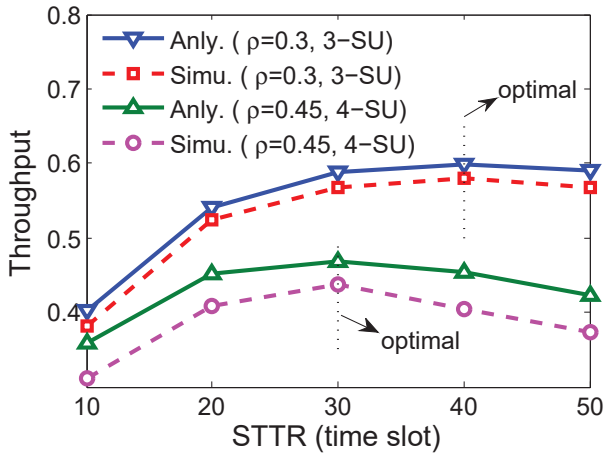


Figure 3.10: Numerical results of throughput vs. STTR (RCH adopted).

Note that the optimal STTR shifts to the shorter side when the local secondary network becomes busy (ρ increases) and crowed (K_S increases). This phenomenon indicates the long-STTR factor dominates the throughput decrease when the network is under a severe condition.

3.2.5.2 The Impact of Optimal STTR

Figure 3.11 shows the performance of different CH algorithms with the optimal STTR under various scenarios. Compared with their performance lacking our STTR updating scheme, the throughput of both the RCH and EJS is largely maintained against different network conditions.

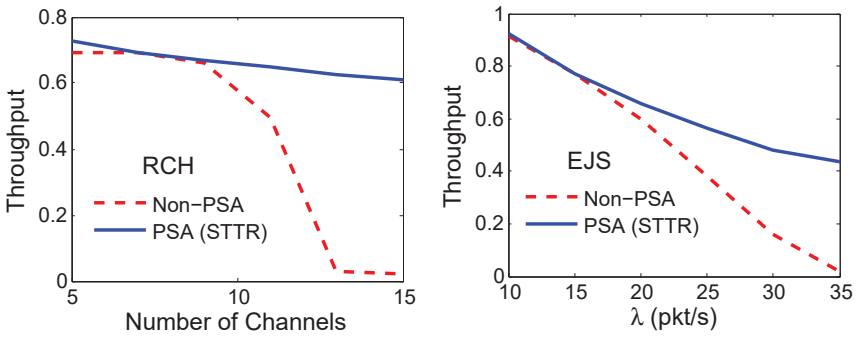


Figure 3.11: Performance of CH algorithms with the STTR scheme.

The impact of the number of total channels (N) is shown in the left figure. When

the number of channels in the network increases, the theoretical rendezvous time increases ($O(N)$). However, the STTR cannot be simply increased to match with the theoretical MTTR without considering the generated traffic congestion. Therefore, the throughput is stuck by the congestion without our proper STTR and declined to 0 after N is larger than 13. On the other hand, unlike our STTR scheme, since existing CH schemes only consider service time for each single packet, they lack a timely method to adapt to the various traffic rate to avoid throughput congestion, as illustrated in the right figure.

3.2.5.3 The Impact of C/R scheme

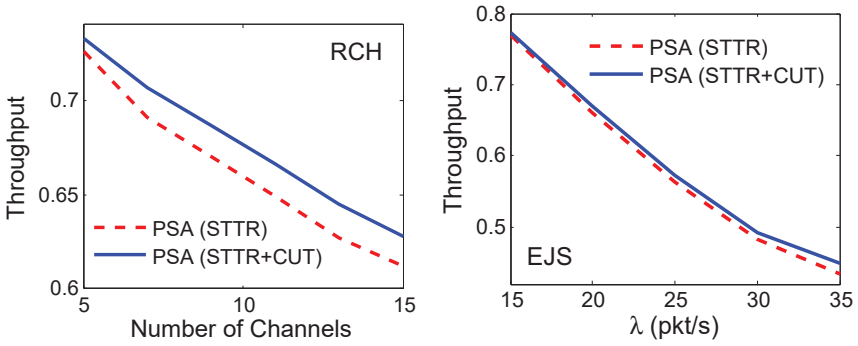


Figure 3.12: Performance of CH algorithms with STTR and C/R schemes.

Similarly, we conduct simulation to evaluate the C/R scheme. As shown in Figure 3.12, C/R scheme can further increase the throughput with the optimal STTR. Meanwhile, the influence of C/R scheme gains more weight as the N and λ increases since it compensates more rendezvous delay in such scenarios, which can also be derived from Table 3.2. Nevertheless, the difference of the change is negligible in both conditions, which also explains why the rendezvous delay cannot influence the shape pattern of the throughput in Figure 3.10 compared with the STTR.

3.2.5.4 The PSA Performance

Figure 3.13 demonstrates the different PSA performance due to its imposed CH algorithm. PQoS (QoS algorithm with PSA framework) has an obvious better per-

formance than PRCH and PEJS, since only PQoS can intelligently downsize its ACS to adapt with the optimal STTR for higher throughput.

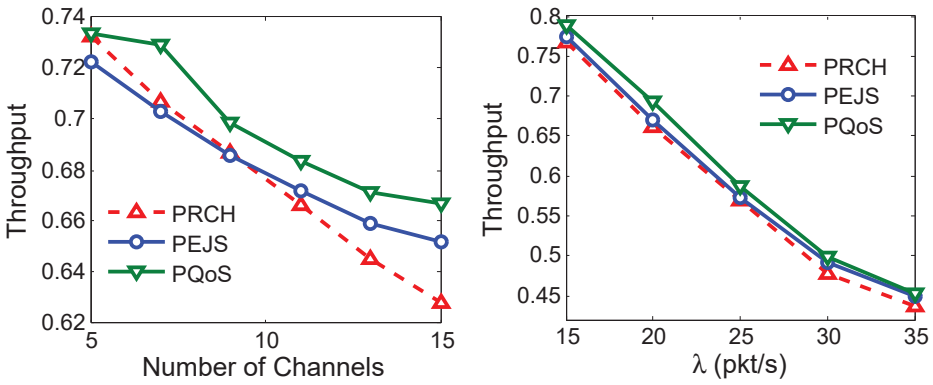


Figure 3.13: Performance of PSA under different CH algorithms.

Nevertheless, they all have a overwhelming performance against their non-PSA schemes. Based on such results, researchers should pay more attention to the practical rendezvous design instead of the theoretical CH design.

At last, the ETTR under PSA framework compared with its original value is summarized in Table 3.5. We use the ETTR of EJS to represent the non-framework situation since others' ETTR is very similar. We conclude that: *i)* without PSA, each SU spends more time in CH period when the local traffic increases since its destination SU has a more chance to be an invisible-busy-role; *ii)* the CH time of each SU is given less by PSA when traffic goes higher, which helps others' throughput and notifies the transport layer more timely for congestion control; and *iii)* the merits of PSA is unified for whichever CH algorithm.

Table 3.5: ETTR vs λ

λ (pkt/s)	10	15	20	25	30	35
EJS	23.59	39.10	56.15	72.85	88.77	98.85
PRCH	27.63	25.42	27.94	18.12	20.19	14.43
PEJS	29.26	24.31	29.92	23.63	19.89	14.31
PQoS	27.03	23.54	28.29	22.56	19.58	13.96

3.3 The Proposed Slot-asynchronous Handshake Protocol

In this section, the challenge of slot-asynchronous rendezvous in CRNs is addressed for the first time. A protocol aiming to improve the handshake performance during the CH process is proposed. By analyzing the potential factors leading to the handshake failure, we design a novel MAC protocol with an optimal size of a time slot which can mitigate the effects of these factors and provide the shortest time for rendezvous. In addition, we also propose a probabilistic model for estimating the average rendezvous time under different CRNs. Simulation results validate our analytical model and demonstrate that our proposed protocol can achieve the rendezvous time close to the theoretical value under slot-asynchronous scenarios.

3.3.1 Analysis of Handshake Failure Problems

We analyze three main problems that may result in handshake failure in one time slot and at the same time establish our protocol design to solve these problems.

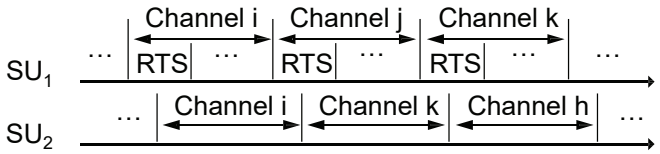


Figure 3.14: The RTS failure receiving cases.

Analysis of the Failure Receiving Problem: in an asynchronous CRN, as illustrated in Figure 3.14, a passive SU may not receive a complete RTS due to hopping onto a potential rendezvous channel later than the starting time of an RTS sending (on channel i), or leaving earlier before the sending finishes (on channel k).

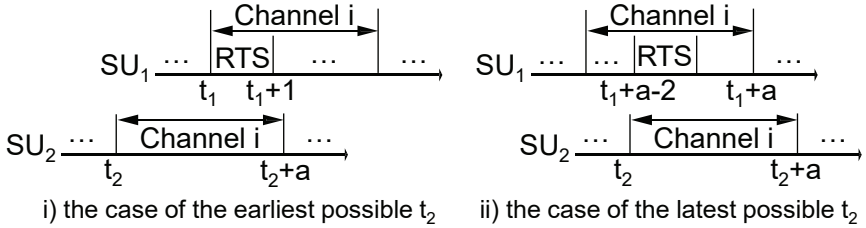
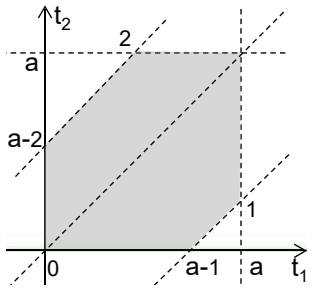


Figure 3.15: The cases that at least one RTS can be completely received.

Let SU_1 and SU_2 arrive on a same channel at moments t_1 and t_2, respectively. We normalize the length of sending an RTS/CTS to 1. If the length of a time slot is a, a should be longer than 2 so that at least one pair of RTS and CTS exchange can be completed in a time slot. In addition, we have the constraint $|t_1 - t_2| \leq a$ to ensure that SU_1 and SU_2 have overlapping time on the common channel. There are also the following constraints (see Figure 3.15) to help SU_2 hear at least one complete RTS from SU_1. If $t_1 \geq t_2$ (SU_1 hops on the channel later than SU_2), the leaving time of SU_2 should be at least later than the end time of the first RTS sent by SU_1 on the common channel, i.e., $t_2 + a \geq t_1 + 1$. If $t_1 \leq t_2$ (SU_1 hops on the channel earlier than SU_2), the arriving time of SU_2 should be at least earlier than the start time of the last possible RTS sent by SU_1 on the common channel, i.e., $t_2 \leq t_1 + a - 2$. Note that after each RTS is sent out, a SU must wait for a while for the potential CTS. Thus, the last RTS in the current time slot must be sent before $a - 2$.

To sum up, we have the following equivalent inequalities and their corresponding graphic illustration:

$$\begin{cases} 0 \leq t_1 \leq a \\ 0 \leq t_2 \leq a \\ t_1 - a + 1 \leq t_2 \leq t_1 \\ t_1 \leq t_2 \leq t_1 + a - 2 \end{cases}$$



The above shadow area represents the feasible ranges of t_1 and t_2 to ensure the receiving of at least one complete RTS. Therefore, we can derive the probability that a passive SU receives a complete RTS from another SU on a channel in the asynchronous scenario, P_1,

$$P_1 = \frac{\text{size of the shadow area}}{\text{size of the } a \times a \text{ square}} = \frac{a^2 - 2.5}{a^2}, \qquad a \geq 2. \tag{3.14}$$

In synchronous CRNs, it is natural to define the size of one time slot to be the length of an RTS and a CTS exchange for the sake of quick rendezvous, i.e., $a = 2$. However, according to Eq.(3.14), this design leads to a probability of 0.375 to have a successful RTS reception even after two SUs hop on a same channel in the asynchronous scenario. Since P_1 is a monotonically increasing function of a, from this point of view, we should design the length of a time slot as long as possible and let a SU keep sending RTS until the current time slot ends.

Analysis of the Neighboring Interference Problem: in CRNs, especially in CRAHNs, several other SUs may be within a SU's sensing range. Hence, three or more SUs may hop on a same channel in one time slot during their rendezvous processes. They may interfere with each other in two scenarios. One is the presence of RTS collisions in the hidden terminal case. The other is the continuous contention for sending RTS between active neighboring SUs in one time slot.

In traditional wireless networks, one reason that an active node cannot receive the correct CTS is the RTS collision from a hidden terminal. Thus, 802.11 CSMA/CA requests a node to perform a binary exponential backoff when experiencing the absence of CTS. However, this mechanism may not increase the successful rate of handshake when applied to CRNs, since a more possible reason for the absence of CTS is that the destination node is not on the same channel. In addition, to support backoff, the size of a time slot needs to be unacceptable long. Moreover, the backoff SU may still collide with a new SU who just hops on this channel after the backoff under the asynchronous scenario. On the other hand, each time when a SU resends an RTS, it is an additional contender for other SUs. If the destination SU is not on the same channel, the source SU keeps rejoining the contention, which affects other SU's opportunities to send the RTS.

For example, in Figure 3.16, SU_1 has successfully sent an RTS several times in a time slot (gray RTS/CTS means the supposed sending/receiving but not achieved).

If SU_2 is a hidden terminal of SU_1, it will collide with SU_1's jth resend. If SU_2 is a neighbor of SU_1, it will lose the opportunity to send its RTS because of SU_1's kth resend. If SU_1's destination SU is absent on this channel during this time slot, this contention keeps happening till SU_1 leaves the channel.

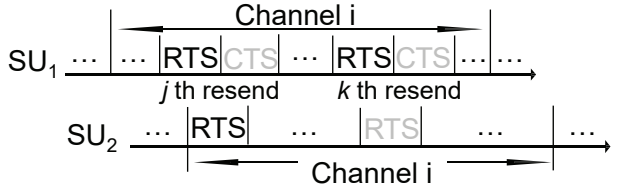


Figure 3.16: The neighboring interference cases.

Therefore, the traditional method for resolving the RTS collision is not desirable for asynchronous rendezvous in CRNs. A better mechanism is required in our protocol. Note that P_1 is only affected by three factors: the sending moments of the first and the last RTS in a time slot, and the length of a time slot. Based on this observation, we redesign the protocol which can solve the neighboring interference problem and meanwhile has an equivalent effect as the previous design.

We propose that an active SU only sends an RTS twice in a time slot: one at the beginning and one at the end of a time slot if a channel is idle, and listens to the channel during other periods, as the SU_1 illustrated in Figure 3.17. However, if the length of a time slot, a, is not long enough for sending the second RTS (with CTS receiving), a SU gives up the resend and listens to the channel until the current time slot ends. Thus, we design the length of a time slot to be either $a > 4$ (neglect the lengths of DIFS, SIFS, and the contention-window in 802.11) or $a = 2$ (without resending and contention mechanisms in a time slot). On the other hand, if a SU senses a channel busy, it waits to send the first RTS until the channel idle long enough for a CTS time, as the SU_2 in Figure 3.17. The gap between the dashed line and the solid line is the time for DIFS and the backoff time for contention. We neglect these obligatory frames in our analysis.

Let SU_2 be a source SU and SU_1 be its neighbor. Let t_2 and t_1 be their arrival times

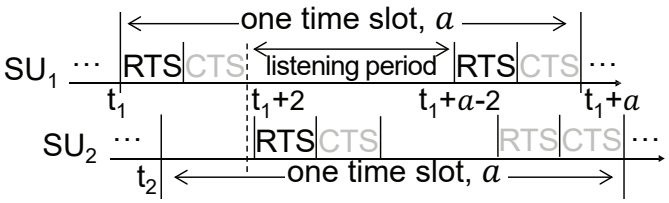


Figure 3.17: The revised resending mechanism.

on a same channel. Assume that their destination SUs are not on the same channel. If $t_2 < t_1$, SU_2 can send its RTS in this time slot. If $t_2 \geq t_1$, in the $a = 2$ case, SU_2 can send its RTS if arriving after SU_1 finishing its RTS sending, i.e., $t_2 \geq t_1 + 1$. Using the same constraint $|t_1 - t_2| \leq a$ and solving the inequalities the same way as P_1, the probability that a SU can successfully send an RTS with a neighbor SU on the same channel is 0.625 when $a = 2$. In the $a > 4$ case, even SU_2 arrives during SU_1's first RTS sending time as in Figure 5, SU_2 can still send its RTS as long as the moment it starts to send is earlier than SU_1's second RTS sending, i.e., $t_1 + 2 < t_1 + a - 2$, which always stands since $a > 4$. Moreover, if SU_2 arrives during SU_1's second RTS sending ($t_1 + a - 2 < t_2 < t_1 + a - 1$), we have $t_2 > t_1 + 2$ when $a > 4$, or, $t_2 + a - (t_1 + a) > 2$. In other words, SU_2 has enough time for sending its RTS after SU_1 leaves the channel. Hence, the probability that a SU can successfully send an RTS, P_2, increases to 100% when $a > 4$. Therefore,

$$P_2 = \begin{cases} 0.625, & a = 2 \\ 1, & a > 4 \end{cases}. \tag{3.15}$$

This design also reduces the RTS collision rate due to the low RTS sending frequency. Compared with the traditional design, the listening period in the middle of a time slot provides the opportunity for another SU to send an RTS without collision. Suppose that SU_2 is a hidden terminal of SU_1. Then, the case that SU_1 can successfully send at least one RTS without collision is when the RTS from SU_1 has no overlap with the RTS from SU_2, i.e., $t_1 + 1 < t_2$, or $t_1 + a - 2 > t_2 + a - 1$. We

name this probability P_3. We derive it in a similar way as P_1,

$$P_3 = \frac{(a-1)^2}{a^2}, \qquad a = 2 \text{ or } a > 4. \tag{3.16}$$

It also requires the a to be as long as possible to achieve the collision-free sending.

Analysis of the Both-Shouting Problem: we already consider the failure receiving problem when the destination SU is a passive SU. However, it may also be another active SU. Consequently, when a SU is hopping and searching for its destination SU, the target SU may also be searching for another SU. An extreme case is that they are searching for each other, which is a deadlock when $a = 2$.

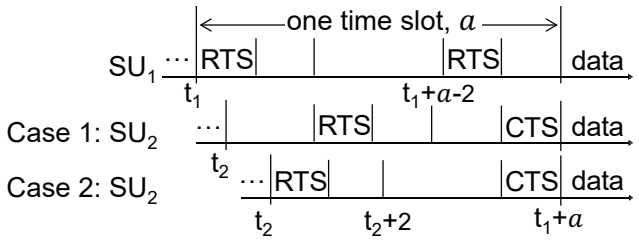


Figure 3.18: Two successful cases when $t_1 < t_2$ under the both-shouting scenario.

Let SU_1 be a source SU and SU_2 be the target SU which is also an active SU when it arrives on the same channel. If $t_1 < t_2$, there are two cases in which SU_2 can hear a complete RTS from SU_1. Case 1 is that SU_2 arrives during SU_1's sending its first RTS. As illustrated in Figure 3.18, based on our previous design, after waiting for a CTS-long idle period, SU_2 starts to send its own RTS. If SU_2's target SU is SU_1, SU_1 replies a CTS and they begin to transmit data. In other words, the deadlock case can be easily solved to have a successful handshake when $a > 4$. If SU_2's target SU is not SU_1, then SU_1 waits until the moment $t_1 + a - 2$. If the channel is idle, SU_1 sends its last RTS and SU_2 will hear the last RTS from SU_1. In this case, it requires that $t_1 + 4 \leq t_1 + a - 2$ (enough time to send the last RTS), or, $a \geq 6$. Case 2 is that SU_2 arrives after SU_1's first RTS ($t_2 \geq t_1 + 1$). SU_2 senses the channel idle and sends its first RTS after arriving. Similarly, SU_1 overhears this RTS and waits until $t_2 + 2$

if it still has time to send its last RTS, i.e., $t_2 + 2 \leq t_1 + a - 2$. It requires that $a \geq 5$ in this case.

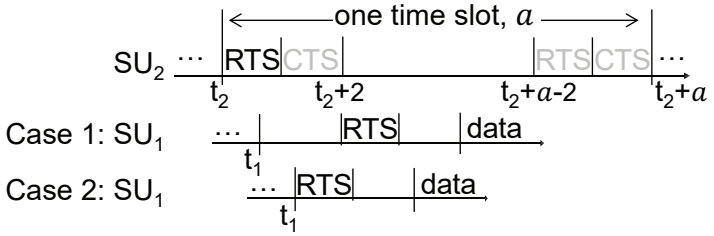
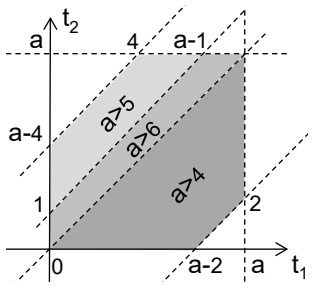


Figure 3.19: Two successful cases when $t_1 > t_2$ under the both-shouting scenario.

If $t_1 > t_2$, we also analyze two cases, as illustrated in Figure 3.19. In case 1, $t_2 < t_1 < t_2 + 1$, SU_1 starts to send at $t_2 + 2$, which should before SU_2's second RTS sending ($t_2 + 2 \leq t_2 + a - 2$). Then, $a > 4$ is required for this case to ensure a successful RTS reception at SU_2. In case 2, $t_2 + 1 \leq t_1$, SU_1 can send its first RTS after arriving. SU_2 will hear this RTS as long as it is sent before SU_2's last RTS ($t_1 \leq t_2 + a - 2$). This case requires that $a \geq 3$. In both cases, SUs end their current time slot immediately once the handshake is finished. To sum up, we have the following equivalent inequalities and their corresponding graphic illustration:

$$\begin{cases} 0 \leq t_1 \leq a \\ 0 \leq t_2 \leq a \\ t_1 \leq t_2 \leq t_1 + 1 \\ t_1 + 1 \leq t_2 \leq t_1 + a - 4 \\ t_2 \leq t_1 \leq t_2 + a - 2 \end{cases}$$



The above shadow area represents the feasible ranges of t_1 and t_2 under different values of a. Therefore, we can derive the probability that a SU's RTS can be heard by its destination active SU on the same channel, P_4, which is also a monotonically

increasing function of a.

$$P_4 = \begin{cases} 0, & a = 2 \\ (a^2 - 4)/2a^2, & 4 < a < 5 \\ (2a^2 - 2a - 19)/2a^2, & 5 \leq a < 6 \\ (a^2 - 10)/a^2, & a \geq 6 \end{cases} . \tag{3.17}$$

3.3.2 Protocol Details

The overall flow chart for our proposed MAC protocol is presented in Figure 3.20. In the initial period, a SU senses all channels and collects its available channel set. Existing sequence/probabilistic-based CH designs can be employed in this step to generate the CH sequence. Then, the SU tunes its radio to the ordered channel and begins a new time slot. During a time slot, any SU can become a destination node once it receives an RTS carrying its ID as the receiver. If the SU also has data to send, it postpones its own data in queue and receives other's transmission first. On the other hand, a passive SU can become a source node once it has data to send. In synchronous environments, it has to wait till the next time slot to change its role. However, in our design, it sends an RTS immediately if there is still time left in the current time slot since there is no need for slot-synchronization. Once a pair of SUs completes the handshake in a time slot, they stay on the same channel transmitting data until they finish the communication. When the pair detects a PU presence on the channel, a spectrum handoff [67] is performed for resuming the transmission on another channel.

In the figure, "Left time = 2" refers to the moment approaching $a-2$. The backoff for the last RTS sending is counted in a reverse time from $a-2$. For example, if a random number 0.2 is generated for the backoff time, the last RTS will be sent from the moment $a-2-0.2$. The use of SIFS and DIFS in our protocol is the same as in

802.11 MAC.

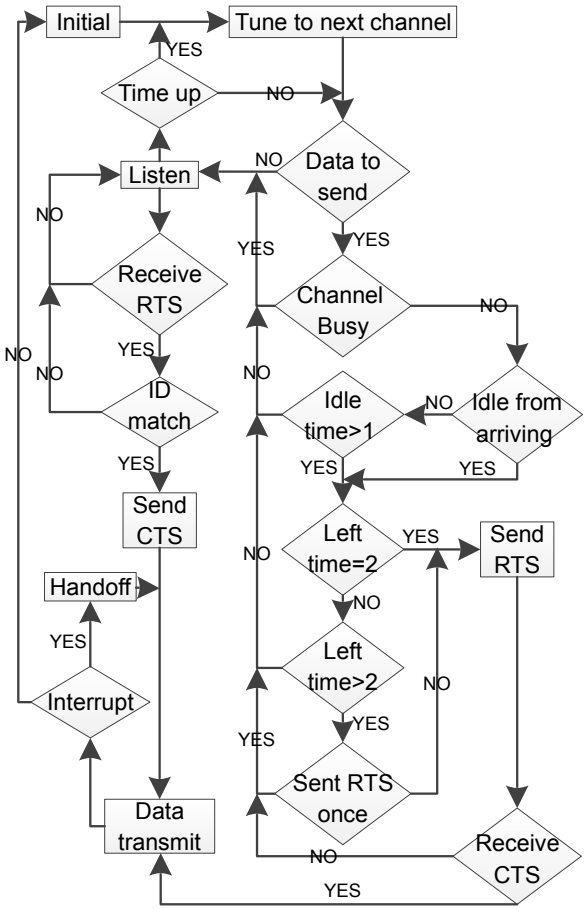


Figure 3.20: The flow chart of the proposed MAC protocol.

3.3.3 Analytical Model for the Optimal Time Slot

When considering the whole rendezvous process, many time slots are needed before a pair of SUs hop on a same channel and have a successful handshake. We denote P as the probability that a source SU successfully handshakes with its destination SU in the next time slot. Use $\overline{X}$ to represent the average service time (TTH) of a SU. Then,

$$\overline{X} = a \sum_{i=1}^{\infty} i(1-P)^{i-1}P = \frac{a}{P}. \tag{3.18}$$

From the analysis in Section 3.1, a long time slot can improve the successful handshake rate in one time slot. Thus, a and P are positive correlated. Therefore, an optimal a

is needed for (3.18) in terms of the shortest $\overline{X}$. We derive the optimal a as follows.

First, let ρ represent the **active rate** of a SU, i.e., the probability that a SU is in an active mode. Assume that the data traffic is homogeneous in the secondary network, i.e., the active rate of a SU is the same everywhere in the network. Let λ be the average packet arrival rate of a SU (in the unit of one RTS sending time). Then, $\rho = \lambda\overline{X}$.

We further denote P_0 as the probability that a source SU successfully hops on a same channel with its destination SU in a time slot. P_0 varies under different CH designs [75]. If we do not consider the neighboring interference problem, we can derive P as P_I (P in an idle environment):

$$P_I = P_0(1-\rho)P_1 + P_0\rho P_4, \tag{3.19}$$

where P_1 and P_4 are the same probabilities defined in Section 3.1 when the destination SU is passive or active, respectively.

However, the neighboring-inference problem cannot be ignored when a SU is in a dense network where the number of its neighbors is large. Assume that there are an average of K neighbors of a SU. Excluding the destination SU, the number of the potential contenders of a source SU is $K_1 = K - 1$. We denote K_2 as the average number of its hidden terminal SUs. Then,

$$\begin{aligned} P =& Pr(K_1 = 0,\ K_2 = 0)P_I \\ &+ Pr(K_1 = 1,\ K_2 = 0)P_2P_I \\ &+ Pr(K_1 = 0,\ K_2 = 1)P_3P_I + ... \end{aligned}$$

where $Pr(K_1 = 0,\ K_2 = 0)$ is the probability that no neighbor and hidden terminal exists on the same channel during the source SU's one time slot, i.e., $(1 - P_0\rho)^{K_1+K_2}$. We can further derive other probabilities regarding different values of K_1 and K_2. In

this way, P can be written as:

$$\begin{aligned} P \approx & (1-P_0\rho)^{K_1+K_2}P_I \\ & + \binom{K_1}{1} P_0\rho(1-P_0\rho)^{K_1+K_2-1}P_2P_I \\ & + \binom{K_2}{1} P_0\rho(1-P_0\rho)^{K_1+K_2-1}P_3P_I. \end{aligned} \tag{3.20}$$

We do not consider the cases when $K_1+K_2>1$ due to two reasons. One is that the probabilities of $K_1+K_2>1$ are negligible due to the $P_0\rho$ part. P_0 is usually on the order of $\frac{1}{N}$, where N is the total number of channels in a CRN. Meanwhile, ρ should be small enough in CRNs to avoid network congestion as analyzed in Section 2. Then, $P_0\rho$ is a quite small value. Moreover, the probability that $K_1+K_2 \geq 2$ involving $(P_0\rho)^2$ or higher power can be neglected. The other reason is that the probabilities of successful handshake under $K_1+K_2>1$ are also negligible, referring the derivation part of P_2 and P_3 in Section 3.1.

From (3.18)-(3.20), we can get

$$a \approx \overline{X}P_I(1-P_0\lambda\overline{X})^{K_1+K_2-1}\left[1-P_0\lambda\overline{X}(1-K_1P_2-K_2P_3)\right]. \tag{3.21}$$

It is a transcendental equation because of independent variables K_1 and K_2. Once the network parameters K_1, K_2, M, and λ are given, the expression of $\overline{X}$ in terms of a can be derived and the optimal a that minimizes $\overline{X}$ can be obtained.

3.3.4 Performance Evaluation

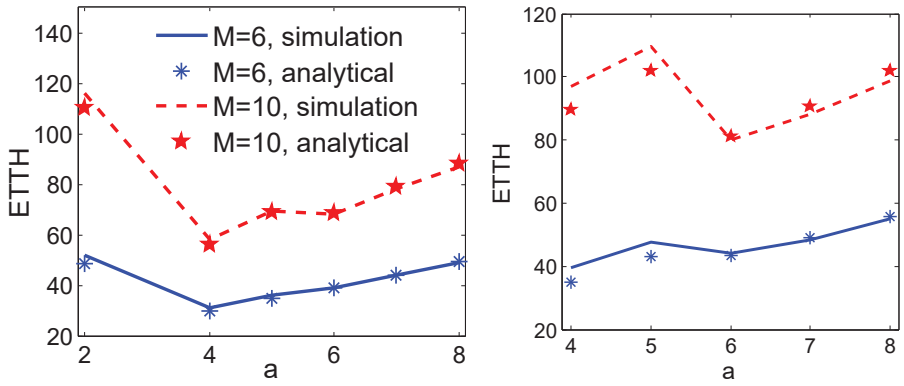
In this section, we evaluate the performance of our proposed MAC protocol under different scenarios by comparing simulation results with the analytical values. In our simulation, we assume that the packet arrival of each SU follows the Poisson distribution. Moreover, since P_0 is a variable independent of our analysis, we adopt the random CH algorithm under which P_0 is exactly $\frac{1}{M}$ in order to easily adjust the value of P_0. Additionally, we consider a grid network where $K_1=3$ and $K_2=3$. More importantly, each SU in the simulation has its own clock and is not required to

be synchronized with others. Other parameters used in our simulation are listed in Table 3.6.

Table 3.6: Simulation parameters for slot-asynchronous scenarios

Number of SUs	64 (8×8 grid)
Channel data rate	2 Mbps
The size of (RTS+CTS)	$160 + 112$ bits (802.11b/g)
Simulation time	10000

Figure 3.21 illustrates the ETTH of the whole network under different numbers of channels. The simulation results match the analytical results very well with a maximum difference of 5%. Figure 3.21a shows that $a = 4$ is the optimal size of a time slot when the average packet arrival rate is low, or, the active rate of a SU is low ($\lambda = 50$ pkt/s, $\rho \approx 0.1 - 0.2$). Since ρ is small, there are more idle time slots during rendezvous. Consequently, the advantages of a large a ($a > 4$) when dealing with complicated cases (P_2, P_3, and P_4) cannot be fully utilized. Therefore, the increasing rate of the ETTH after $a = 4$ is higher when there are more channels to hop ($M = 10$).



(a) $\lambda = 50$ pkt/s. (b) $\lambda = 100$ pkt/s.

Figure 3.21: ETTH vs. a in different traffic conditions.

Figure 3.21b shows the impact of different a in a nearly saturated network. When M is small, $a = 4$ still holds the optimal size of a time slot. However, note that the ETTH when $a = 6$ is already a bit lower than when $a = 5$. This means that the

advantages of a large a become dominant in the results. In the $M = 10$ case, the optimal size is when $a = 6$ (even when $a = 7$ has the same effect as when $a = 4$). Furthermore, the design of $a = 2$ cannot stand under this scenario. This is because when $a = 2$, the low probability of the handshake successful rate increases the TTH. Then, the long TTH leads to a high ρ which further results in $P_4 = 0$ and an infinite TTH. On the other hand, the improvement of each probability becomes less and less when a is larger than 6. It is also shown in Figure 3.21 and 3.22 that the ETTH monotonically increases after $a = 6$.

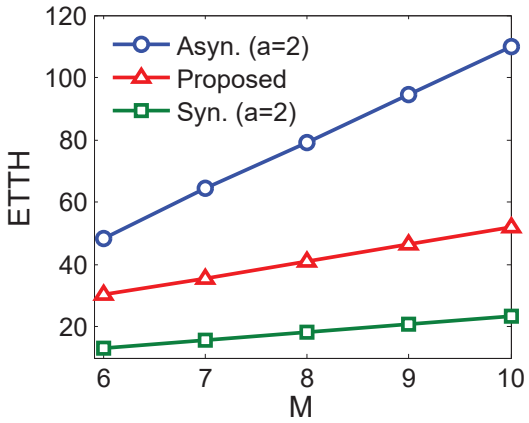


Figure 3.22: Compare with the MAC without our design in different scenarios.

Figure 3.22 compares the performance of different MAC protocols under the same traffic condition ($\lambda = 50$ pkt/s). Since we already derive the optimal size of a time slot under such scenario, the proposed line is the performance equipped with our MAC with $a = 4$ over different M. The asynchronous line belongs to the random CH protocol with the traditional MAC under asynchronous scenarios. The performance of this traditional MAC under the synchronous slot scenario is shown as the square-line. From Figure 3.22 we can see that the proposed MAC performs much better than traditional MAC and closer to the synchronous one (the ideal case).

To obtain the TTH in the unit of slots, we divide the minimum TTH using its corresponding a. For example, the minimum TTH for the case where $M = 10$ and $\lambda = 100$, is $80.99/6 = 13.50$ slots. Then, the average numbers of time slots a SU

Table 3.7: TTH vs. TTR

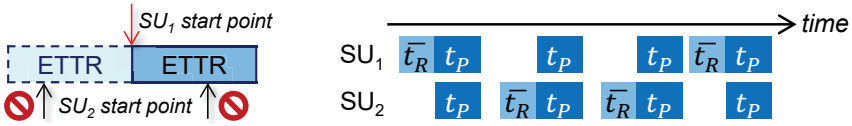
M (λ)	6 (50)	6 (100)	10 (50)	10 (100)
ETTH (in unit of slots)	7.75	8.95	14.32	13.50
ETTR (theoretical)	6	6	10	10

spent under different scenarios are shown in Table 3.7. It is shown that our proposed MAC protocol under slot-asynchronous scenarios can maintain the ETTH with the theoretical ETTR. Therefore, using our proposed protocol, it is not necessary to have slots synchronized in CRNs.

3.4 The Proposed Contention Window-based Deadlock-free Handshake Protocol

In this section, the challenge of deadlock-free blind rendezvous in CRNs is addressed for the first time. By analyzing the deadlock issues in both two-user and multi-user scenarios, we propose a novel MAC protocol with an optimal contention window size which can avoid deadlock and provide high network throughput. In addition, we also propose a probabilistic model for analyzing the network performance with our MAC. Simulation results validate our analytical model and demonstrate that our proposed protocol outperforms other possible attempts.

3.4.1 Deadlock Analysis



(a) deadlock prohibition (b) ideal scenario for network throughput

Figure 3.23: Analysis of deadlock in 2-SU scenario.

Deadlock in 2-SU CRNs: first, consider a system of two SUs. As explained in Figure 1.5, a deadlock forms as long as one SU turns to be a source SU while the other SU has already started rendezvous. In this case, without a proper MAC, both SUs will stay in the rendezvous state and cannot achieve a successful rendezvous. To achieve a successful rendezvous, one SU should not have traffic generated during the other's rendezvous time. As illustrated in Figure 3.23a, when SU_1 starts its rendezvous, if

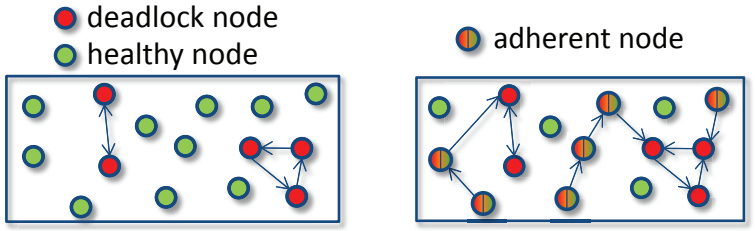
SU_2 also starts its rendezvous during the ETTR period either before or after the starting point of SU_1's rendezvous, this will lead to a deadlock.

Suppose that the ETTR is known which is usually between $O(N)$~$O(N^2)$. Then, the average rendezvous time in seconds is

$$\overline{t_R} = ETTR \times t_{slot}, \tag{3.22}$$

where $t_{slot} = \frac{\text{(RTS+CTS) bits}}{\text{Bandwidth}}$. Suppose the number of packet arrivals k in a given time t follows the Poisson distribution, $P_k(t) = \frac{(\lambda t)^k}{k!}e^{-\lambda t}$, where λ is the average arrival rate, the probability that a source SU can have a deadlock-free rendezvous is approximately $P_0(2\overline{t_R}) = e^{-2\lambda\overline{t_R}}$.

A failure rendezvous is hard to be confirmed (especially in multi-user cases) and thus no corresponding retransmission scheme exists so far. Hence, once a deadlock occurs, no more throughput can be generated. In such a system, to have network throughput in the ideal scenario as shown in Figure 3.23b, the probability is $e^{-n\lambda\overline{t_R}}$, where t_P in the figure represents data transmission time between the two SUs and n is the number of consecutive rendezvous between the two SUs. In other words, even if λ and N are moderate, the probability of deadlock, $1 - e^{-n\lambda\overline{t_R}}$, exponentially increases with time.



(a) formation (b) blackhole

Figure 3.24: Deadlock formation and its influence.

Deadlock in Multi-SU CRNs: when it comes to the multi-user CRN, more than one deadlock may be generated. In Figure 3.24a, 2-SU-deadlock and 3-SU-deadlock coexist, where SUs cannot contribute to the throughput. On the other hand, other

SUs who want to communicate with any of these deadlock SUs will also get trapped into a both-shouting situation. It is just like a black hole in the network. At the beginning, there might be only two SUs in a deadlock. However, other SUs will be adhered to the deadlock as long as they want to communicate with any of the deadlock SUs, which causes the black hole becoming bigger, as illustrated in Figure 3.24b. The bigger the black hole, the higher probability that destination SUs of the rest SUs are in the black hole. Consequently, the growing speed of the black hole becomes faster and faster. Finally, the black hole devours the whole network.

3.4.2 Protocol Details

Based on the CSMA/CA mechanism in 802.11, we propose a contention window-based deadlock-free MAC protocol, CWDF-MAC, which is presented in Figure 3.25. The definition of SIFS, RTS and CTS frames are the same as those in CSMA/CA in 802.11. However, according to the uniqueness of blind rendezvous in CRNs, CWDF-MAC has different features summarized as follows and in the corresponding protocol shown in Algorithm 4 (for convenience, the corresponding SIFS and propagation delay are not shown).

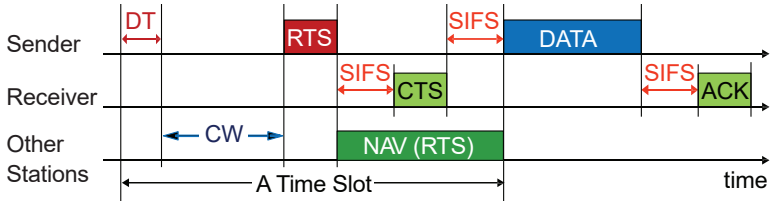


Figure 3.25: The diagram of the CWDF-MAC.

At the beginning of each time slot on a new channel, a detection time (DT) is employed to help a SU ensure the status of the channel not changed. Since a full spectrum sensing is only done in a periodic manner, an available channel may become unusable during a sensing period due to new transmissions of PUs or SUs in the same channel. The optimal duration of DT can be determined based on [76]. We regard the length of DT as a mini-slot.

After the DT, if the channel is detected busy, a SU sets its network allocation vector (NAV) value to be the rest time of the current time slot and sleep. After the DT, if the channel is confirmed idle, a source SU chooses a random number between 0 and CW (in terms of mini-slots) as its backoff time to avoid potential contentions on the same channel. Unlike 802.11, the size of the CW is fixed to maintain the constant length of time slots. Similarly, the binary-exponential-backoff retransmission mechanism is not practical for a slotted system. Even the RTS retransmission in one time slot is not wise. For example, if the destination user is not on the same channel, the retransmitted RTS in the same time slot can only decrease other contenders' sending chance and lead to unnecessarily long time slots. Thus, CWDF-MAC only offers one attempt for RTS sending in one time slot.

A SU does not sleep during the backoff period. Instead, it keeps monitoring the channel. Then, even if one source SU's backoff timer is one unit ahead of another source SU's, the later SU can detect the signal and suspend its own timer (this is why we set DT as the basic unit of mini-slots).

If the detected signal finally turns out to be an RTS message, the decoding SU may face two possible cases. One is that the decoding SU itself is the destination SU of the RTS, which will lead to a deadlock without our MAC. As illustrated in the left part of Figure 3.26, SU_C will buffer its own packet and send a CTS to rendezvous with SU_A. This action may help the network to have one more packet throughput and at the same time become less crowded because of one less source SU. Note that the data transmission can start immediately within the same time slot in order to increase the channel utilization before PUs' return.

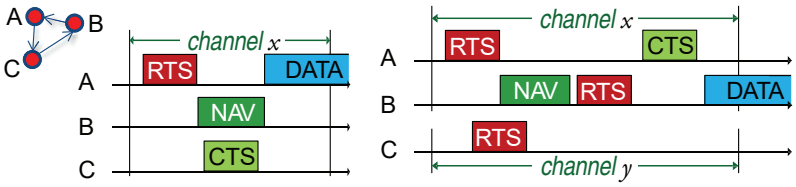


Figure 3.26: An example of CWDF-MAC.

If the RTS is aiming at another SU, the decoding SU sets two SIFS time and one CTS time as its NAV value (see Figure 3.25). This SU does not need to listen during the NAV period. On one hand, a CTS may or may not be sent depending on whether the destination SU is on the same channel. On the other hand, even if a CTS is sent, this SU may or may not hear this message depending on whether the destination SU is a hidden terminal of it. However, the SU can tell which case happens by sensing whether the channel is busy after the NAV period.

An RTS may not be decoded if there are multiple RTSs sent at the same time, or RTSs from hidden terminals collide. Under such circumstance, the decoding SU still sets the same NAV value.

After the NAV period, if the channel is still busy, it can be deferred that the RTS and CTS have been successfully exchanged and the corresponding source SU starts to transmit data, as shown in Figure 3.25. Then, this SU has to keep silent till the end of this time slot. On the other hand, if the channel is idle after the NAV period, this means a failed CTS receiving. If the left time is still enough for an RTS/CTS exchange, this source SU can send its own RTS. As illustrated in the right case in Figure 3.26, after a failure attempt of SU_A, SU_B still has the chance to rendezvous with SU_A, which avoid the deadlock in such scenarios.

3.4.3 Analytical Models for the Optimal CW

Due to the CW-based design, the CW has to be incorporated in every time slot which is not the rendezvous slot during a CH cycle. Thus, a long CW design increases the t_{slot} in Eq.(3.22) and then decreases the network throughput. An optimal CW size is derived in this section.

Consider two source SUs on the same channel in the same time slot. Under CWDF-MAC, each of them should generate a random number between 0 and CW. Each SU has equal probability to get a shorter backoff time than the other, excluding those cases when they generate a same number. Then, the probability that a chosen SU

Algorithm 4: The CWDF-MAC protocol for a source SU

Require: t_{DT}, t_{RTS}, t_{CTS}, and t_{CW};
1: CH to a new channel. Set $t = 0,\ t_{NAV} = 0$;
2: **if** $t \leq t_{DT}$ **then**
    if (*detects channel busy*) $t_{NAV} = t_{slot} - 1$;
    if (*detects channel idle*) $t_{backoff} = t_{DT} + rand(t_{CW})$;
3: **if** $t_{DT} < t \leq t_{backoff}$ **then**
    if *detects signal* **then**
        keep listening for t_{RTS};
        if *right RTS* **then**
            Buffer current packet;
            Send CTS; Prepare to receive data;
        else $t_{NAV} = t + t_{CTS}$;
4: **if** $(t_{NAV} \leq t_{backoff} < t)$ send RTS;
5: **if** $t_{backoff} \leq t_{NAV} < t \leq t_{DT} + t_{CW}$ **then**
    send RTS;
6: **if** *all other situations* **then**
    Set NAV until the end of current time slot;

finishes backoff prior than the other SU is

$$P(\text{first send}) = \frac{CW^2 - CW}{2CW^2} = \frac{CW - 1}{2CW}. \tag{3.23}$$

Next, we derive the ETTR of CH under CWDF-MAC. Since we focus on the analysis of the MAC design, we employ a simple yet effective random CH (RCH) algorithm for our analysis. The successful rendezvous probability in each time slot under the RCH algorithm is $P_0 = \frac{1}{N}$ [75]. Let P_s and P_l be the probabilities that the destination SU is a source SU or a listening SU on the rendezvous channel, respectively. Thus, under CWDF-MAC, the probability that a chosen source SU can successfully rendezvous in a time slot without role-preassignment is

$$P_0' = P_0 P_s P(\text{first send}) + P_0 P_l = \frac{P_s(CW - 1)}{2N \cdot CW} + \frac{P_l}{N}. \tag{3.24}$$

Then, the ETTR in our protocol, denoted as $\overline{X}$, can be derived as

$$\overline{X} = \sum_{i=1}^{\infty}(1 - P_0')^{i-1}P_0' = \frac{2N \cdot CW}{(P_s + 2P_l)CW - P_s}. \tag{3.25}$$

We then get the ETTR in terms of seconds with our new t_{slot}. As shown in Figure 3.25, we have

$$t_{slot} = \frac{\text{(DT+CW+RTS+CTS+2SIFS) bits}}{\text{Bandwidth}} + 2\alpha, \tag{3.26}$$

where α is the propagation delay. For the convenience in the following analysis, we ignore SIFS and α which are negligible as compared with other time durations. Meanwhile, assume that the size of an RTS and CTS is k times longer than that of the DT. Then t_{slot} can be expressed as $t_{slot} = (CW + k + 1)t_m$, where t_m is the length of a mini slot in seconds. According to (3.22),

$$\overline{t_R} = \frac{2N \cdot CW(CW + k + 1)t_m}{(P_s + 2P_l)CW - P_s}. \tag{3.27}$$

From (3.27), when $CW = 1$ (without our MAC), $\overline{t_R} = \frac{N(k+2)t_m}{P_l}$. When a deadlock occurs or in a saturated network, $P_l = 0$ leads to endless CH time, which agrees with our analysis. However, when $P_l = 0$ with our MAC ($CW > 1$), $\overline{t_R}$ has a finite value once a CW is selected. Let t_P be the packet transmission time. We treat $\overline{t_R} + t_P$ as the average service time of each packet. Then, each SU can be considered as a $M/D/1$ system based on queuing theory [29]. Thus, a SU under our protocol can still have a steady normalized throughput even in a saturated CRN, denoted as Γ_{th}, which is the lower bound of the network throughput. Since the size of CW is a predefined parameter in our protocol, we aim to find its optimal value in terms of maximizing Γ_{th}.

Throughput in 2-SU CRNs: in a saturated 2-SU scenario, a SU is either in

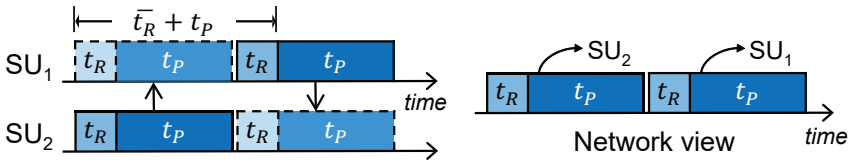


Figure 3.27: The analytical model for 2-SU CRNs.

a rendezvous state or in a packet transmission/receiving state. Since there are only two SUs in the network, they must be in the same state at the same time. As shown in Figure 3.27, when each of them rendezvous, the other one must also be in the rendezvous state. Thus, $P_s(\text{2-user}) = 1$.

From the view of the network as shown in Figure 3.27, the rendezvous period and the transmission period alternately take place along the time line. Since the successful rendezvous probability is the same for both SUs ($P(\text{first send})$), during each *network rendezvous period* t_R, either SU_1 successfully rendezvous or SU_2 does. Thus, $t_R = \frac{1}{2}\overline{t_R}$ ($\overline{t_R}$ is calculated from the SU's view). Meanwhile, SU_1 and SU_2 should have the same amount of throughput on average. Therefore, we have

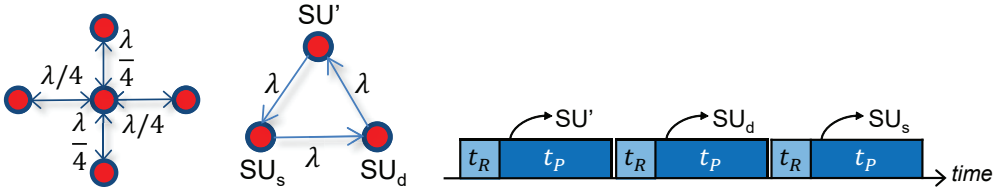
$$2\lambda\Gamma_{th}(t_R + t_P) = \lambda\Gamma_{th}(\overline{t_R} + 2t_P) = 1. \tag{3.28}$$

Then, the optimal CW value can be derived by solving the following optimization:

$$\begin{aligned} &\underset{CW}{\text{Maximize}} \quad \Gamma_{th} \\ &\text{subject to} \quad P_l = 0, P_s = 1, (3.27), \text{and } (3.28). \end{aligned} \tag{3.29}$$

Throughput in Multi-SU CRNs: it is difficult to precisely model the throughput in the multi-user scenario due to the topology changing in two dimensions independently (i.e., the pairing space and the channel space). However, for a chosen SU, we can classify its traffic into the inner traffic λ and the outside traffic λ'. Inner traffic includes the packets generated by a SU itself which need to be sent to other SUs and outside traffic includes the traffic received from other SUs. Since the desti-

nation SU of each packet is randomly assigned, the traffic to each neighboring SU is evenly distributed. In such a system, it is easy to infer that $\lambda = \lambda'$. As illustrated in Figure 3.28(a), suppose that each node has 4 neighbors within its transmission range on average. The outer traffic of each node is $4\frac{\lambda}{4} = \lambda$. Then, for a given SU, it is equivalent to regard all the outer traffic generated from one SU, say, SU'.



(a) traffic pattern (b) flow model (c) network view

Figure 3.28: The analytical model for multi-SU CRNs.

In this way, the pairing topology can be largely simplified. As shown in Figure 3.28(b), let $\mathrm{SU_s}$ be the source SU and $\mathrm{SU_d}$ represent all destination SUs of $\mathrm{SU_s}$. Then, for $\mathrm{SU_s}$, its inner traffic goes to $\mathrm{SU_d}$ and its outer traffic comes from SU'. Similarly, for $\mathrm{SU_d}$, it also has inner traffic to SU' representing all its destination SUs and outer traffic from $\mathrm{SU_s}$. Then, the network throughput can be modeled as shown in Figure 3.28(c).

Note that each SU has equal probability to rendezvous with its destination SU or be rendezvoused with its source SU, which is just like the 2-SU scenario. Thus, the rendezvous period t_R is also $\frac{1}{2}\overline{t_R}$. Therefore, the throughput of the whole network is

$$3\lambda\Gamma_{th}(t_R + t_P) = \lambda\Gamma_{th}(\frac{3}{2}\overline{t_R} + 3t_P) = 1. \tag{3.30}$$

Comparing (3.28) and (3.30), we can get

$$\Gamma_{th}(\text{multi-SU}) = \frac{2}{3}\Gamma_{th}(\text{2-SU}). \tag{3.31}$$

Thus, the optimal CW of 2-SU CRNs also holds the optimality for multi-SU scenarios when other parameters are the same.

Impact of a larger CW: when the CW is large, P_0' differs. As explained in Figure 3.26, a SU still has the chance to send its own RTS even after selecting a larger backoff time than other source SUs on the same channel, as long as the left time in the current time slot is long enough for another RTS and CTS exchange, i.e., $CW + k + 1 \leq 2(k+1)$, or, $CW > k$.

Suppose the backoff time expiration moment of other source SUs is earlier than the chosen SU. This moment can be selected from CW-1 mini slots. Among these selections, only those moments early enough so that the CW still opens even after $t_{RTS+NAV}$ are considered, which requires the moment to be selected within $CW - k$. Suppose that the average number of neighbors of each SU is m. The P(first send) under this circumstance should be replaced with

$$P(\text{RTS send}) = \frac{CW-1}{2CW} + \frac{CW-1}{2CW}\frac{CW-k}{CW-1}\frac{m-1}{m}, \tag{3.32}$$

where $\frac{m-1}{m}$ denotes the probability that the prior RTS is sent to SUs except the chosen SU.

Thus, if the CW size is allowed to be larger than $k+1$, its optimal value should be recalculated with condition (3.32) taken into (3.29).

3.4.4 Performance Evaluation

In this section, we evaluate the performance of our proposed MAC protocol under different scenarios by comparing simulation results with both the analytical values and other related protocols. Parameters used in our simulation are listed in Table 3.8. In our simulation, each PU is randomly assigned a channel when a new packet needs to be transmitted.

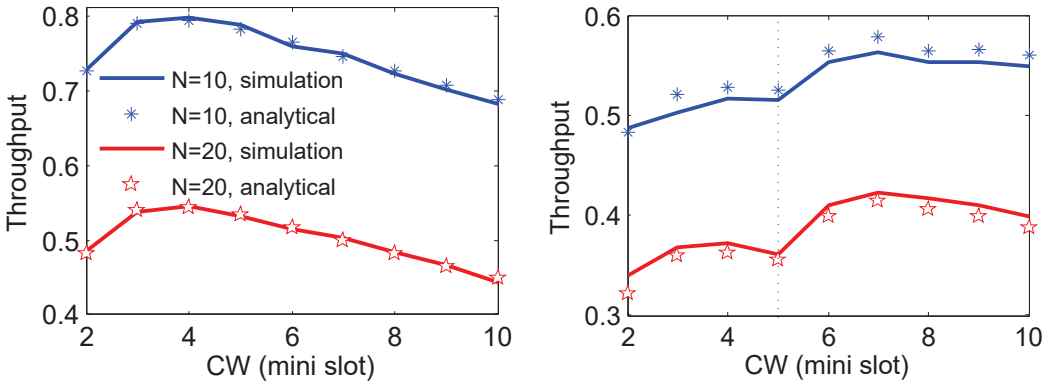
Figure 3.29 illustrates the normalized network throughput under different size of CW in two saturated CRNs (λ = 100 pkt/s). The simulation results match the analytical results very well with a maximum difference of 3%. Figure **??** shows that $CW = 4$ is the optimal size of the CW in terms of the highest normalized network throughput in 2-SU CRNs. When CW is large, the improving space of P(first send)

Table 3.8: Simulation parameters for CWDF

Simulation area	5×5
Simulation time	60000 time slots
SU sensing range	1
The number of PUs	50
Bandwidth	2 Mbps
The size of (RTS+CTS)	$160 + 112$ bits (802.11b/g)
The size of DT	54 bits
PU/SU packet size	1700 / 850 bytes
Average packet arrival rate of each PU	50 pkt/s

becomes smaller and smaller to approach its bound $\frac{1}{2}$. Instead, the impact of a larger time slot dominates the performance, which leads to the throughput decrease linearly.

On the other hand, in Figure **??**, the throughput pattern before $CW = k$ ($k = 5$ can be inferred from our simulation setting) is similar to that of the 2-SU scenario with a proportion about $\frac{2}{3}$, which agrees with the analysis. When $CW > k$, $CW = 7$ holds the optimality, which demonstrates the advantage of a larger CW in multi-user case. Note that the performance patterns when $N = 10$ and $N = 20$ are very similar in both 2-SU and 50-SU CRNs, which reflects the number of channels, N, is only a gaining factor.

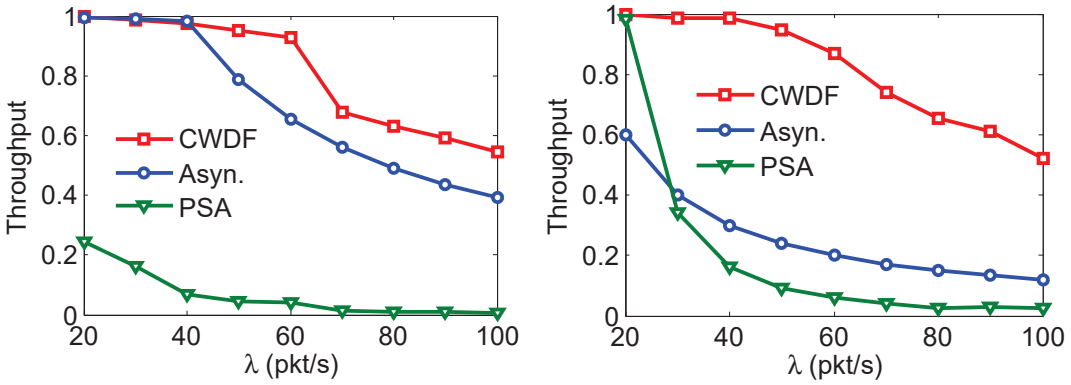


(a) 2-SU CRN (b) 50-SU CRN

Figure 3.29: Throughput vs. CW under different traffic conditions.

Figure 3.30 compares the performance of our proposed CWDF-MAC with other related protocols PSA-MAC [77] and Asyn.-MAC [74] under different traffic conditions. The performance of CWDF-MAC shown in the figure is our proposed MAC

with the optimal CW we already derived under saturated scenarios. From the figure we conclude that: 1) our protocol performs better than other MACs which has nearly 100% throughput under moderate traffic and at least 50% throughput under saturated traffic; 2) the network under our protocol can keep congestion-free (almost 100% throughput) under high traffic load. For example, the network congests when $\lambda = 70$ pkt/s in the 2-SU CRN under our protocol because the throughput begins to drop from 100%. Meanwhile, the network starts to saturate with low traffic load ($\lambda = 50$ pkt/s) under Asyn.-MAC and becomes saturated even easier in the multi-SU CRN ($\lambda = 20$ pkt/s); 3) when saturated, the throughput under our protocol linearly decreases with the increase of λ in both two networks, while exponentially decreases under other MAC in both networks; and 4) the Asyn.-MAC performs worse in the multi-SU scenario than in the 2-SU scenario, while CWDF-MAC can maintain its performance in a same level in multi-SU networks.



(a) 2-SU CRN, $N = 20$ pkt/s. (b) 50-SU CRN, $N = 10$ pkt/s.

Figure 3.30: Comparison with other deadlock-free MACs in different scenarios.

After we obtain the throughput from our simulation, the network rendezvous time t_R can be derived. To obtain the ETTR in the unit of slots, we divide $\overline{t_R}$ by its corresponding t_{slot}. For example, the network ETTR for the case where $N = 20$ in the 50-SU CRN is $0.0087/((7+6) * t_m) = 24.62$ slots. Then, the network ETTR under different scenarios is shown in Table 3.9. It is shown that CWDF-MAC under practical scenarios can maintain the ETTR similar to the role-preassigned ETTR

which is the theoretical ideal value. Therefore, our proposed MAC protocol does not affect the performance of the CH algorithm, but can eliminate the deadlock at the same time.

Table 3.9: CWDF-MAC vs. role-preassigned

CRN (N)	2 (10)	2 (20)	50 (10)	50 (20)
ETTR (CWDF-MAC)	10.48	20.92	13.85	24.62
ETTR (role-preassigned)	10	20	10	20

3.5 The Proposed Packet Fragmentation MAC Protocol

In this section, by mathematically modeling these impacts and dynamically mining the related parameters, we propose a self-adaptive protocol guiding the SU to derive the up-to-date optimal packet fragmentation. The proposed protocol is based on practical assumptions and taking other necessary CR functions into account such as spectrum sensing, channel hopping, and spectrum handoff. Simulation results validate our probabilistic model and the optimality of the fragmentation we derived. To the best of our knowledge, this is the first practical fragmentation protocol for multi-channel CRNs.

3.5.1 Retransmission Model

In a short term, each SU has k PUs on average randomly distributed within its transmission range r. Without loss of generality, suppose r is also the transmission range of a PU. Then, if a SU and any neighboring PU are both active on the same channel, their transmissions interfere with each other. We assume that the PU traffic follows the Poisson distribution in a short term with the average packet arrival rate λ_P which is homogeneous for all neighboring PUs of a given SU. The PU packet size follows an arbitrary probability distribution with the average length L_P. Each PU is randomly assigned a channel not occupied by other PUs concurrently like 2G/3G cellular networks. The time-varying parameters such as k, λ_P, and L_P differentiate our practical assumptions with the dedicated parameters assumed in most papers.

Other important notations used in the following are summarized in Table 3.10.

Table 3.10: Notations used in the retransmission model.

m	The number of busy channels from spectrum sensing
k	The number of neighboring PUs of a SU
B	The data rate of the channel
t_d	The spectrum handoff delay
λ_P/λ_S	Average packet arrival rate of the PU/SU
r	The radius of the sensing area of the SU
$\bar{v}$	The average relative velocity of a SU with respect to PUs
L_P/L_S	The length of the PU/SU packet
c	The optimal number of SU packet segmentation
l	The length of the segmented SU packet
h	The length of the header and trailer in the SU frame
q	The average channel switching times during handoff
$H(l)$	The handoff occurrence probability in terms of l
$T(l)$	The average service time of a frame in terms of l
$X(L_S)$	The average service time of a packet in terms of L_S
$\Gamma(L_S)$	Normalized SU throughput in terms of λ_S

Communication Steps: before SUs set up their communications, spectrum sensing and channel hopping are performed by each SU individually. After rendezvous, the SU pair exchange their channel information to form their common available channel set. If later the communication is interrupted by PUs, they can hop on to their next common available channel and resume the transmission.

Packet Fragmentation: each time when a SU has a string of data (unpacked from higher layer packet) to transmit, the SU MAC splits the packet into c equilong pieces. Then, each piece is added with a header and trailer to form a frame to transmit independently. At the end of a frame, if a SU frame does not collide with a PU packet, the SU transmitter continues to transmit the following frames on the same channel until all frames are successfully transmitted. Therefore, if a SU packet collides with a PU packet, only the collided frame needs to be retransmitted. As for the long PU packet, we assume that it is already fragmented by PU MAC based on the BER control. Since PU packet owns the priority of the assigned channel, its frames (including the retransmitted frames) are seamlessly transmitted in the given

channel, which can still be treated as an uncut PU packet in the analysis of SU frame transmission.

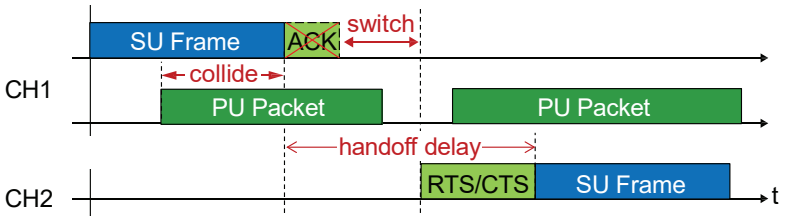


Figure 3.31: The spectrum handoff process.

Handoff Delay: Figure 3.31 shows the spectrum handoff process considered in this section. If a PU starts its packet transmission during a SU's frame transmission in the same channel (say, channel 1), the SU pair will know the failed transmission till the end of the frame (i.e., the receiver cannot decode the collided frame and the transmitter does not receive an acknowledgment (ACK) from the receiver). Then, the SU pair switches to their next common available channel (say, channel 2) for the retransmission of the previously unsuccessful frame. However, this new channel may already be unavailable due to the expired sensing information: i) at least one of the PU-neighbors of the SU pair reoccupied this channel after the rendezvous; and ii) SU-contention: other SU pairs nearby initiate the transmission on this channel before they arrive. Therefore, the CSMA mechanism is commonly employed to assist the handoff [78]. In other words, only after a successful RTS/CTS handshake on a new channel, can the SU pair finish the handoff process. Denote q as the average number of channel switching before handoff finishing ($q \geq 1$). The total handoff delay, t_d, can be calculated as:

$$t_d = t_{ACK} + q(t_{switch} + t_{RTS} + t_{CTS}) \tag{3.33}$$

where t_{switch} is the average operation delay for channel switching.

Performance Metric: from the perspective of the SU MAC layer design, the main tasks are the throughput increase and the congestion/delay control to support higher layers. For a SU with the average packet arrival rate λ_S and payload length L_S,

its congestion/busy ratio is $\rho_S = \lambda_S X(L_S)$ where $X(L_S)$ denotes the average packet service (transmission) time in terms of L_S. Meanwhile, the average good throughput for such a SU is $\Gamma(L_S) = \frac{L_S}{X(L_S)}$. As we can see, for a given L_S, a lower $X(L_S)$ can increase the throughput and at the same time better serve the congestion control by decreasing the busy rate.

On the other hand, $X(L_S)$ depends on its number of frames c, which is a tradeoff parameter. A higher c can decrease the retransmission rate for each relatively shorter frame but increase the overhead of both the total packet length and the operation times. Thus, one of the design goal is to get the optimal c for different L_S in terms of the minimum $X(L_S)$.

Derivation of the $X(L_S)$: denote l as the length of a frame, then $l = \frac{L_S}{c} + h$ where h is the combined header and trailer size. Since B is the data rate of the channel, the length of time needed to transmit one SU frame without retransmission is $\frac{l}{B}$. However, due to the PU activities, the transmission of the SU frame may fail and the frame needs to be retransmitted multiple times before it is successfully received by the receiver, each time with a handoff performed for help. If the total number of transmissions for a SU frame to be successfully received is i, the total transmission time is $\frac{l}{B}i + t_d(i-1)$, where t_d is the handoff delay from Eq.(3.33). In addition, let $H(l)$ be the probability that a handoff performed during a frame transmission with length l, which is actually the retransmission probability. Then, the probability that a SU frame is transmitted i times is $H(l)^{i-1}(1 - H(l))$. Then, the transmission time of a frame is written as

$$\begin{aligned} T(l) =& \sum_{i=1}^{\infty} \left[\frac{l}{B} i + t_d(i-1) \right] H(l)^{i-1}(1 - H(l)) \\ =& \frac{1}{1 - H(l)} \frac{l}{B} + \frac{H(l)}{1 - H(l)} t_d. \end{aligned} \tag{3.34}$$

For an original data with size L_S, its service time is then

$$X(L_S) = cT(l) = cT(\frac{L_S}{c} + h). \tag{3.35}$$

3.5.2 Retransmission Analysis

In Eq.(3.34), $H(l)$ is an unknown function, the retransmission probability of a SU frame with length l, or, the interference probability during the transmission. For a given l, $H(l)$ is determined by the PU activities and the mobility of SUs and PUs. We derive the interference probabilistic model considering these factors in the following.

Analysis of One-PU-Neighbor Scenarios: first of all, we assume that there is averagely only one PU-neighbor of a SU. That is, $k = 1$. Then, there are two scenarios to be considered. On one hand, the PU is idle before the SU frame transmission. Then, the transmission will be interfered if the PU has packets arrival during $\frac{l}{Cp}$ and the channel assigned to the PU is the same channel the SU is using $(1/\binom{N}{1})$. Let $N(t)$ be the number of packet arrivals of the PU in time t. Then, the probability of n packet arrivals in time t is $Pr[N(t) = n]$, which is associated with the distribution of packets arrivals. For example, if the traffic of this PU follows Poisson distribution, $Pr[N(t) = n] = \frac{(\lambda_P t)^n}{n!} e^{-\lambda_P t}$. Without loss of generality, suppose the average PU packet size is larger than a SU frame size and all PU frames are consecutively transmitted on its chosen channel. Then, as long as the PU has at least one packet arrival together with above conditions, the SU frame will be interfered. Overall, the interference probability under PU-idle-scenario is:

$$H_{k=1}^{idle}(l) = \left(1 - Pr[N(\frac{l}{B}) = 0]\right) \frac{1}{N}. \tag{3.36}$$

On the other hand, if the PU is transmitting a packet before the SU frame transmission on some other channel, then two cases may take place: i) the SU finishes its transmission before the PU does, which is free of interference; and ii) the PU completes its current packet transmission before the SU frame, which need to be further discussed later. To derive the probability of the second case, we denote t_1 and t_2 as the transmission starting time of the PU and SU respectively with an illustration in Figure 3.32. The probability of the second case equals to the probability that PU

finishes earlier cases ($caseA : t_1 + L_P \leq t_2 + l$) among all the cases that the SU starts transmission in the middle of the PU's transmission ($caseB : t_1 \leq t_2 \leq t_1 + L_P$): $Pr[(A, B)|B] = \frac{l}{L_P}$.

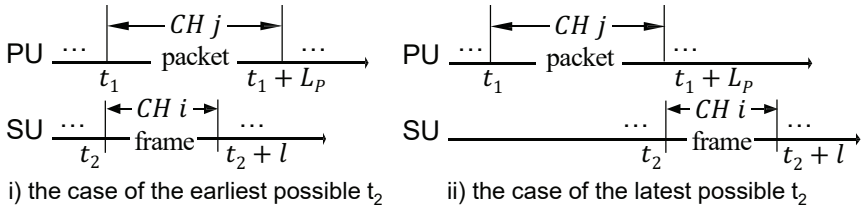


Figure 3.32: The cases that the PU is busy before SU frame transmission.

For the second case, there are further two inherent conditions which result different interference probabilities, as illustrated in Figure 3.33: i) if the PU still has packets in the queue waiting for the service (the PU congestion/busy ratio $\rho_P = \lambda_P \frac{L_P}{B}$), it will immediately starts another transmission. The new selected channel may also be the same channel that the SU is currently using ($1/\binom{N}{1}$); and ii) if the PU has zero packet waiting in its buffer ($1 - \rho_P$), it restores to the first scenario where the PU is idle at the beginning of the left transmission time of the SU frame. Since the average remaining time of the SU frame in such cases is $\frac{l}{2}$ (similar derivation as that of Figure 3.33, we can substitute it for $\frac{l}{B}$ in Eq.(3.36) to represent the interference probability under such cases. Finally, the interference probability under PU-busy-scenario can be written as:

$$H_{k=1}^{busy}(l) = \frac{l}{L_P}[\frac{\rho_P}{N} + (1 - \rho_P)(1 - Pr[N(\frac{l}{2B}) = 0])\frac{1}{N}]. \tag{3.37}$$

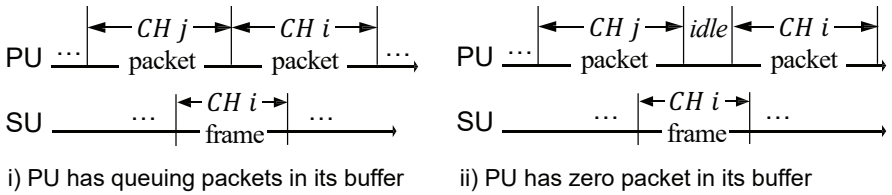


Figure 3.33: The cases that the PU finishes transmission before SU does.

Analysis of Multi-PU-Neighbor Scenarios: in the k PU-neighbors scenario,

suppose m PUs ($m \leq k$) are busy at the beginning of a SU frame transmission. Similar as the analysis in one-PU-neighbor scenario, we can elaborately derive the probability for each case. For example, the probability that i PUs among the $(k-m)$ idle PUs have traffic generated during the SU frame transmission is $(P_0[l/B])^{k-m-i}(1-P_0[l/B])^i$ where $P_0[l/B]$ is the short term of $Pr[N(l/B)=0]$. In addition, the probability that one of these re-active PUs choose the same channel with the SU is $\binom{N-1}{i-1}/\binom{N}{i} = \frac{i}{N}$. Then, the interference probability of the $(k-m)$ idle PUs can be derived as:

$$H_{k-m}^{idle}(l) = \sum_{i=1}^{k-m} \left(P_0[\frac{l}{B}]\right)^{k-m-i} \left(1-P_0[\frac{l}{B}]\right)^i \frac{i}{N}. \tag{3.38}$$

We can also derive the interference probability under the m busy PUs in the same way. However, in order to reduce the computational flexibility, the probability expression can be simplified to some extent with negligible difference. Consider the fact that cognitive radio technique is always used under the spectrum not fully utilized environment. That is, m and i are relatively much smaller than N. On the other hand, we know in mathematics, when N is much larger than i, $1-(\frac{N-1}{N})^i \approx \frac{i}{N}$. Therefore, we replace $\frac{i}{N}$ in Eq.(3.38) with $(1-(\frac{N-1}{N})^i)$ and the probability can be derived as a simpler form:

$$H_{k-m}^{idle}(l) \approx 1 - \left[1 - H_{k=1}^{idle}(l)\right]^{k-m}.$$

With the same revision of the probability under busy PUs, the total interference probability under (k,m) can be written as:

$$H_k^{static}(l) \approx 2 - \left[1 - H_{k=1}^{idle}(l)\right]^{k-m} - \left[1 - H_{k=1}^{busy}(l)\right]^{m}. \tag{3.39}$$

It is only the interference probability of the static CRNs and the total probability considered the nodes mobility is analyzed in the following.

Analysis of Mobility Scenarios: since PUs are evenly distributed in the system, the average number of PU neighbors (k) does not change within the moving duration of a SU. However, a new scenario may contribute to the interference probability

compared with the network with statistic nodes. As illustrated in Figure 3.34, when SU moves from location A to location B, it may encounter the new PU who is currently using the same channel with the SU (say, channel 6). The probability of such cases need to be added to the original $H(l)$.

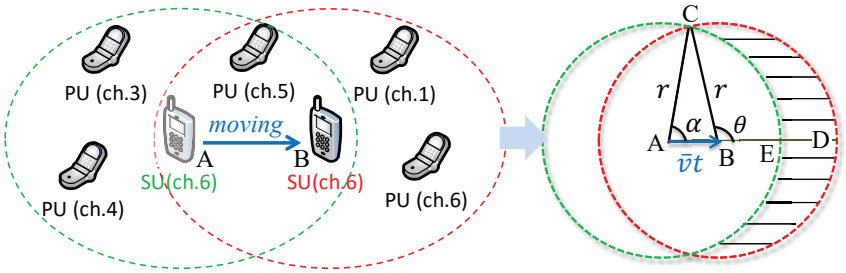


Figure 3.34: An interference example under mobile scenarios.

To derive this probability we denote k' as the number of new encountered PUs within the transmission range of a SU during its moving. The ratio of k' to k, is the same as the ratio of the crescent shadow area size (S_C) to the original circular area size (πr^2) in the right part of Figure 3.34. The circle represents the transmission range of a SU with the radius r. We assume that the speed of the SU is $\bar{v}$ which is a relative speed compared to surrounding nodes. The shadow part is the new transmission area during the nodes' moving. Note that the moving time during a frame transmission is $t = l/B$. Then, we derive S_C as a function of l.

Derivation of $S_C(l)$: $S_C = 2(S_{CBD} - S_{CBE})$. Meanwhile, we know $S_{CBE} = S_{CAE} - S_{CAB}$. Suppose r and $\bar{v}t$ are known, we derive $\alpha = arccos\frac{\bar{v}t}{2r}$ and $\theta = \pi - \alpha$. Then $S_{CAB} = \frac{\bar{v}t}{2} r sin\alpha$, $S_{CAE} = \pi r^2 \frac{\alpha}{2\pi}$ and $S_{CBD} = \pi r^2 \frac{\theta}{2\pi}$. Then, $S_C(l) = (\pi - 2\alpha)r^2 + \bar{v}tr sin\alpha$.

After calculating $S_C(l)$, $k'(l) = \frac{S_C(l)}{\pi r^2} k$. The probability that a new encountered PU is busy on the SU's transmission channels as $\rho_P \frac{1}{N}$. Therefore, the interference probability due to nodes' mobility is

$$H_k^{mobile}(l) = 1 - \left(1 - \rho_p \frac{1}{N}\right)^{k'(l)}. \tag{3.40}$$

Consequently, the total interference probability with the consideration of mobility is

$$H_k^{total}(l) = H_k^{static}(l) + H_k^{mobile}(l). \tag{3.41}$$

3.5.3 SAOF Protocol

Optimal Fragmentation: from Eq.(3.35), denote c_i as the optimal fragmented number for L_S^i. Followed the optimization problem:

$$\begin{aligned} \underset{c_i}{\text{Minimize}} \quad & c_i T(\frac{L_S^i}{c_i} + h) \\ \text{subject to} \quad & (3.33),\ (3.34),\ (3.36),\ (3.37),\ (3.39) - (3.41), \\ & c_i > 0,\ \text{and}\ c_i \in \mathbb{Z}. \end{aligned} \tag{3.42}$$

From the later protocol analysis, we know that $X(L_S^i)$ is only the function of c_i for a given L_S^i. If we assume $c_i{}' \in \mathbb{R}$, then the near-optimal $c_i{}'$ can be derived mathematically: in the final expression, the items including $c_i{}'$ in the power position can use Taylor expansion to approximate. Then the optimal value of $c_i{}'$ in the new expression can be calculated by the method of derivation. Due to the space limitation, the trivial derivational process is not given here. After calculating $c_i{}'$, the optimal c_i can be determined by comparing the nearest integer in terms of the minimum $X(L_S^i)$.

Then, we have $l_i = L_S^i/c_i$ where l_i is the corresponding frame size for L_S^i. Note that a global optimal frame size l does not exist. To prove it, we suppose there is a global optimal frame size l. Then the arrival/generated data length of a SU must satisfy $L_i = c_i l$. However, since c_i is an integer, L_i contradicts with the arbitrary-size assumption. Therefore, l_i is a local optimal size depending on the given L_S^i.

Protocol Details: Figure 3.35 is the complete block diagram of the proposed protocol. Each time when a data string with an arbitrary size L_S^i need to be transmitted (say, L_S^1), SAOF intelligently fragments it into equilong smaller frames with size l_i to get the maximum throughput. l_i is calculated by Eq.(3.42) with the pa-

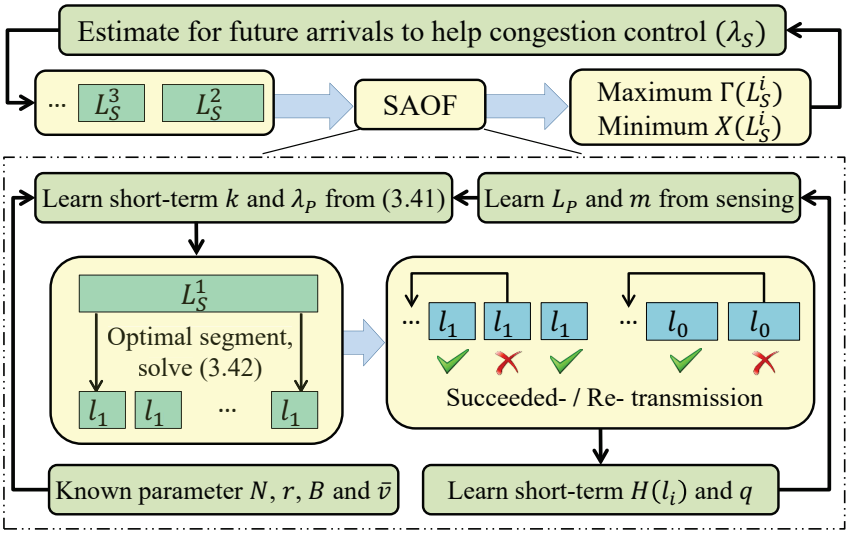


Figure 3.35: The block diagram of the proposed protocol.

rameter set $\{N, r, B, \bar{v}, q, m, L_P, k, H(l_i), \lambda_P\}$. Among these parameters: i) N, r, B and $\bar{v}$ is the known information from the system and the SU itself; ii) q is hard to mathematically derived but fortunately it is a independent parameter which can be counted and concluded through each frame transmission process; iii) a timely m can be directly obtained from the spectrum sensing since it equals to the number of unavailable channels; iv) L_P is straightforward to obtain from the sensing statistics since we only need the average values of the PU traffic information. In addition, if L_P also changes with time, the updated L_P can also be calculated from the short-term sensing history; v) similarly, an updated $H(l_i)$ can also be elaborately counted through the frame-transmission short-term history; and vi) the two primary network parameters k and λ_P cannot be obtained as easy as L_P. For k, the number of idle PUs cannot be detected through spectrum sensing. On the other hand, since neither k nor the channel selected by each PU on each transmission is known to the SU, λ_P cannot be inferred from the sensing. However, these two parameters can be learned by regression calculation from Eq.(3.41) since all the parameters required in Eq.(3.41) except k and λ_P are known or can be obtained as we claimed above. In fact, there should be enough simultaneous equations originated from Eq.(3.41) to derive k and

λ_P by recording different $H(l_i)$ values for different l_i.

Meanwhile, the calculated k and λ_P can on the other hand assist the SU estimate the minimum service time ($X(L_S^i)$) of possible L_S^i arrived in the near future (L_S^2, L_S^3, etc). Such information complements the SU to finish another task: congestion control of the packet arrival rate λ_S. At last, the protocol works in a dynamical way to keep mining and updating these time-varying parameters in order to better serve the SU transmission under the changing network environment.

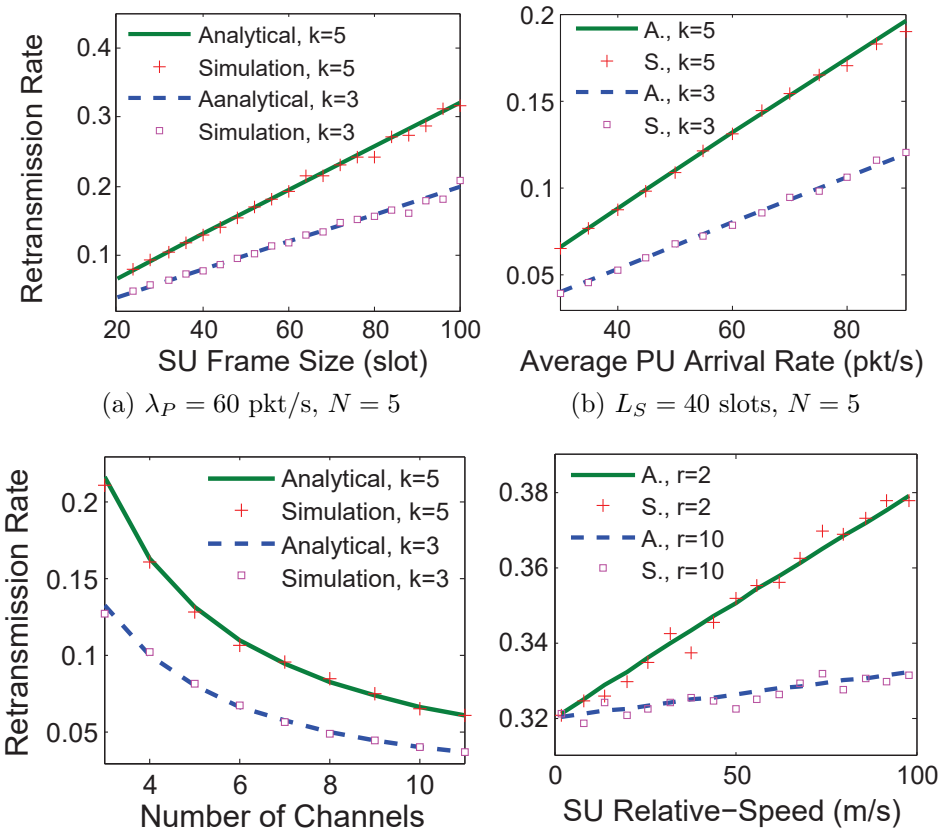
3.5.4 Performance Evaluation

In this section, we evaluate the throughput performance in SAOF. Firstly, we validate the proposed analytical models via extensive simulations. Then, we compare the network performance under SAOF with the scenario under various SU packet fragmentation. Finally, we compare the optimal fragmentation results under different SU packet size with varying primary environment. The default simulation parameters are summarized in Table 3.11 which mainly adopted from 802.11. In our system, one time slot equals to the transmission time of an ACK packet.

Table 3.11: Simulation parameters for SAOF

The radius of SU transmission range	10 m
The size of MAC (header+trailer)	30 + 4 Bytes
The average channel switching delay	100 μs
The size of (RTS+CTS)	20 + 14 Bytes
The size of a MAC ACK	14 Bytes
Channel data rate	2 Mbps
The average length of PU packets	100 slots

Figure 3.36 illustrates the impact of different parameters (l, λ_P, N, and $\bar{v}$ respectively) on SU frame transmission. From the results we can see that: i) the simulation and analytical results coincide very well, i.e., the simulation results validate the correctness of our retransmission/interference model $H(l)$; ii) a larger k always dramatically increases $H(l)$ under various conditions; iii) as shown in Figure 3.36a, the $H(l)$ increases with the frame length which motivates the design for the optimal



(a) $\lambda_P = 60$ pkt/s, $N = 5$

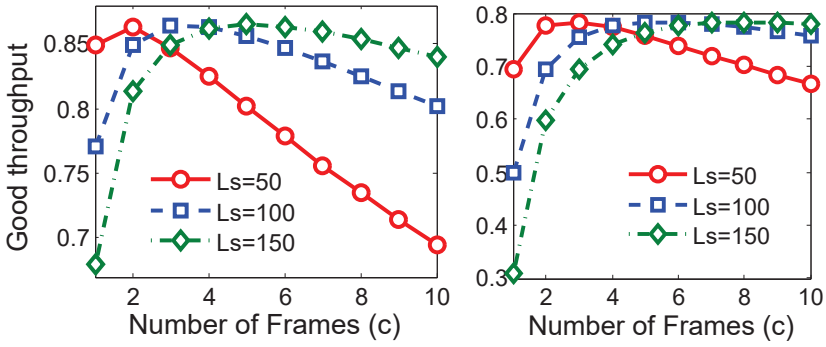
(b) $L_S = 40$ slots, $N = 5$

(c) $\lambda_P = 60$ pkt/s, $L_S = 50$ slots

(d) $L_S = 100$ slots, $k = 5$

Figure 3.36: $H(l)$ under different intra- and outer- conditions.

fragmentation; iv) it is observed in Figure 3.36b that λ_P is also a key factor together with k that can largely affect $H(l)$. Therefore, our protocol is highly required for mining the changing k and λ_P; and v) from Figure 3.36c and (d), fragmentation definitely needs to be performed when there are less channels and high $\frac{\bar{v}}{r}$ in the network due to the high $H(l)$.



(a) $\lambda_P = 60$ pkt/s, $k = 3$ (b) $\lambda_P = 90$ pkt/s, $k = 5$

Figure 3.37: Average SU packet throughput under different fragmentation.

Next, Figure 3.37 demonstrates the simulation results of the average SU throughput for a given packet under various fragmentation with different k and λ_P. It is illustrated that, for a given L_S, when the number of frames increases, the good throughput of the SU (as claimed in Section **??**, $\Gamma(L_S) = \frac{L_S}{X(L_S)}$) first increases and then decreases. Thus, there always exists an optimal fragmentation that maximizes $\Gamma(L_S)$ for a given L_S. Besides, it is observed that: i) under the same primary network, if each SU packet with whatever L_S takes its own optimal fragmentation, they can always achieve almost the same throughput (86.3% in Figure 3.37b and 78.3% in Figure 3.37a). Such equilong service rate of packets can further help to decrease the queuing delay for SU packets [29]; ii) for packet with a smaller payload ($L_S = 50$), more fragments beyond the optimal number can largely decrease the throughput due to the high ratio of the overhead (the header and the handoff delay) to the frame size; iii) except the fragmentation, the throughput is heavily influenced by the environment of the primary network. When SUs under a high-traffic dense-node network (Figure 3.37a), the throughput is degraded from that under a relatively sparse network (Figure 3.37b);

and iv) the throughput under the optimal c is much higher than that under no fragmentation ($c = 1$). Particularly, for packets with higher payload ($L_S = 150$), the prior throughput is almost 2.7 times than the latter one in the dense primary network.

Table 3.12: The optimal packet fragmentation

L_S (slot)	50		100		150	
Environment	sparse	dense	sparse	dense	sparse	dense
Optimal c	2	3	3	5	5	8
l (slot)	25	17	33	20	30	19

Finally, the optimal SU packet size for above scenarios calculated by our proposed protocol is given in Table 3.12. It is shown that: i) the optimal c derived by SAOF coincides with that observed in Figure 3.37, which validates the optimality of SAOF; ii) there is no universal optimal frame size for all L_S, which identifies SAOF as a practical protocol; and iii) even for a given L_S, its optimal fragmentation is different depends on the network environment, which enhances the necessity of SAOF's self-adaptive feature.

CHAPTER 4: PROPOSED CROSS-LAYER CD2DN RENDEZVOUS PROTOCOL

In this chapter, we propose a joint design of channel selection and channel hopping for guaranteed blind rendezvous, which is based on practical assumptions and worst-case scenarios. An analytical model of TTR is also proposed and validated against the simulation. For the first time, the TTR is significantly reduced to $O(1)$ with a low operation requirement. More importantly, under our proposed protocol, TTR decreases with the increasing number of available channels in the network. This is a very attractive feature in spectrum-under-utilized scenarios which has not been achieved by any existing CRN rendezvous work.

4.1 System Model and Parameter Analysis

The system considered in this chapter is basically same as the system in Chapter 3. Note that time-slotted synchronization is *not a necessary requirement*, because the listening SU in the proposed protocol stays on one channel most of the time. The important notations used in the following are summarized in Table 4.1.

Table 4.1: Notations used in the cross-layer rendezvous protocol

R_{i_ST}	The interfering range of a secondary sender (ST)
Γ_{sir}	The threshold of the signal-to-interference ratio
P_{r_PT}	The received signal power from primary senders (PT)
P_{i_ST}	The received interference power from STs
$P_{t\_PT/ST}$	The transmitting power of a PT/ST
D	The longest transmission distance between PUs
R_{i_SR}	The interference range of a secondary receiver (SR)
d	The distance between two SUs within one hop
R	The sensing range of a SU
P_{d_SU}	The SU's detection threshold of the received signal
$P_{r\_c_i}$	The received power on channel i
Γ	A SU's maximum interfering range to form SUBSET
d_r	The rendezvous range between two SUs in SUBSET
D_{ST}	The maximum one-hop distance between two SUs

4.1.1 Important Relationships

We first analyze the relationships that parameters must satisfy in our design. In [79], the relationship between a sender's transmission range and a receiver's interference range is studied in 802.11 ad hoc networks. Similar relationship in cooperative sensing in CRAHNs is used in [80]. Nevertheless, these relationships cannot satisfy the scenario discussed in this chapter, where we derive specific relationships of parameters in rendezvous scenarios in CRNs.

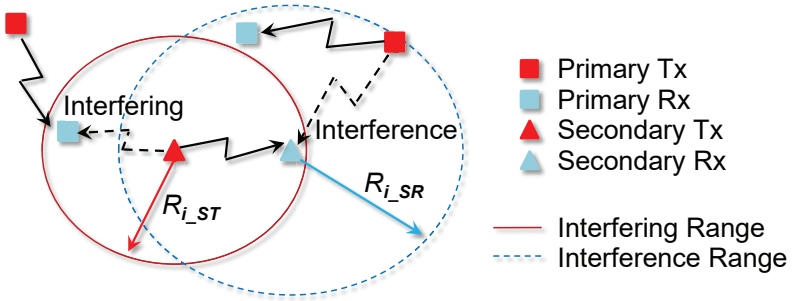


Figure 4.1: Two ranges of the rendezvous pair.

As mentioned in Chapter 1, the ACS of a SU depends on its interfering/interference range. In particular, we consider secondary transmitter (ST) interfering range and secondary receiver (SR) interference range, as illustrated in Figure 4.1.

Interfering Range: (R_{i_ST}) is the range centered at a source SU. Any PU receiver (PR) within this range has the possibility of failing to decode a packet sent from its PU transmitter (PT) due to the interference signal from the ST. In other words, for any PR outside this range, its SIR (the ratio of the received signal power from a PT to the received interference power from a ST) should be greater than a certain threshold (Γ_{sir_PU}) which is usually set to be 10 as in the 802.11b specification. Let P_{r_PT} be the received signal power from the PT and P_{i_ST} be the received interference power from the ST. Ignoring the additive white Gaussian noise, then,

$$\frac{P_{r\_PT}}{P_{i\_ST}} \geq \Gamma_{sir\_PU}. \tag{4.1}$$

According to the path loss model [81] commonly used in wireless networks, when a

transmitter propagates a signal to a receiver, the received signal power at the receiver is

$$P_r = P_t G_t G_r \frac{h_t^2 h_r^2}{d^\alpha},$$

where P_t is the transmission power, G_t and G_r are antenna gains of the transmitter and receiver, respectively, h_t and h_r are the height of the antennas, and d is the transmitter-receiver distance. α is the path loss exponent reflecting the signal attenuation rate which is equal to 4 in the two-ray ground reflection model and is equal to 2 within the Freznel zone.

About fading: since our design (explained in detail in the next section) only requires the existence of different receiving power from different channels, we do not need to establish a very accurate path loss model to estimate the exact location of each PU. Instead, only the closest possible PRs (worst-case) on each channel need to be estimated. Thus, we suppose that every received signal experiences a certain kind of fading, i.e., the received signal power can be represented as

$$P_r = \frac{kP_t}{d^\alpha}, \tag{4.2}$$

where k is a constant containing the average amplitude of fading.

Let P_{t_PT} and P_{t_ST} be the transmission power from the PT and ST, respectively. From Eq.(4.1) and Eq.(4.2), the relationship between $\boldsymbol{P_{t\_ST}}$ and $\boldsymbol{R_{i\_ST}}$ is given by

$$\frac{P_{r\_PT}}{P_{i\_ST}} = \frac{\frac{kP_{t\_PT}}{D^\alpha}}{\frac{kP_{t\_ST}}{R_{i\_ST}^\alpha}} = \frac{P_{t\_PT}}{\boldsymbol{P_{t\_ST}}} \left(\frac{\boldsymbol{R_{i\_ST}}}{D}\right)^\alpha \geq \Gamma_{sir\_PU}, \tag{4.3}$$

where D is the longest possible PT-PR distance (i.e., the radius of the PU's transmission range).

Interference Range: (R_{i_SR}) is the range centered at a SR. Any PT within this range may cause the SR failing to receive a packet correctly due to the PT's radio interference. Let Γ_{sir_SU} be the SIR threshold of a SU, P_{r_ST} and P_{i_PT} be the received signal power from the ST and PT, respectively, and d is the ST-SR distance. The

relationship between $\boldsymbol{P_{t\_ST}}$, $\boldsymbol{R_{i\_SR}}$, and $\boldsymbol{d}$ is:

$$\frac{P_{r\_ST}}{P_{i\_PT}} = \frac{\frac{kP_{t\_ST}}{d^{\alpha}}}{\frac{kP_{t\_PT}}{R_{i\_SR}^{\alpha}}} = \frac{\boldsymbol{P_{t\_ST}}}{P_{t\_PT}} \left(\frac{\boldsymbol{R_{i\_SR}}}{\boldsymbol{d}}\right)^{\alpha} \geq \Gamma_{sir\_SU}. \tag{4.4}$$

4.1.2 Practical Assumptions

In a CRN, the design of blind rendezvous should also consider the following practical issues.

Sensing resolution: due to the limitation of SU's antenna sensitivity, there is usually a sensing range (R) of a SU within which any PT can trigger SU's carrier sense detection. The detection threshold (P_{d_SU}) is thus given by

$$P_{d\_SU} = \frac{kP_{t\_PT}}{R^{\alpha}}. \tag{4.5}$$

Since the activities of PUs outside this sensing range cannot be detected by a SU, $R_{i\_ST} \leq R$ is required to avoid interfering possible PUs outside the sensing range. On the other hand, for a successful rendezvous, $P_{r\_ST} \geq P_{d\_SU}$ is necessary, which indicates

$$P_{t\_ST} \geq P_{t\_PT} \left(\frac{d}{R}\right)^{\alpha}. \tag{4.6}$$

Role-exchange problem: after the listening SU receives a RTS or a data packet on the rendezvous channel, a CTS or ACK is expected to send. Consequently, the listening SU also needs to consider the transmission interfering issue.

Let P'_{t_ST} be the transmission power of the listening SU who just received a correct RTS or data packet. Now, the interference range of this SU becomes its interfering range since the closest PU may be just outside this range. To avoid interfering this potential PU when sending CTS/ACK, by Eq.(4.3),

$$\frac{P_{t\_PT}}{P'_{t\_ST}} \left(\frac{R_{i\_SR}}{D}\right)^{\alpha} \geq \Gamma_{sir\_PU}. \tag{4.7}$$

Interference among SUs: the interference among SUs can be avoided by limiting SU's transmission power. For example, if a SU's maximum transmission power is

higher than a PU's transmission power (i.e., $P_{t\_ST}^{max} > P_{t\_PT}$), replace P_{t_PT} in Eq.(4.4) with $P_{t\_ST}^{max}$ and the interference range may not be able to prevent the interference from an outside ST. Hence, we should set $P_{t\_ST} \leq P_{t\_PT}$.

On SR's side, if its minimum received signal power is lower than that of a PU (i.e., $P_{r\_ST}^{min} < P_{r\_PT}$), replace P_{r_PT} in Eq.(4.3) with $P_{r\_ST}^{min}$ and the interfering range may still affect an outside SR's reception. Therefore, $\frac{kP_{t_PT}}{D^{\alpha}} \leq \frac{kP_{t_ST}}{d^{\alpha}}$is required.

Combining the above two requirements together, the transmission power of a SU should have the following constraint to avoid interference among SUs themselves,

$$\left(\frac{d}{D}\right)^{\alpha} \leq \frac{P_{t\_ST}}{P_{t\_PT}} \leq 1. \tag{4.8}$$

It is a practical setting since $d \leq D$ and $P_{t\_ST} \leq P_{t\_PT}$ are usually common assumptions in CRNs.

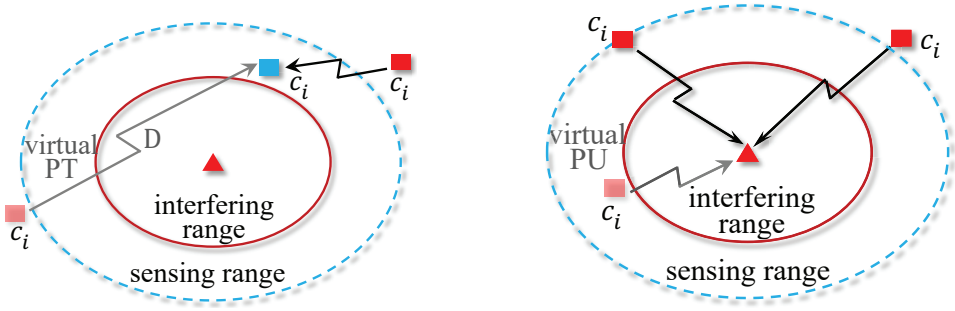
4.1.3 Worst-Case Derivations

When a source SU determines its interfering range R_{i_ST} on a channel i, $R_{i\_ST}^{c_i}$ should exclude the closest PU in order not to cause interference to any nearby PUs. However, the location of PUs are invisible to SUs and hard to estimated due to various reasons such as the hidden PR, PUs outside the sensing range, and the aggregated received signal. Sometimes all these factors affect together which makes the estimation impossible. In order to protect PUs from any possible interference, we propose the worst-case derivation to estimate the closest possible PU with existence of above factors.

Hidden PR derivation: a PR cannot be sensed immediately by the ST since it does not generate any power while receiving, which is the hidden primary receiver problem. Nevertheless, the ST can still recognize a PR indirectly. For example, if the primary network is in the half-duplex mode, a PU will send an acknowledge (ACK) back to its communication user after each reception. Hence, a hidden PR's location can be derived from a long-term sensing. In CRNs, the long-term sensing can be

substituted by mining a SU's own sensing history or cooperative sensing [80, 82]. If the primary network is in the full-duplex mode, the locations of the communication PU pair can also be inferred from their combined signals [83, 84]. Since we aim to avoid interfering any possible PR, we regard each detected PU as a potential PR.

Outside PT derivation: the signal from PT outside the sensing range of a SU cannot be detected, but may still exist. Under such scenario, one possible case is that such PT's destination PR is within the SU's sensing range. Though the PT is undetectable to the SU, the SU needs to control its transmitting power to avoid interfering the recipient of this PT. As shown in Figure 4.2a, on channel i, the SU estimates the location of a PU from its detection and sets the corresponding interfering range $R^{c_i}_{i\_ST}$. The detected PU is a recipient of another PU outside the SU's sensing range but close to the detected PU. For the SU, it needs to prepare the transmitting power for the worst-case: the invisible PU is just outside its sensing range and with D distance to the detected PU where the lowest possible signal power is received for the detected PU and the nearest possible PU outside the SU's sensing range. Thus, *the SU in our design always set a virtual PT for its detected PU whose location is described as above*. Then, the corresponding transmitting power derived by the SU can satisfy all other possible cases in such a scenario.



(a) outside PT (b) aggregated signal

Figure 4.2: Worst-case derivation under different scenarios.

Aggregated PU signal derivation: usually, the received signal on a given channel may be aggregated from several PUs. However, it is extremely difficult to differ-

entiate the number of surrounding PUs and the nearest one of them. Instead, the SU always suppose there is only one PU nearby on each channel. As shown in Figure 4.2b, the SU sets a virtual PU whose location can generate the same received signal power as the detected aggregated power. In other words, the virtual PU should always be closer to the SU than the nearest actual PU.

Let the *aggregated* received power at the SU from all PUs on c_i be $P_{r\_c_i}$ for the full-duplex mode. In the half-duplex mode, the maximum $P_{r\_c_i}$ from the ST's recent sensing history is chosen. For easy explanation, we use $P_{r\_c_i}$ uniformly for both modes. Then,

$$P_{r\_c_i} = \sum_{j=1}^{n} \frac{kP_{t\_PT}}{D_j^{\alpha}}, \tag{4.9}$$

where n is the number of PUs on c_i and D_j is their distance to the ST. The interfering range on this channel, $R_{i\_ST}^{c_i}$, should satisfy

$$R_{i\_ST}^{c_i} = Minimum\{D_1, D_2, ..., D_n\}, \quad n \geq 1.$$

Thus,

$$P_{r\_c_i} = \sum_{j=1}^{n} \frac{kP_{t\_PT}}{D_j^{\alpha}} \leq \frac{nkP_{t\_PT}}{(R_{i\_ST}^{c_i})^{\alpha}}$$

$$\Longrightarrow (R_{i\_ST}^{c_i})^{\alpha} \leq \frac{nkP_{t\_PT}}{P_{r\_c_i}}.$$

Consider the closest PR case: when $n = 1$, the relationship between $R_{i\_ST}^{c_i}$ and $P_{r\_c_i}$ is:

$$R_{i\_ST}^{c_i} = \left(\frac{kP_{t\_PT}}{P_{r\_c_i}}\right)^{\frac{1}{\alpha}}. \tag{4.10}$$

In this way, the SU further expands the margin of its interference-free guarantee to PUs.

4.2 Proposed SUBSET Design

In this section, we establish the design of SUBSET from channel selection to channel hopping. The main goal of SUBSET is to achieve blind rendezvous in CRNs as quickly

as possible in a energy saving manner.

Motivation: a short TTR of current CH algorithms is based on two requirements: more common available channels and less uncommon available channels in the ACSs of the rendezvous pair. Motivated by this observation, a fundamental step in our design is to build desirable ACSs for the CH algorithm, i.e., *the ACS of a listening SU is the largest possible subset of the ACS of a source SU.*

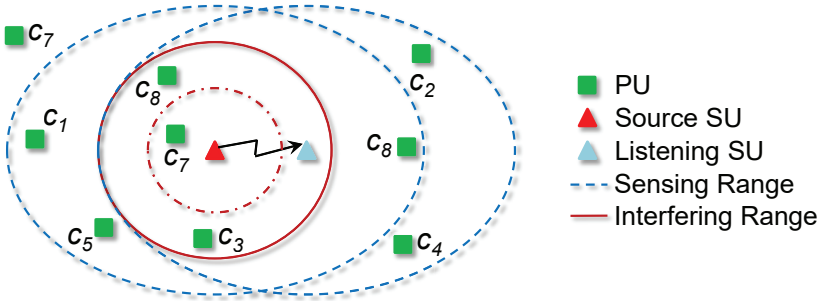


Figure 4.3: An illustration of desirable ACSs for the rendezvous pair.

Example: we illustrate the subset relationship in Figure 4.3. In existing CH-based rendezvous papers, both the source SU and the listening SU only use the channels not occupied by PUs for constructing the CH sequence. In practice, these are channels not being used in their sensing range. Thus, the ACSs of the source SU and the listening SU in Figure 4.3 are $\{c_2, c_4, c_6\}$ and $\{c_1, c_5, c_6\}$, respectively. They only have one common available channel c_6 and four uncommon available channels, c_2, c_4, c_1, and c_5.

However, in SUBSET, whether a channel is available depends on the interfering/interference range. In Figure 4.3, assume that the interference range is equal to the SU's sensing area. Then, the ACS of the listening SU is still $\{c_1, c_5, c_6\}$. In order to have this set be the source SU's largest possible subset, the interfering range of the source SU should be the red solid circle in Figure 4.3. Now, the ACS of the source SU is $\{c_1, c_2, c_4, c_5, c_6\}$, which makes all channels in the ACS of the listening SU, c_1, c_5, and c_6, common available channels. At the same time, this set has the

least number of uncommon available channels, c_2 and c_4. A smaller interfering range such as the dash-dot circle may harm the rendezvous by increasing the number of uncommon available channels and decreasing the reception power at the listening SU.

4.2.1 ACS Construction

Next, we show how to construct the desirable ACS with the subset relationship step by step.

ACS for the listening SU: for a listening SU, in order to generate minimum interference to all potential source SUs' signal, from (4.4), we prefer a listening SU to stay on the channel with the largest interference range ($R_{i\_SR}^{max}$). Based on the analysis in the role-exchange problem in the last section, after a packet is correctly received, the interference range will become the listening SU's interfering range which is R. Thus, we have $R_{i\_SR}^{max} = R$. Those channels sensed idle can thus be chosen into its ACS:

$$ACS_{listening} = \{c_i \mid P_{r\_c_i} \leq P_{d\_SU}, i = 1, 2, ..., N\}. \tag{4.11}$$

Transmission power of the listening SU: a listening SU sends a CTS with the maximum transmission power it can use, because the location of the source SU is unknown. Since the interfering range is R, based on (4.3), this transmission power (P'_{t_ST}) should be:

$$P'_{t\_ST} = \frac{P_{t\_PT}}{\Gamma_{sir\_PU}} \left(\frac{R}{D}\right)^{\alpha}, \tag{4.12}$$

which is the upper bound of the constraint (4.7).

ACS for the source SU: to ensure that the ACS of the listening SU is the largest possible subset of the ACS of the source SU, the source SU's minimum interfering range of the selected channel should be included in the interference range of the listening SU as shown in Figure 4.3.

Suppose that the distance between a source SU and its destination SU is $\boldsymbol{d}$. Then,

the source SU's maximum interfering range Γ should satisfy

$$\Gamma + \boldsymbol{d} = R. \tag{4.13}$$

Any idle channel or channel occupied by PUs located farther than this range Γ can be selected to the source SU's ACS. From the relationship in (4.10), this means that any channel with received power lower than P_γ can be selected, where

$$P_\gamma = \frac{kP_{t\_PT}}{(\Gamma)^\alpha}. \tag{4.14}$$

Then, the ACS for the source SU can be formed by:

$$ACS_{source} = \{c_i \mid P_{r\_ci} \leq P_\gamma, i = 1, 2, ..., N\}. \tag{4.15}$$

Transmission power of the source SU: for those channels with $R_{i\_ST}^{c_i} \geq \Gamma$, from (4.3), the maximum transmission power of the source SU on such channels should be:

$$P_{t\_ST}^{c_i} = \frac{P_{t\_PT}}{\Gamma_{sir\_PU}} \left(\frac{R_{i\_ST}^{c_i}}{D}\right)^\alpha. \tag{4.16}$$

However, since there is an interfering range limit to form the subset relationship (4.13), the source SU should only use the limit transmission power:

$$P_{t\_ST} = \frac{P_{t\_PT}}{\Gamma_{sir\_PU}} \left(\frac{\Gamma}{D}\right)^\alpha. \tag{4.17}$$

Derivation of the maximum d: so far, for a given $\boldsymbol{d}$, the source SU can form a desirable ACS from (4.15). However, when $\boldsymbol{d}$ becomes larger, the transmission power from (4.17) may not be enough for the distant listening SU to decode. To guarantee the rendezvous, we require an upper bound of $\boldsymbol{d}$ that can satisfy the relationship for decoding in (4.4) as well as other necessary constraints in Chapter 4.1. We list them as follows:

$$\begin{cases} P_{t\_ST}(d) = \dfrac{P_{t\_PT}}{\Gamma_{sir\_PU}} \left(\dfrac{R - \boldsymbol{d}}{D}\right)^\alpha \\ \dfrac{P_{t\_ST}^{min}}{\boldsymbol{d}^\alpha} \geq \Gamma_{sir\_SU} \dfrac{P_{t\_PT}}{R^\alpha} \\ P_{t\_ST}^{min} \geq P_{t\_PT} \left(\dfrac{\boldsymbol{d}}{D}\right)^\alpha \end{cases}. \tag{4.18}$$

Finally, we derive the upper bound of $\boldsymbol{d}$ as

$$d_r = \frac{R^2}{R + D(\Gamma_{sir\_SU}\Gamma_{sir\_PU})^{\frac{1}{\alpha}}} \tag{4.19}$$

with the condition

$$\frac{\Gamma_{sir\_SU}}{R^\alpha} \geq \frac{1}{D^\alpha}. \tag{4.20}$$

Or

$$d_r = \frac{R}{1 + (\Gamma_{sir\_PU})^{\frac{1}{\alpha}}} \tag{4.21}$$

with the condition

$$\frac{\Gamma_{sir\_SU}}{R^\alpha} \leq \frac{1}{D^\alpha}. \tag{4.22}$$

We denote this upper bound as the rendezvous range (d_r) for SUBSET. Since the source SU does not know how far away the destination SU is during rendezvous, it should limit its interfering range for the worst case $d = d_r$ in order to form the subset relationship with any listening SU that is located within d_r distance. Thus,

$$\Gamma = R - d_r; \tag{4.23}$$

Since homogeneous antennas are considered in this research, we assume that the sensing range (R) of SUs and PUs is the same and their SIR thresholds are also the same, i.e., $\Gamma_{sir\_PU} = \Gamma_{sir\_SU} = 10$. In 802.11 design, $R > D$ is required [85, 86]. Particularly, we adopt the default setting $R = 2.2D$ in ns-2 [87]. Thus, the left-hand side of both (4.20) and (4.22) is $\frac{10}{(2.2D)^\alpha}$. If $\alpha = 2$, (4.20) holds and $d_r = 0.4D$. Similarly, if $\alpha = 4$, (4.22) holds and $d_r = 0.8D$.

4.2.2 CH Algorithm

After forming such an ACS pair, we propose a specific CH algorithm which can further reduce the TTR. Algorithm 5 and Algorithm 6 give the pseudo code for the rendezvous pair respectively. For a source SU, it orders the channels in its ACS by their indexes (low to high) and hops on to them one by one with the transmitting

power $P_{t\_ST}(d_r)$ based on (4.18). The CH is performed in a cyclic way until rendezvous. For a receiver, it also arranges its ACS by the indexes of channels (low to high) and keeps staying on the first ordered channel until a correct RTS is received. If the first ordered channel becomes busy (i.e., a power above the detection threshold is sensed), it hops on to its next ordered channel and stays there.

Algorithm 5: The CH algorithm for the source SU

Require: ACS and P_{t_ST};
1: Order channels by their index: $\{c_{k1}, c_{k2}, ..., c_{km} | k1 < k2 < ... < km\}$;
2: $i = 1$;
3: Hop on to c_{ki} and send RTS with P_{t_ST};
4: **while** *not rendezvous* **do**
  $i = i + 1$;
  $j = ((i-1) \bmod m) + 1$;/* *next ordered channel* */ Hop on to c_{kj} and send RTS with P_{t_ST};

Algorithm 6: The CH algorithm for the listening SU

Require: ACS and P'_{t_ST};
1: Order channels by their index: $\{c_{k1}, c_{k2}, ..., c_{km} | k1 < k2 < ... < km\}$;
2: $i = 1$; $j = i$;
3: Stay on c_{ki};
4: **while** *no correct RTS is received* **do**
  if *not idle* **then**
    $i = i + 1$;
    $j = ((i-1) \bmod m) + 1$;
    Stay on c_{kj};
5: Send CTS on c_{kj} with P'_{t_ST};

Based on our CH design, the listening SU stays on the first ordered channel most of the time, which saves the energy consumption of SUs.

4.2.3 Long-Distance SUBSET

Motivation: we consider a special issue namely *long-distance rendezvous*. In this scenario, the one-hop range of SUs (D_{ST}) is longer than the required rendezvous range in SUBSET. Then, the distance (d) between the rendezvous pair may exceed

the rendezvous guaranteed range, $d_r < d < D_{ST}$. Note that to have a successful transmission, a minimum interfering range Γ is generated given a transmission distance d. From (4.4) and (4.5), $R_{i\_ST} \geq \frac{d}{R}\sqrt[\alpha]{\Gamma_{sir\_SU}\Gamma_{sir\_PU}}D$, which implies $\Gamma = R_{i\_ST}^{min} = 4.54d$ when $\alpha = 2$ or $1.44d$ when $\alpha = 4$ in our setting. Therefore, under $\alpha = 2$, when $d > 0.4D$, $d + \Gamma > 0.4D + 4.54 * 0.4D = 2.2D = R$. The subset relationship no longer exists and the rendezvous cannot be guaranteed. Similarly, $d_r < d < D_{ST}$ cannot guarantee the rendezvous under $\alpha = 4$.

Example: ss illustrated in Figure 4.4, if the rendezvous pair still follows Algorithm 5 and Algorithm 6, the rendezvous is unsuccessful: the listening SU keeps staying on c_1 and the source SU keeps hopping on c_2 and c_3.

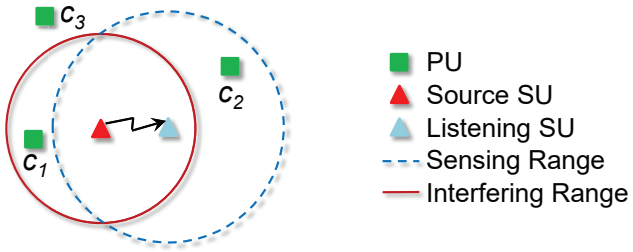


Figure 4.4: An illustration of long-distance rendezvous.

Algorithm: to solve this problem, we propose to modify Algorithm 5 for the source SU. Instead of only hopping onto the selected channels with fixed transmission power, the source SU sends an RTS with the maximum allowable transmission power, $P_{t\_ST}^{ci}$ in (4.16), on each channel in the network $\boldsymbol{C}$ including the "unavailable" channel for the source SU (e.g., c_1 in Figure 4.4). Thus, when it hops onto the channel where the listening SU stays (say, c_1), the transmission power required on this channel may not be enough for the listening SU to decode the RTS signal. However, the received power combined with the original undetected power (from PU on c_1) may be more than the detection threshold and trigger the listening SU's carrier sense. Then, according to Algorithm 6, the listening SU will choose the next ordered channel in its ACS to stay. As the process continues, the listening SU has to keep changing channels until the

channel it stays on can receive the correct RTS. The detailed process is described in Algorithm 7.

In this way, the rendezvous can be guaranteed as long as the rendezvous pair has at least one common available channel. For example, in Figure 4.4, by running the new algorithm, the source SU starts to hop from c_1. The listening SU detects that c_1 is not idle and chooses c_3 to stay. Finally, they will rendezvous on c_3.

New rendezvous range: we derive the new rendezvous range d'_r that can support the new rendezvous method. The worst case is that a PR is located at the edge of the listening SU's sensing range and meanwhile, it is the closest PU to the source SU. Under this circumstance, the minimum allowable interfering range of the source SU, $R_{i\_ST} = R - d$. Then, the corresponding transmission power should satisfy (4.3):

$$P_{t\_ST} \leq \frac{P_{t\_PT}}{10} \left(\frac{R-d}{D} \right)^{\alpha}. \tag{4.24}$$

On the other hand, the maximum allowable transmission power should be able to trigger the listening SU's carrier sense. From (4.6) and (4.24),

$$\frac{P_{t\_PT}}{10} \left(\frac{R-d}{D} \right)^{\alpha} \geq P_{t\_PT} \left(\frac{d}{R} \right)^{\alpha}. \tag{4.25}$$

Using the same setting $R = 2.2D$ in (4.25),

$$d \leq \frac{4.84D}{2.2 + 10^{\frac{1}{\alpha}}}.$$

$$\implies d'_r = \begin{cases} 0.9D, & \alpha = 2 \\ 1.2D, & \alpha = 4 \end{cases}. \tag{4.26}$$

Rendezvous guaranteed: note that in CRNs, when the transmission distance $d \geq 0.48D$ in the $\alpha = 2$ scenario, the interfering range exceeds the sensing range: $\Gamma = 4.54d \geq 4.54 \times 0.48D = R$. In other words, the interfere cannot be controlled since it may reach to those undetectable PUs. Thus, the one-hop distance $D_{ST} \leq 0.48D$ should be the default setting for all CRN designs. Under such circumstance, $d'_r = 0.9D > D_{ST}$ guarantee the rendezvous between SUs within one-hop distance.

Correspondingly, $D_{ST} \leq 1.53D$ is required in the $\alpha = 4$ scenario. On the other hand, $D_{ST} < D$ is also required in (4.8). Therefore, $d'_r = 1.2D > D_{ST} = min\{1.53D, D\}$ also support the one-hop rendezvous. Overall, the new design can satisfy one-hop long-distance rendezvous under both $\alpha = 2$ and $\alpha = 4$. It is worth to mention that *such inherent limitation of the one-hop range is valid in all secondary networks.* However, such basic feature is ignored in other CH efforts and not utilized the subset relationship it might brings.

4.2.4 Protocol Details

Algorithm 7: The SUBSET protocol for SU

Require: k, P_{t_PT}, D, R, D_{ST} and Γ_{sir};
1: **if** (*condition (4.20) holds*) Calculate d_r using (4.19);
2: **if** (*condition (4.22) holds*) Calculate d_r using (4.21);
3: Sense all channels and obtain P_{r_ci} $(i = 1, 2, ..., N)$;
4: **if** *source SU* **then**
  Calculate $P_{t\_ST}(d_r)$ using (4.18);
  Calculate P_γ using (4.14) and (4.23);
  if $D_{ST} \leq d_r$ **then**
    Calculate ACS using (4.15);
    Run Algorithm 5;
  if $D_{ST} \geq d_r$ **then** `/* long-distance */`
    $ACS = \{c_i, i = 1, 2, ..., N\}$; $i = 1$;
    Calculate $P^{ci}_{t\_ST}$ using (4.10) and (4.16);
    Hop on to c_i and send RTS with $P^{ci}_{t\_ST}$;
    while *not rendezvous* **do**
      $i = i + 1$; $j = ((i - 1) \bmod m) + 1$;
      Hop on to c_j and send RTS with $P^{cj}_{t\_ST}$;
5: **if** *listening SU* **then**
  Calculate P_{d_SU} using (4.5);
  Calculate ACS using(4.11) and P'_{t_ST} using (4.12);
  Run Algorithm 6;

Algorithm 7 gives the entire protocol showing our joint design of channel selection and channel hopping for both near-distance SUBSET and long-distance SUBSET. If the one-hop range (D_{ST}) is unknown, SUBSET by default uses the long-distance algorithm for the sake of rendezvous guarantee. Though the long-distance algorithm

can cover all possible cases, SUBSET prefers to use the near-distance algorithm if D_{ST} is known and $D_{ST} \leq d_r$ due to its faster rendezvous speed analyzed in the later section. As we can see, our proposed SUBSET is very easy for implementation, yet very efficient based on the TTR analysis in the next section.

4.3 TTR Analysis

In this section, we first propose two analytical models for calculating ETTR and MTTR of our CH algorithm with ACSs of *any* subset relationship. Then, we derive the ETTR and MTTR of our SUBSET protocol using these models.

4.3.1 Analytical Models

Let n be the number of channels in the source SU's ACS and m be the listening SU's. Since they share the subset relationship, $n \geq m$. Figure 4.5 illustrates a possible distribution of the paired ACSs. Assume that channels are already ordered by their indexes from low to high. The ACS of the listening SU (ACS_2) is a subset of m channels randomly chosen from the ACS of the source SU (ACS_1).

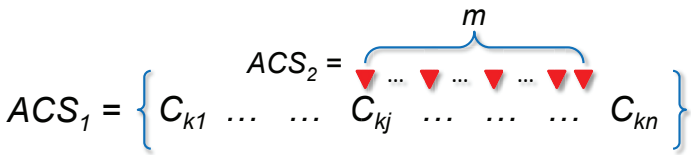


Figure 4.5: An illustration of subset distribution.

MTTR: the source SU will hop onto each channel in ACS_1 one by one until it rendezvous with the listening SU who is staying on its first channel in ACS_2. The TTR in such cases is when the source SU hops onto the same channel as the first channel in ACS_2, as shown in Figure 4.5. On the other hand, the latest possible first-channel in ACS_2 is the *(n-m+1)th* channel in ACS_1, i.e., the m channels in ACS_2 are exactly the last m channels in ACS_1. Therefore, in SUBSET,

$$MTTR(n, m) = n - m + 1 \tag{4.27}$$

which is *only related to* n *and* m.

ETTR: in order to derive the ETTR of our design, we first collect all possible cases after the channel selection process, or, all the subset cases from Figure 4.5. We can treat the channels in ACS_2 as m channels chosen from n channels in ACS_1. Then, the total number of such ACS-pair distribution is $\binom{n}{m}$. In each distribution case, by our CH algorithm, the TTR is deterministic.

Next, we calculate the probability of each possible TTR. If the first channel in ACS_2 is the *jth* channel in ACS_1 (C_{kj} in Figure 4.5), then $TTR = j$. To satisfy this requirement, other $(m-1)$ channels in ACS_2 can be chosen from the channels after C_{kj} in ACS_1. Then, the number of cases for $TTR = j$ equals to $\binom{n-j}{m-1}$. Let $P(j)$ be the probability of $TTR = j$. Consequently, $P_j = \binom{n-j}{m-1}/\binom{n}{m}$.

Finally, the ETTR in SUBSET can be expressed as

$$\begin{aligned} ETTR &= \sum_{j=1}^{n-m+1} jP(j) \\ &= \frac{\binom{n-1}{m-1} + 2\binom{n-2}{m-1} + \cdots + (n-m+1)\binom{m-1}{m-1}}{\binom{n}{m}}. \end{aligned} \tag{4.28}$$

To derive the final form, we first analyze a related expression:

$$f(k) = \binom{n-k}{m-1} + \cdots + \binom{m}{m-1} + \binom{m-1}{m-1}.$$

Using $\binom{m-1}{m-1} = \binom{m}{m}$ and the combinatorial law $\binom{p}{q} = \binom{p-1}{q-1} + \binom{p-1}{q}$ to the last two items, $\binom{m}{m-1} + \binom{m-1}{m-1} = \binom{m}{m-1} + \binom{m}{m} = \binom{m+1}{m}$. Then, keep applying the law to the last two items of the new formed equation iteratively, $f(k) = \binom{n-k}{m-1} + \cdots + \binom{m+1}{m-1} + \binom{m+1}{m} = \binom{n-k}{m-1} + \cdots + \binom{m+2}{m} = \cdots = \binom{n-k+1}{m}$. Thus, the numerator of (4.28) (NUM) can be rewritten as

$$\begin{aligned} NUM &= f(1) + f(2) + \cdots + f(n-m+1) \\ &= \binom{n}{m} + \binom{n-1}{m} + \cdots + \binom{m+1}{m} + \binom{m}{m}, \end{aligned}$$

which has the same pattern as $f(k)$. In fact, $NUM = f(1)|_{n=n+1,m=m+1}$. Using the

same method, we can derive $NUM = \binom{n+1}{m+1}$. Therefore,

$$ETTR(n,m) = \binom{n+1}{m+1} / \binom{n}{m} = \frac{n+1}{m+1}, \tag{4.29}$$

which also *only depends on two variables n and m.*

So far, we derive the formula for both ETTR and MTTR. The deduction and ratio parts are very powerful features which help ETTR and MTTR get rid of the positive relation with the size of the ACS pair. For example, when the source SU has 10 channels and the listening SU has 5 channels, the ETTR and MTTR are 1.83 and 6 time slot respectively. On the other hand, for the ACS pair (20,15), the (ETTR, MTTR) is (1.3,6) which is similar to the last case. Compared with latest CH algorithms which usually have about (5,10) in the prior case and (15,20) in the latter case, the rendezvous time is significantly reduced. Moreover, due to the independence feature of the formula, our CH algorithm can also be used independently for rendezvous in CRNs as long as the ACS pair following the subset relationship. Especially, after the network set-up period, nearby SUs may know the ACS information of each other. They can easily control their ACS to form the desired pair and use our algorithm to rendezvous. In other words, the proposed CH algorithm itself is easy to implement and is welcome in well developed CRNs.

4.3.2 Near-Distance SUBSET

From above models, in order to analyze the performance of SUBSET, we only need to know the average n and m in a CRN with SUBSET. Assume that PUs are evenly distributed. Denote K as the number of PUs in a unit area. If the active rate of a PU is ρ, then the average number of channels occupied by PUs in a listening SU's sensing range is $K\rho\pi R^2$. Then, the average number of available channels for the listening SU is $m = N - K\rho\pi R^2$. Using the same way, we can derive that $n = N - K\rho\pi\Gamma^2$. Therefore, referring (4.29) and (4.27), the estimations of ETTR and MTTR in the

normal case are:

$$\begin{cases} ETTR = \dfrac{N+1-K\rho\pi(R-d_r)^2}{N+1-K\rho\pi R^2} \\ MTTR = K\rho\pi d_r(2R-d_r)+1 \end{cases}, \tag{4.30}$$

both of which are $O(1)$ for N since d_r is independent of N and K from prior derivation. On the other hand, considering the impact of the number of PUs, ETTR is also $O(1)$ for K. MTTR is $O(K)$ which is also largely reduced compared with the latest effort ($O(K^2)$ in [31]).

Note that $\lim_{N\to\infty} ETTR = 1$, which means that ETTR approaches 1 when there are more channels in the network *no matter what the primary network condition is and what the PU and SU devices are* (K, ρ, R, and d_r). Therefore, SUBSET achieves the original intention of the CR technique, the rendezvous performs better in spectrum more underutilized networks.

4.3.3 Long-Distance SUBSET

In the long-distance design, the listening SU will eventually stay on a common available channel of both the source SU and the listening SU. Therefore, the TTR is the index number of their first common available channel since the source SU hops on every channel one by one. Thus, the problem is similar to the normal case shown in Figure 4.5, but with ACS_2 as the common available channel set and ACS_1 as the set of all channels.

In this way, we have $n = N$. Let m' be the number of channels in the common available channel set. Note that when $d \leq d_r$, $\Gamma + d \leq R$, which means that the rendezvous pair still shares the subset relationship and thus $m' = m_1 = N - K\rho\pi R^2$. When $d > d_r$, they do not own the subset relationship and the common available channels are those channels outside both the listening SU's sensing range and the source SU's interfering range: $m' = m_2 = N - K\rho S_U$, where S_U represents the size of the union area of the rendezvous pair. Using Figure 4.6, we first derive S_U as a function of their rendezvous distance d.

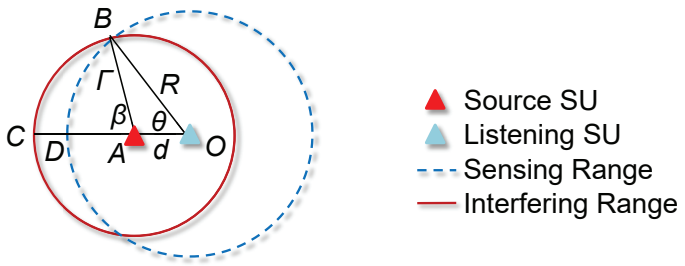


Figure 4.6: Changing interfering range with a known d.

Derivation of S_U: in order to derive S_U, we first need to know the size of the intersection area S_I in Figure 4.6. By observation, S_I is equal to the interfering area of the source SU minus the crescent area S_C: $S_I = \pi\Gamma^2 - S_C$, where $S_C = 2(S_{BAC} - S_{BAD})$. Meanwhile, we know $S_{BAD} = S_{BOD} - S_{BOA}$. Since R, d, and Γ are known ($\Gamma = 1.44d$ from the analysis in Section 2.3), the following parameters can be derived using law of sine and cosines: $\theta = cos^{-1}\frac{R^2+d^2-\Gamma^2}{2Rd}$, $\beta = \pi - cos^{-1}\frac{\Gamma^2+d^2-R^2}{2\Gamma d}$, and $S_{BOA} = \frac{1}{2}Rd\sin\theta$. With θ and β, we have $S_{BOD} = \pi R^2\frac{\theta}{2\pi}$ and $S_{BAC} = \pi\Gamma^2\frac{\beta}{2\pi}$. Finally, $S_U = \pi\Gamma^2 + \pi R^2 - S_I$ can be derived by substituting corresponding parameters with the known R and d.

After calculating m_2, the average number of channels in ACS_2 can be derived by: $\overline{m'} = \frac{1}{D_{ST}}\left(\int_0^{d_r} m_1 d(d) + \int_{d_r}^{D_{ST}} m_2(R,d)d(d)\right)$. Since d is independent of N, the ETTR can be derived in a similar way. Finally, the ETTR and MTTR for the long-distance SUBSET are

$$\begin{cases} ETTR_l = \dfrac{1}{D_{ST}}\left(\displaystyle\int_0^{d_r}\frac{N+1}{m1+1} + \int_{d_r}^{D_{ST}}\frac{N+1}{m2+1}\right) \\ MTTR_l = MTTR(N,m') = N - m' + 1 \end{cases}. \tag{4.31}$$

Further, since m_1 and m_2 only contain the first power of N, $ETTR_l$ is also $O(1)$. On the other hand, in order to get the maximum value of $MTTR_l$, m' should be as small as possible. The extreme value can be derived when the union area S_U is the largest, i.e., when $d = D_{ST}$. Replace m' in (4.31) with $N - K\rho S_U(R, D_{ST})$. Then, $MTTR_l = K\rho S_U(R, D_{ST}) + 1$, which also owns $O(1)$ for N and $O(K)$ for number of PUs. Therefore, *the performance of ETTR and MTTR in one-hop long-distance*

SUBSET is in the same level with the normal SUBSET. Meanwhile, compared (4.31) with (4.30), both $ETTR_l$ and $MTTR_l$ are relatively larger than the ETTR and MTTR in near-distance SUBSET. This also accords with the facts that TTR largely depends on the number of common available channels, since distant rendezvous-pair has less common available channels.

4.4 Application Analysis

Besides the practical assumptions and worst-case based design elaborated in Section 1, further analysis is needed when applying SUBSET to CRNs, since the information impacting our model may be changed throughout the whole network life. Generally, two issues need to be considered: 1) the spectrum sensing process is imperfect. There might be false information that impacts the ACS forming process; 2) the unknown information may become known to SUs which turns the blind rendezvous into partially visible rendezvous. Our SUBSET has inherent merits against these changes and can be easily adjust to adapt with them.

4.4.1 With False Information

The false alarm about the spectrum sensing reflects on two ways. One is that the SU senses a busy channel as available and put it into its ACS. The other one is vise versa, that the SU senses an idle SU as busy and excludes it from its ACS. In short, we treat the prior one as the (false) idle-alarm and the latter one as the (false) busy-alarm. The increased TTR caused by the false alarm for SUBSET and other CH schemes, especially in the worst case (MTTR), are concluded in Table 4.2. Compared with existing CH efforts, SUBSET can better deal with the sensing false from following perspectives.

First, the listening destination SU in SUBSET is exempt from idle-alarm. Note that our listening SU keeps staying on an available channel for a long time and listening for the potential source SU. If the idle-alarm channel is not the staying channel of

Table 4.2: Increased MTTR under one false-alarm channel

false alarm		idle-alarm	busy-alarm
Source SU	SUBSET	$+1$	$-1/+O(N)$
	Others	$+O(N^2)$	$+O(N^2)$
Destination SU	SUBSET	$+0$	$\times$
	Others	$+O(N^2)$	$\times$

the listening SU, it will not affect the TTR since the SU pair still rendezvous on the staying channel. If it is the staying channel, the listening SU will realize that it is actually a busy channel from the listening results. Consequently, the staying channel in SUBSET is always the true available channel. On the other hand, the destination SU in other CH schemes keeps hopping from one channel to another and has no way to tell the idle-alarm channel from other available channels, since both of them may not be the rendezvous channel. However, the idle-alarm channel does occupied a position in their CH sequence and thus may delay the potential rendezvous to several rounds later. In the worst case, it affects the rendezvous with MTTR delay $O(N^2)$.

Based on the same reason, the source SU in both SUBSET and other CH schemes cannot get rid of the idle-alarm. However in SUBSET, since the idle-alarm channel could not be the staying channel of the listening SU as mentioned above, the source SU can still rendezvous on such staying channel within its MTTR. The worst case is that the index of the idle-alarm channel is before that of the staying channel. Then, compared with the correct case, the source SU has to hop *one more channel* before the rendezvous. On the other hand, the idle-alarm channel may ruin the rendezvous sequence of the source SU in other schemes, which is similar to the destination SU analysis, and results in an unwanted rendezvous delay.

Last, neither the source SU nor the listening SU in SUBSET can be affected by the busy-alarm except one scenario: the channel excluded from one's (or both) ACS is the only common available channel of the rendezvous pair. Then, the rendezvous will become unsuccessful both in SUBSET and other CH schemes. In this case, if the

channel is false alarmed by the source SU, the SU have to figure out the rendezvous failure after MTTR period. It may rescan the spectrum to correct its ACS and resume the CH process. Due to the different MTTR level between SUBSET ($O(N)$) and other CH schemes ($O(N^2)$), the source SU in SUBSET can quicker response to this case. If the channel is false alarmed by the listening SU in this case, the rendezvous fails until the SU updates its spectrum sensing. In other busy-alarm scenario, the SU pair in SUBSET can rendezvous on other common available channels. Especially, when the source SU has the busy-alarm and there are other common available channels, the MTTR becomes even less since $n = n - 1$ in (4.27) compared with the correct information case.

4.4.2 With Known Information

As mentioned in Section 3.1, after the rendezvous setup period, SUs may know some information about their neighbors. For SUBSET, the most useful information is the distance between the rendezvous pair, d. With the known d, SU can further adjust its channel selection process and channel hopping strategy to fasten the rendezvous process. However, the known d does not show any influence to existing CH algorithms.

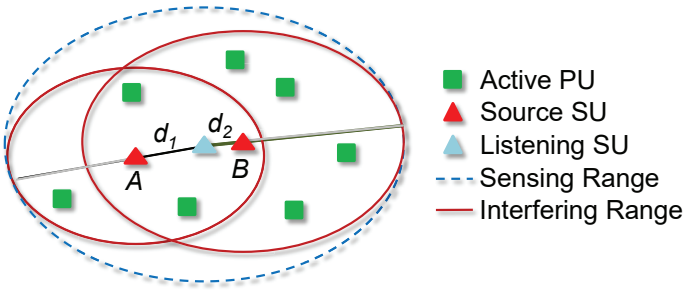


Figure 4.7: Channel selection adjusts for known d in near-SUBSET.

For $d \leq d_r$, the rendezvous pair is within the subset relationship and the near-SUBSET is used. When the source SU does not know the exact distance to its one-hop destination SU, in channel selection step, it uses the smallest possible Γ

generated by d_r ($\Gamma = R - d_r$) for the sake of interfere-free to PUs. For example, in Figure 4.7, let $d_1 = d_r$. Then, the corresponding Γ can satisfy the subset relationship with any destination SU within d_1. However, if the rendezvous distance is known to the source SU (say, d_2, and $d_2 < d_1$), then the same Γ will not fully utilize the subset relationship. In this case, $\Gamma = R - d_2$ can generate a larger interfering range, which makes the largest possible subset and thus reduces more unrelated channels for the source SU to hop. In other words, n in (4.27) and (4.29) becomes smaller and closer to m, which leads to smaller ETTR and MTTR. Therefore, the source SU will replace d_r with the known d in (4.23) to get the optimal Γ in terms of minimizing the rendezvous time.

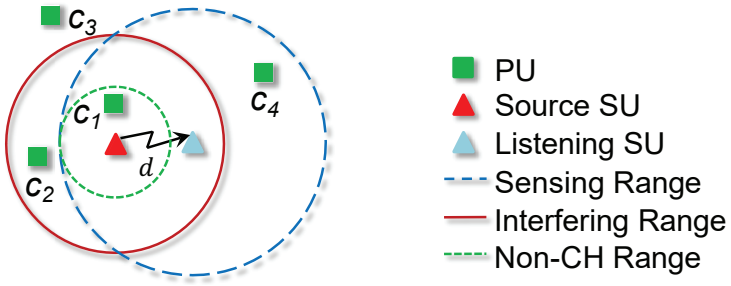


Figure 4.8: Channel hopping adjusts for the known d in distant-SUBSET.

For $d_r \leq d \leq d'_r$, the distant SUBSET is used. If d is unknown, the source SU has to hop every possible channel to avoid missing the staying channel of the destination SU. For example, in Figure 4.8, the ACS of the destination SU is $\{c_2, c_3\}$ and at the beginning it stays on c_3. The source SU hops from c_1. When it hops onto c_2, the aggregated signal power will push the destination SU to c_3 for staying. Then, they will successfully handshake on c_3 where the TTR is 3 time slot. However, if d is known, the source SU does not need to hop from the beginning, since from Figure 4.8 the source SU will derive that c_1 is definitely not the available channel of the destination SU. Then, it will hop from c_2 and the TTR will be reduced to 2 time slot. In this way, the source SU can always hop from the channels detected outside

the non-CH area $(R - d)$, within which the channel is used surely within the sensing range of the destination SU. Similarly, this CH strategy decreases n for the source SU, which proves that the ETTR and MTTR can always be further reduced with a known d.

4.5 Performance Evaluation

In our simulation, 1) PUs and SUs are evenly distributed in the simulation area; 2) Each PU is randomly assigned a channel when a new packet needs to be transmitted; 3) Packet arrivals follow the Poisson distribution; and 4) Each SU randomly chooses a SU within its transmission range as its destination SU when it has a new packet to transmit and becomes a source SU. The parameters used in our simulation are listed in Table 4.3.

Table 4.3: Simulation parameters for SUBSET

The antenna related constant	-25.54 dB
The minimum required SIR for PU/SU	10 dB
The path-loss factor α	2
The transmission power of a PU	2w
Side length of the simulation area L	500 m
Channel data rate	2 Mbps
PU/SU packet size	50 slots
The size of (RTS+CTS) (802.11 b/g)	$160 + 112$ bits
Simulation time	10000 slots

Figure 4.9 illustrates the spectrum condition detected by a potential rendezvous pair (d = 80m) in a moment during simulation. If there are only five channels in a primary network, as shown in the top figure, under the SUBSET protocol, the listening SU will stay on channel 2 and the hopping sequence for the source SU is $\{c_1, c_2, c_5, c_4, c_3\}$. Then, TTR is 2. When the number of channels increases, the number of common available channels also increases, which expedite the rendezvous process and both SUs hop on channel 5 with 1 time slot.

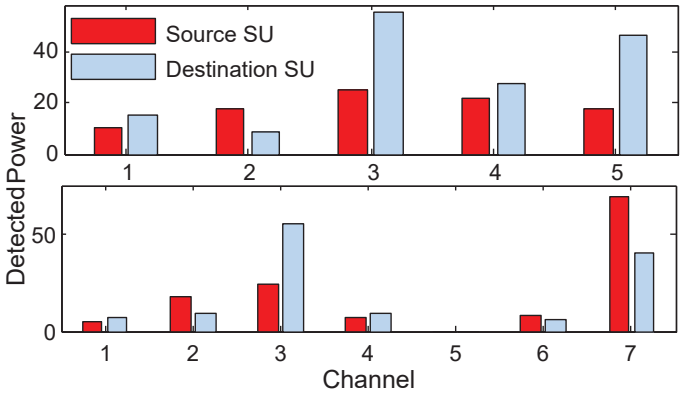


Figure 4.9: Detected power on each channel.

4.5.1 Analysis Validation

Figure 4.10 shows the analytical and simulation results of the ETTR under different number of channels. The analytical results are calculated using our analytical model where $K = 1$, $\rho = 0.375$.

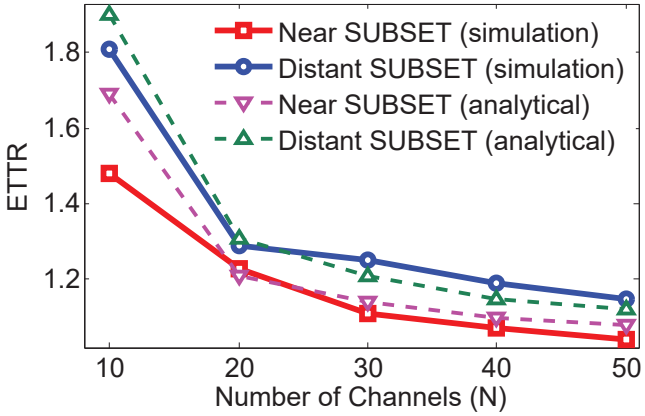


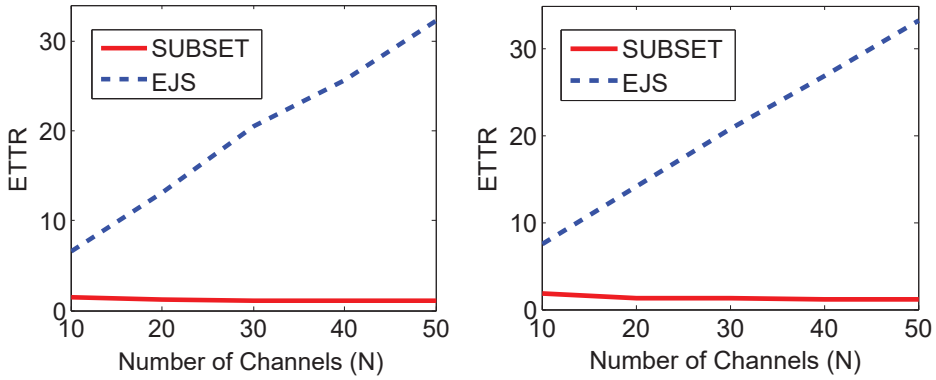
Figure 4.10: ETTR vs. N.

From Figure 4.10 we summarize: *i)* the ETTR of both SUBSET designs approaches 1 as the number of channels in the network increases. This feature truly accords with the goals of cognitive radios to perform better in spectrum-under-utilized scenarios; *ii)* the difference between the simulation and analytical results is 4.3% with 0.07 standard derivation, which validates our analytical models; *iii)* Near-SUBSET performs better than the long-distance SUBSET because the source SU in the design has to hop on every channel before rendezvous; and *iv)* the average number of unavailable channels

is about $K\rho\pi R^2 = 5.7$. When $N = 10$, the network is in spectrum scarcity. Even under this scenario, our proposed SUBSET protocols can still achieve rendezvous within 2 slots.

4.5.2 ETTR and MTTR

The ETTR and MTTR of both the near- and distant-SUBSETs are compared with the typical CH protocol Enhanced Jump-Stay (EJS) [19]. All these protocols can achieve a 100% successful rendezvous rate. The near-SUBSET protocol is used when ($D_{ST} \leq d_r$) and distant-SUBSET is applied when ($D_{ST} > d_r$). From Fig 4.11, when the number of channels in the network increases, the ETTR of EJS increases with $O(N)$, while our SUBSET protocol maintains the same performance due to our $O(1)$ design.

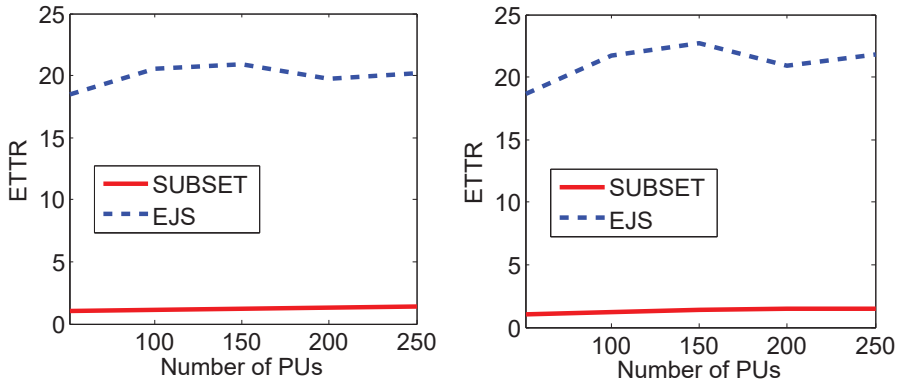


(a) $D_{ST} \leq 0.8D$. (b) $D_{ST} \leq 1.2D$.

Figure 4.11: ETTR vs. N in different protocols ($K = 1$ and $\rho = 0.37$).

From Fig 4.12, when the number of PUs increases, the ETTR performance of all the protocols does not change much. However, the ETTR of SUBSET is still less than 2 time slots even in high-density high-traffic-volume primary networks.

The performance of MTTR is shown in Table 4.4. The results well reflect our $O(1)$ advantage of MTTR, which is significantly lower in SUBSET. A high MTTR can easily cause network congestion. In a recent study [77], a rendezvous threshold is derived for avoiding network congestion under similar parameters. This threshold is



(a) $D_{ST} \leq 0.8D$. (b) $D_{ST} \leq 1.2D$.

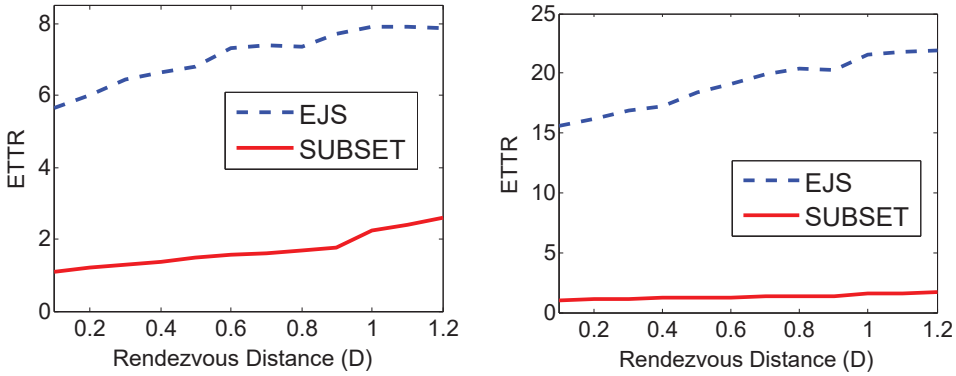
Figure 4.12: ETTR vs. K in different protocols ($N = 30$ and $\rho = 0.52$).

around 10. The MTTR of SUBSET shown in the table is lower than this threshold, which indicates that SUBSET can support a congestion free network.

Table 4.4: MTTR vs. number of channels

Number of channels		10	20	30	40	50
Near Rendezvous	SUBSET	4	4	3	3	2
	EJS	275	296	377	361	463
Distant Rendezvous	SUBSET	9	4	4	3	2
	EJS	269	283	284	379	468

4.5.3 Application Performance

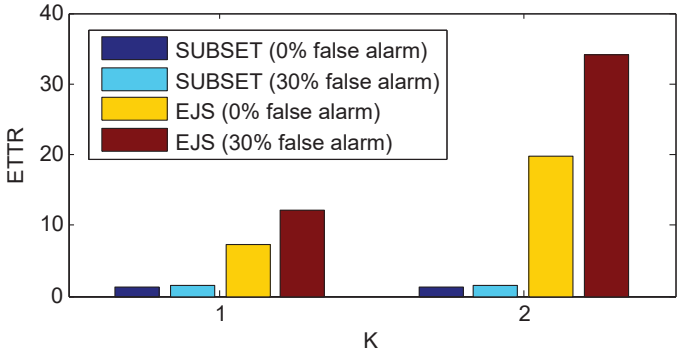


(a) $K = 1$. (b) $K = 2$.

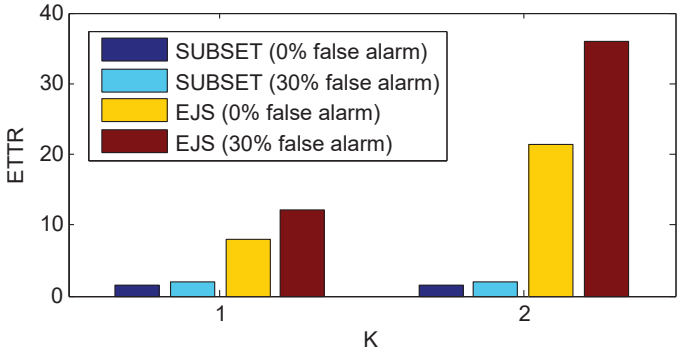
Figure 4.13: Performance comparison with known distance information.

Figure 4.13 demonstrates the ETTR of both SUBSET and EJS over different known rendezvous distance. The performance of SUBSET monotonically increases with the

distance. Actually, it falls into two period according to the rendezvous distance (SU chooses near-SUBSET to rendezvous before $0.9D$ and distant-SUBSET after that) and it is linearly increased in both periods, which validates both the application analysis and the mathematical model. On the other hand, EJS does not have corresponding strategies for different distance. Therefore, its ETTR trend cannot be controlled steadily and the increasing speed is higher than SUBSET when the rendezvous distance is longer, especially in a larger primary network ($K = 2$).



(a) $D_{ST} \leq 0.8D$.



(b) $D_{ST} \leq 1.2D$.

Figure 4.14: Performance comparison under false alarm.

Figure 4.14 shows the performance comparison under false alarm, which consists of idle- and busy- alarm uniformly. K is the number of PUs in a unit area. When $K = 1$, there are 10 channels in the system. When $K = 2$, there are 30 channels in the system. As we can see, both near-SUBSET and distant-SUBSET gain little influence from the false alarm over all conditions. However, the current CH method suffers a lot under sensing mistakes. Especially, when the network is amplified with

more PUs and more channels, the EJS performance degrades more with the same sensing error rate. The impact of the false alarm for both protocols accords with our analysis.

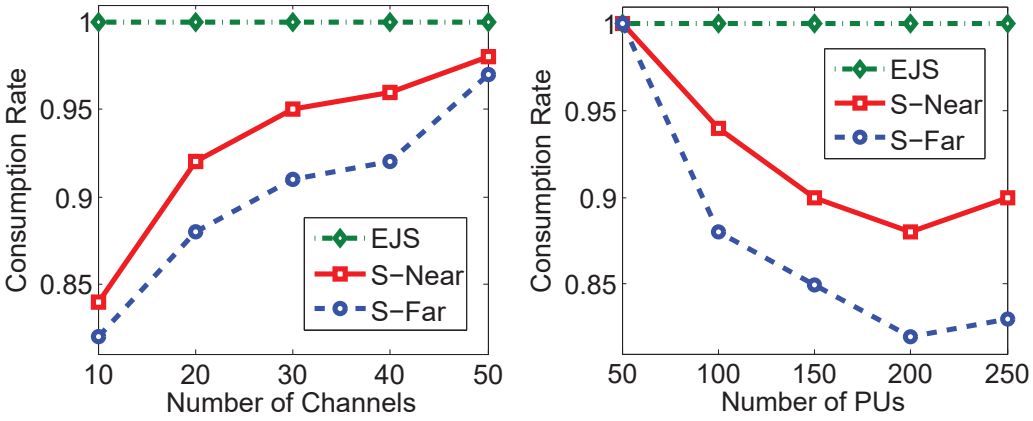
Table 4.5 demonstrates the rendezvous successful rate under false alarm. When there are only 10 channels in the system, with sensing error, the rendezvous sometimes (the special case in busy-alarm analysis cannot be guaranteed, no matter using which CH scheme. However, the successful rate in SUBSET is still higher than that in current effort.

Table 4.5: Rendezvous successful rate under false alarm

false alarm		0%	30%
Near Rendezvous	SUBSET	100%	99.98%
	EJS	100%	99.75%
Distant Rendezvous	SUBSET	100%	99.95%
	EJS	100%	99.66%

4.5.4 Energy Consumption of Idle SUs

We define a metric C to evaluate the energy consumption of an idle SU in CRAHNs. Let $n(l)$ be the number of channels a listening SU hopped in a CH process. Then, $C = \frac{n(l)}{TTR}$ which represents the consumption rate of a listening SU during the rendezvous time. Figure 4.15a shows the impact of spectrum scarcity on C and Figure 4.15b illustrates C in different PU distributions. It is obvious that SUs with SUBSET can enjoy a longer battery life due to less activities during rendezvous, especially when the rendezvous may take a longer time (SUBSET-far), a worse spectrum condition (smaller N as in (a)), or a worse network condition (higher K as in (b)).



(a) C vs. N ($K = 1$). (b) C vs. K ($N = 30$).

Figure 4.15: Idle SU consumption rate under different conditions in CRNs.

CHAPTER 5: PROPOSED 2D-HR PROTOCOL FOR MWB SCENARIOS

In this chapter, we propose a two-dimensional heterogeneous rendezvous (2D-HR) protocol which can support MWB-CRNs with a significantly reduced rendezvous delay and energy consumption for various rendezvous scenarios, such as the pair-wise rendezvous, any-wise rendezvous, and multi-wise rendezvous. The proposed design also performs better than existing efforts even when dealing with traditional single-band rendezvous. The merits of 2D-HR are proved theoretically and validated against extensive simulations. To the best of our knowledge, this is the first work that addresses heterogeneous rendezvous in MWB-CRNs.

5.1 System Model

In MWB-CRNs, spectrum ranging from 30 KHz to 300 GHz is allocated to various service providers. Each spectrum band is divided into multiple channels for different PUs. For instance, the channel bandwidth and the number of non-overlapping channels in different spectrum bands from the VHF band to the 5 GHz band in the U.S. are listed in Table 5.1, where f_c is the central frequency. These frequency bands exhibit vast differences in the available airtime, transmission range, and power consumption. A SU can therefore choose its desired band with great flexibility according to the coverage provisioning, energy management, and QoS provisioning to diverse types of applications. On the other hand, it also needs to avoid the channels used by PUs when performing rendezvous with other SUs based on the pool of channels.

5.1.1 Network Environments

The network considered in this research consists of finite number of PUs and SUs whose locations are randomly distributed but are able to maintain the network con-

Table 5.1: Channels in MWB networks

Band	Bandwidth	# of channels
$f_c = 5$ GHz	20 MHz	24
$f_c = 4.9$ GHz	20 MHz	2
$f_c = 3.65$ GHz	20 MHz	2
$f_c = 2.4$ GHz	20 MHz	3
3G celluar	1.25 MHz	128
2G celluar	200 KHz	124
1G celluar	30 KHz	832
VHF&UHF	6 MHz	68

nectivity, i.e., each PU/SU is within at least one another PU/SU's transmission range. There are totally M spectrum bands in the network. Each of them has a set of non-overlapping channels denoted as $\boldsymbol{B_i} = \{i_1, i_2, i_3, ..., i_{N(i)}\}$, where $N(i)$ is the total number of channels in the ith band.

Primary networks: each time a PU in band i wants to transmit, a channel i_j is randomly selected for the PU. Unlike the PU model considered in other related work that each PU is associated with a unique channel, our model is closer to realistic PU networks in which each PU may be randomly assigned a channel on each transmission or several PUs may share one frequency channel when using mechanisms like time division multiple access (TDMA). Since each channel's availability is not associated with one PU's availability, the channel status is not predictable through PU activities in our design, which adds additional challenges as compared to existing works.

Secondary networks: each SU equipped with one half-duplex radio senses primary networks to get its available channel set (ACS). This process is done periodically and the ACS is updated to avoid the channel status change due to PU/SU activities. Due to the various location and sensibility of SUs, they have heterogeneous ACSs. Without loss of generality, we assume that any two SUs within the transmission range of each other have at least one common available channel in each band (otherwise there is no chance for them to rendezvous anyway).

5.1.2 Rendezvous Algorithm

Each SU keeps hopping onto the channels in its ACS in a predefined order in consecutive time slots. If a source SU has data packets to a destination SU, it sends an request-to-send (RTS) message on each channel it hops to until receiving a correct clear-to-send (CTS) message. Other SUs keep listening on each channel they hop on until receiving a correct RTS. The time slots a source SU spent before completing an RTS/CTS handshake with its destination SU is called time-to-rendezvous (TTR). The current state-of-the-art rendezvous algorithms for heterogeneous ACSs can guarantee the rendezvous for any two SUs who have at least one common available channel within finite number of time slots. Under these algorithms, the maximum TTR (MTTR) is in the order of N^2, where N is the total number of channels in the primary network, and the expected TTR (ETTR) is $O(N)$.

Figure 5.1: Existing rendezvous method for single-band CRNs.

Figure 5.1 illustrates the rendezvous process for a single band CRN where $N = 6$. There are 4 channels in SU_A's ACS and 3 channels in SU_B's. Then, based on an existing hopping algorithm proposed in [21], $TTR = 10$ and channel 2 is the rendezvous channel. Note that in our system, we do not require synchronous hopping which assumes the rendezvous pair to start hopping at the same time slot. For example, if SU_B is on channel 3 at the beginning of SU_A's hopping, they can still rendezvous on channel 2 with $TTR = 6$.

5.1.3 Communication Steps

For a SU pair, after a successful rendezvous, they enter the data transmission phase. In existing CRNs, instead of the rendezvous channel, a common available channel with

the best quality is usually chosen for the transmission in order to avoid potential collisions with PUs [88–90]. However, this step is not necessary in MWB-CRNs since we ask the listening SU to limit its hopping activity in the best suitable band which is dynamically changed with its status or service requirement (e.g., when the SU is in a low battery condition, it chooses a low frequency band; when it has a high data-rate application, it chooses a high frequency band). Therefore, the rendezvous channel itself in our MWB-CRNs is already a proper band with good quality. After data transmission, SUs return to the hopping mode and perform spectrum sensing if necessary.

For a newly joined SU or a re-activated SU, after obtaining its ACS, it first performs any-wise rendezvous to collect the updated network information (explained in Section III). Then, it enters the hopping mode for potential future transmissions.

5.1.4 Performance Metrics

ETTR and MTTR are regarded as two crucial performance metrics in existing rendezvous efforts in CRNs, since they reflect the average and worst-case rendezvous delay which directly affect the throughput in CRNs. In addition, we introduce $C = \frac{l}{ETTR}$ as the consumption rate, where l is the number of channels a listening SU hops during the rendezvous process. C is a parameter between 0 and 1. The lower the C, the longer battery life the SU has. $C = 1$ means that the SU is restless in the network. Meanwhile, we denote t_w as the waiting time for a packet in the queue. t_w does not include the transmission time of other previously transmitted packets. Therefore, t_w is the summation of the rendezvous time of other packets in front of the waiting packet in the queue.

5.2 Proposed 2D-HR Design and Analysis

In this section, we propose a novel rendezvous design for MWB-CRNs including a 2D hopping structure, rendezvous-guaranteed hopping algorithms for SUs in different

status, adaptive RTS frame, and three optimized protocol parameters, all of which are not considered in existing work. Our proposed design can coordinate the three types of rendezvous coexisting in the network. The goal of our design is to achieve a fast and energy-saving rendezvous for SUs in every case.

5.2.1 Pair-wise Rendezvous

Algorithm 8: The pair-wise rendezvous for SUs

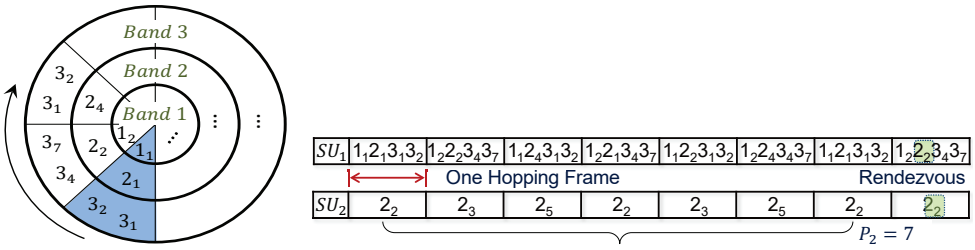
Require: ACS of SU, n_i, and N_i $(i = 1, 2..., M)$;
- **if** *source SU* **then**
  - $ACS_i = ACS$(channels in band i);
  - $I_i = [0, 1, ..., n_i - 1]$;
  - $L_i = length(ACS_i)$;
  - In the initial frame hop and send RTS onto channels $ACS_i((I_i \ mod \ L_i) + 1)$ (take i from 1 to M);
  - **while** *no handshake* **do**
    - $I_i = I_i + n_i$;
    - In the next frame hop and send RTS onto channels $ACS_i((I_i \ mod \ L_i) + 1)$ (take i from 1 to M);
- **if** *listening SU* **then**
  - Find the staying band j;
  - P_j is the first prime number larger than N_j;
  - $k = 0$ and $L = length(ACS_j)$;
  - In the initial frame listening to the channel $ACS_j((k \ mod \ L) + 1)$;
  - **while** *no handshake* **do**
    - $k = (k + 1) \ mod \ P_j$;
    - In the next frame listening to the channel $ACS_j((k \ mod \ L) + 1)$;
  - Send CTS;

For a source SU, when it wants to send data to a particular destination SU, it needs to perform the pair-wise rendezvous as in traditional CRNs. However, in MWB-CRNs, most of the time it neither knows the current staying band nor the hopping channel of its destination SU. Unlike most of existing rendezvous schemes which ask the source SU to hop through all its available channels in the whole network one by one, we divide the available channels into M different classes based on their belonging bands. Furthermore, we construct the hopping sequence using a two dimensional (2D) structure, as shown in Figure 5.2a. We define that a hopping frame has n hopping

time slots. During each frame, a source SU hops onto n_i available channels in band i one by one in n_i time slots and $\sum_{i=1}^{M} n_i = n$, while a listening SU stays on only one of its available channels from its staying band during each hopping frame, which saves the energy consumption of the listening SU. Moreover, i) the channel chosen during each hopping frame for the listening SU is in a predefined order to guarantee the rendezvous; and ii) n_i is optimized to minimize the rendezvous delay. In this way, the rendezvous is achieved at the frame level, i.e., the rendezvous delay is in terms of the number of hopping frames, which then can be converted into the time slot unit by multiplying n.

5.2.1.1 Algorithm for Guaranteed Rendezvous

Algorithm 1 shows our proposed pair-wise rendezvous scheme. We use an example to illustrate our algorithm. Suppose there are totally 3 bands with $N_1 = 3$, $N_2 = 5$, and $N_3 = 7$. The ACS of a source SU (SU_1) on each band is $\{1_1, 1_2\}$, $\{2_1, 2_2, 2_4\}$, and $\{3_1, 3_2, 3_4, 3_7\}$, respectively. The hopping process of SU_1 under our algorithm is illustrated in Figure 5.2a. The shaded sector is a hopping frame with $n = 4$, $n_1 = 1$, $n_2 = 1$, and $n_3 = 2$ (n_i and n are calculated later in the paper). In the indicated frame, SU_1 hops onto $\{1_1, 2_1, 3_1, 3_2\}$ sequentially and check the existence of its destination SU (SU_2) one by one using RTS/CTS (the CSMA mechanism adopted in 802.11). If there is no handshake, it performs another hopping frame and selects the hopping channels in a cyclic manner from each band.



(a) the hopping frame for basic rendezvous.

(b) The rendezvous process in band 2.

Figure 5.2: An example of the 2D rendezvous structure.

On the other hand, suppose SU_2 selects band 2 to stay on (based on its status and service requirements) with the ACS $\{2_2, 2_3, 2_5\}$. Under our algorithm, in each hopping frame, it hops to and stays on one channel in its ACS. The channel is selected one by one also in a cyclic manner but with the period $P_2 = 7$. Then, the rendezvous process for SU_1 and SU_2 is illustrated in Figure 5.2b. The handshake happens on the 2nd hopping slot in the 8th frame, where $TTR = 7n + 2 = 30$.

Note that the hopping period of SU_1 and SU_2 are 3 and 7 frames for band 2, respectively, which are co-prime numbers. By Chinese Remainder Theorem [21], the rendezvous is guaranteed with a bounded delay no matter when SU_2 begins its hopping period (In Figure 5.2b, they begin in the same frame). In fact, our algorithm always guarantees the rendezvous.

Proof. Under Algorithm 1, if the destination SU is in band i, the hopping period is P_i frames and P_i is the first prime number larger than N_i. Meanwhile, the hopping period of the source SU is L_i frames for channels in band i. Since L_i is the number of available channels in band i while N_i is the total number of channels in band i, we have $L_i \leq N_i < P_i$. Therefore, L_i and P_i must be co-prime for every possible i. Furthermore, the MTTR of our proposed scheme is $\max_i nL_iP_i$ in the unit of time slots. Since $n \geq M$, $L_iP_i \approx {N_i}^2$, and $MN_i \approx N$, our MTTR is $O(\frac{N^2}{M})$, which is largely reduced from the existing $O(N^2)$. □

5.2.1.2 Optimal n_i for Fast Rendezvous

First, let a source SU only hop onto one channel in a single band (N) in one hopping frame. Since the ACS of each SU is a subset of the set of all channels, when a SU hops on to a channel from this subset, it is equivalent to randomly choose a channel from a total of N channels. Therefore, the rendezvous probability in this frame is $p_r = \binom{N}{1}\frac{1}{N}\frac{1}{N} = \frac{1}{N}$. The ETTR of the rendezvous is then $ETTR = n\sum_{j=1}^{\infty} j(1 - p_r)^{j-1}p_r = \frac{n}{p_r}$. Next, if the source SU hops onto n channels in this band during

this frame, the one-frame rendezvous probability then becomes $p_r = \binom{n}{1}/\binom{N}{1} = \frac{n}{N}$. Moreover, suppose there are M bands and the SU hops n_i channels in each band i. Then, we have the optimization problem:

$$\begin{aligned} \underset{\{n_i\}}{\text{Minimize}} \quad & E(pair.) = \frac{n}{\sum_{i=1}^{M} \boldsymbol{p_d}(i)\frac{n_i}{N_i}} \\ \text{subject to} \quad & \sum_{1}^{M} n_i = n,\ n_i \leq L_i,\ \text{and } n_i \in \mathbb{Z}^+. \end{aligned} \tag{5.1}$$

Two parameters need to be mentioned in (1). One is $\boldsymbol{p_d}$, the probability that the destination SU stays on each band ($\sum_1^M \boldsymbol{p_d}(i) = 1$), which can be generated by the rendezvous history. For example, if a source SU rendezvous with another SU 10 times in the past, and among them 5 times are in band 1, 3 times in band 2, 1 time in band 3, and 1 time in band 4, then we can generate $\boldsymbol{p_d} = [0.5\ 0.3\ 0.1\ 0.1]$. If there are other bands, always count 1 time for each of them for the sake of rendezvous guarantee. $\boldsymbol{p_d}$ can also be calculated in a more practical way. If a SU just rendezvous with its current destination SU on band i not long ago, it can adjust $\boldsymbol{p_d}$ with more weight on $\boldsymbol{p_d}(i)$. The specific/optimal $\boldsymbol{p_d}$ generating process is out of the scope of this paper. On the other hand, if there is no relevant information about the destination SU, $\boldsymbol{p_d}$ can be calculated based on the band size: the band with more channels has a higher probability to be chosen as a listening SU's staying band. For instance, in the network in Figure 5.2a, $\boldsymbol{p_d} = [\frac{3}{15}\frac{5}{15}\frac{7}{15}]$. That is why the algorithm generates more channels to hop on in band 3 ($n_3 = 2$) in order to satisfy (1).

The other parameter is n. If n is not given, it will have an optimal value from the optimal $\{n_i\}$ generated by (1). On the other hand, if n is already generated by the system (explained in Section IV), there also exists an optimal combination of $\{n_i\}$ by (1). Note that the optimal $\{n_i\}$ can always guarantee the ETTR less than that under existing works ($ETTR = N$).

Proof. Let $A(i) = \{n\frac{N_i}{N}\}$ which is one of all $\{n_i\}$ combinations. Suppose the optimal

$\{n_i\}$ is B. Then, B can always satisfy that $E(pair.)|_{\{n_i\}=B} \leq E(pair.)|_{\{n_i\}=A}$, where $E(pair.)|_{\{n_i\}=A} = \frac{n}{\sum_{i=1}^{M} \boldsymbol{p_d}(i)\frac{nN_i/N}{N_i}} = \frac{n}{n\sum_{1}^{M} \boldsymbol{p_d}(i)/N} = N$. □

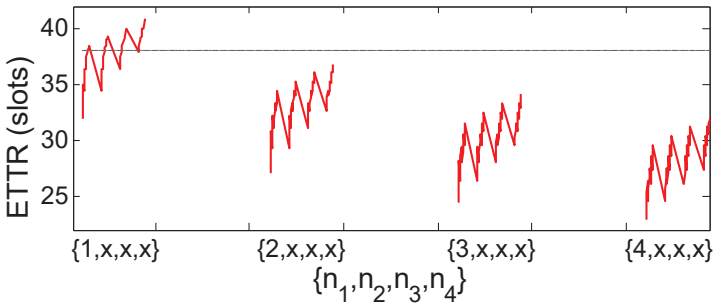


Figure 5.3: The rendezvous performance over different hopping frames.

Figure 5.3 illustrates the performance of different $\{n_i\}$ combinations. The network has 4 bands with 4, 15, 10, and 8 channels in each band. The source SU has $\boldsymbol{p_d}$ = [0.3 0.3 0.2 0.2] for its destination SU. Other settings are the same as those in Section V. The combination of $\{n_i\}$ is listed from $\{1, 1, 1, 1\}$ to $\{4, 4, 4, 4\}$. As we can see from Figure 5.3: 1) ETTR gets its minimum value at $\{4, 1, 1, 1\}$ which accords with our derivation based on (1). Therefore, simulation results shown in Figure 5.3 validate our optimal $\{n_i\}$ derived from (1). That is, hopping 4 channels in band 1 and 1 channel in each of the rest bands during one hopping frame; 2) if n is given, among all the satisfied $\{n_i\}$, there must exist a minimum ETTR; and 3) the horizontal line indicates the performance of an existing state-of-the-art rendezvous algorithm [19] for the same network. Our scheme reduces almost half of the average rendezvous delay. Note that even the $\{1, 1, 1, 1\}$ case has improved the performance significantly, which indicates the success of our 2D structure. Moreover, if the network contains more bands and channels, the performance gap will be further expanded.

5.2.1.3 In-band Rendezvous

Sometimes, the source SU knows which band the destination SU is staying on, i.e., $\boldsymbol{p_d}(i) = 1$. This may happen when the destination SU told the source SU how long it is going to stay on a band during a recent communication. In this case, the source SU

only needs to hop through the channels in the target band, which we call it in-band rendezvous. This is actually applying our scheme to the single band rendezvous.

If the band is a narrow band with only a few channels, the source SU can just let $n_i = n$ and continue Algorithm 1. That is, hopping through n channels in band i during each frame (the size of the frame is unified among all SUs because n is given by the system). In this way, the one-frame rendezvous probability is $p_r = \frac{n}{N_i}$, and then $ETTR = n/(\frac{n}{N_i}) = N_i$. If we consider this rendezvous as the single band rendezvous, our ETTR is $O(N)$ which is the same as that of the existing rendezvous algorithms.

On the other hand, if the target band is a wideband with many channels, we can still use our proposed 2D rendezvous scheme with a little revision as a fractal design: in MWB-CRNs, we classify the channels based on their belonging bands. When it comes to such a wideband, we divide the channels into M classes but by their qualities. The listening SU only hops on the channels in the good quality class. Then, the class with better quality channels has the corresponding high probability to rendezvous with the destination SU. In this way, the rendezvous scenario can be exactly matched to the MWB scenario. Consequently, the optimization (1) and Algorithm 1 can then be used for the single-wideband case, which reduces the ETTR and MTTR in a similar way.

5.2.2 Any-wise Rendezvous

A newly joined or re-activated SU needs to rendezvous with any other SU to obtain the network information such as their neighboring SU distribution over different bands ($\boldsymbol{k}$) and the staying probability distribution ($\boldsymbol{p_d}$) of a listening SU as a reference. This information can help the new SU select its staying band and construct the hopping frame for later use. Since this type of rendezvous does not require a specific destination SU, the receiver's address is not necessary in the RTS frame. In addition, most existing rendezvous schemes require the same time slot size for all SUs all the time to guarantee the potential rendezvous between any two SUs. Therefore, the RTS

frame (20 octets) cannot be altered due to this homogeneous-slot requirement.

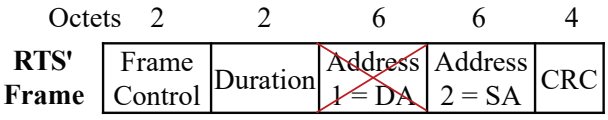


Figure 5.4: The RTS for the any-wise rendezvous.

However, in our scheme, we only require homogeneous-frame, i.e., *no matter how the slot size changes, all types of rendezvous between any two SUs can be guaranteed as long as all SUs have the unified hopping frame size.* As shown in Figure 5.4, our RTS frame for the any-wise rendezvous scenario removes the receiver's address. The total size is then the same as the CTS size (14 octets). Suppose the slot size for the pair-wise rendezvous is $(20 + 14)t$, where t is a constant. Then, the homogeneous frame size is $n(20 + 14)t$. On the other hand, the slot size for any-wise rendezvous is $(14 + 14)t$. Then, the number of such slots allowed in our frame is

$$n' = \frac{n(20 + 14)t}{(14 + 14)t} = \lfloor \frac{34}{28} n \rfloor, \tag{5.2}$$

which provides SUs more channels to hop during a frame. Note that the corresponding frame control can be modified at the bit level which is negligible for our analysis.

5.2.2.1 Single Band Analysis

We first analyze the optimal n' for our any-wise rendezvous in the single band scenario. Suppose there are k neighboring SUs of a newly joined SU in the band (N). The rendezvous probability with any of them in a hopping time slot is $1 - (1 - \frac{n'}{N})^k$. Similar to the analysis in the pair-wise scenario, the average rendezvous delay is then $ETTR = \frac{n}{1-(1-n'/N)^k}$ in the unit of the original time slots.

5.2.2.2 MWB Scenario

Next, we investigate the optimal $\{n'_i\}$ for MWB scenarios. For the any-wise rendezvous, since the source SU does not know the distribution $\boldsymbol{k}$ of neighboring SUs, it has to utilize the simple staying probability distribution as mentioned before. The

source SU first sets K as the number of its neighboring SUs (K can be concluded from statistic data). Then, $\boldsymbol{k} = \lfloor K\boldsymbol{p_d} \rfloor$. With $\boldsymbol{k}$, the probability of rendezvous with any neighboring SU in one hopping frame can be derived. Finally, the optimization problem becomes:

$$\begin{aligned} \underset{\{n_i'\}}{\text{Minimize}} \quad & E(any.) = \frac{n}{1 - \prod_{i=1}^{M}(1 - \frac{n_i'}{N_i})^{\boldsymbol{k}(i)}} \\ \text{subject to} \quad & \sum_{i=1}^{M} n_i' = n', \ (2), \ n_i' \leq L_i, \ \text{and} \ n_i' \in \mathbb{Z}^+. \end{aligned} \tag{5.3}$$

For a given n, use (3) to generate the optimal $\{n_i'\}$ in terms of the fastest any-wise rendezvous. Then, the any-wise rendezvous in MWB-CRNs can be done by replacing $\{n_i\}$ and the RTS frame in Algorithm 1 with $\{n_i'\}$ and the frame shown in Figure 5.4, respectively. The optimal $\{n_i'\}$ can also always guarantee a shorter ETTR than existing methods.

Proof. The ETTR for the first rendezvous SU using existing methods is $A = \frac{1}{1-(1-\frac{1}{N})^K}$. Since N is very large in MWB-CRNs, we have $A = \frac{N}{K}$. On the other hand, $E(any.) \leq \frac{\frac{28}{34}M}{1-\prod_{i=1}^{M}(1-\frac{K/M}{N/M})} < \frac{N}{K} = A$. □

5.2.3 Multi-wise Rendezvous

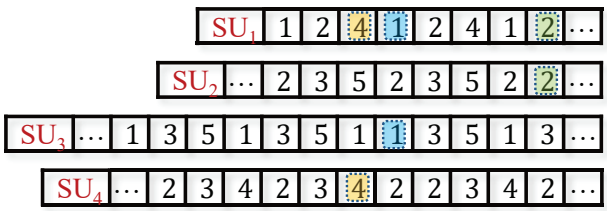


Figure 5.5: The scenario of multi-wise rendezvous.

In existing rendezvous schemes, when a source SU has multiple packets with different destinations in the queue, it has to rendezvous with each destination SU one by one. However, this might be a waste of opportunity. As illustrated in Figure 5.5, SU_1 has data for SU_2, SU_3, and SU_4 to transmit. Each SU has different ACS and

hops asynchronously. As we can see, if SU_1 wants to send SU_2's data first, it will rendezvous with SU_2 on channel 2 with $TTR = 8$. However, it actually misses the rendezvous opportunities with SU_3 on channel 1 with $TTR = 4$ and with SU_4 on channel 4 with $TTR = 3$. In other words, if SU_1 can perform rendezvous with all of them simultaneously, it can always rendezvous with the one with the shortest possible TTR first, which expedites the whole process and reduces the average waiting time $\overline{t_w}$.

Similar to the any-wise rendezvous analysis, existing schemes cannot achieve this function due to the homogeneous time slot. Fortunately, our scheme again can revise the RTS frame as long as the hopping frame is unchanged. As shown in Figure 5.6, we expand the RTS frame by adding more destination SUs' addresses. In this way, it is equivalent to achieve simultaneous hopping for multiple destination SUs, or, multi-wise rendezvous. Suppose there are totally m destination SUs need to be rendezvous, and the number of receivers' addresses in the corresponding RTS frame is d ($d \leq m$). The optimal d in terms of minimizing the first rendezvous delay is a function of m. Then, the number of time slots for m-wise rendezvous in one frame is

$$n'' = \frac{n(20+14)t}{(14+6d+14)t} = \lfloor \frac{34n}{28+6d} \rfloor, \tag{5.4}$$

which means that there are less number of hopping slots in a frame than the pair-wise case but more rendezvous opportunities in each slot. Note that after the first rendezvous, n'' and d changes with $m = m - 1$ until $m = 0$. In this way, the optimal RTS frame dynamically changes with the number of destination SUs in the queue, which reduces the overall rendezvous delay throughout the whole process.

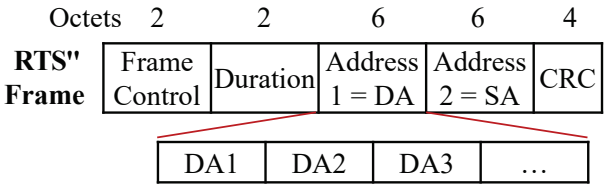


Figure 5.6: The RTS for the multi-wise rendezvous.

5.2.3.1 Single Band Analysis

Again, we first analyze the multi-wise rendezvous in the single band scenario. Similar to previous analysis, the average rendezvous delay for such case is $ETTR = \frac{n}{1-(1-n''/N)^d}$ in the unit of the original time slot.

5.2.3.2 MWB Scenario

Then, we investigate the optimal d for m destination SUs in MWB scenarios. Suppose the $\boldsymbol{p_d}$ for each destination SU is $\boldsymbol{p_d^j}$ $(j = 1, 2, ..., m)$. Then, the one-frame rendezvous probability with the jth destination SU, based on (1), is $p_r = \sum_{i=1}^{M} \boldsymbol{p_d^j}(i)\frac{n_i''}{N_i}$. Since we can treat all the potential rendezvous with d SUs happening simultaneously, the probability that at least rendezvous with one destination SU in this frame, based on (3), is $1-(1-p_r)^d$. Therefore, the optimization problem can be expressed as:

$$\begin{aligned} \underset{d}{\text{Minimize}} \quad & E(multi.) = \frac{n}{1-\prod_{j=1}^{d}(1-\sum_{i=1}^{M}\boldsymbol{p_d^j}(i)\frac{n_i''}{N_i})} \\ \text{subject to} \quad & \sum_{1}^{M} n_i'' = n'', \ (4), \ n_i'' \le L_i, \text{ and } n_i'' \in \mathbb{Z}^+. \end{aligned} \tag{5.5}$$

In addition, if all destination SUs have the same $\boldsymbol{p_d^j}$ (e.g., with information for none of them), then $\{n_i''\}$ can be strictly matched to $\{n_i\}$ obtained from (1) by replacing n and n'' in (4) with n_i and n_i''. Finally, for a given n, use (5) to generate the optimal d in terms of the fastest first rendezvous for such scenarios. Then, the multi-wise rendezvous in MWB-CRNs can be done by replacing $\{n_i\}$ and the RTS frame in Algorithm 1 with $\{n_i''\}$ and the frame shown in Figure 5.6, respectively. Similarly, the ETTR under the optimal d is always shorter than that under traditional ways.

Proof. $E(multi.) \le \frac{n}{1-\prod_{j=1}^{d}(1-\sum_{i=1}^{M}\frac{1}{M}\frac{n_i''}{N/M})} \le \frac{n}{n''d/N} = N/(\frac{34d}{28+6d}) \le N$. On the other hand, since the ETTR of the first rendezvous in this scenario (A) using existing methods is the same as that in the pair-wise scenario (N), we have $E(multi.) \le A$. The equality holds if and only if there is only one destination SU $(d = 1)$ and the

equality condition holds in the ETTR proof of the pair-wise rendezvous. □

5.3 2D-HR Protocol

5.3.1 Parameter Optimization

As mentioned before, since the hopping frame should be equilong for all SUs for the sake of guaranteed rendezvous, the unified size of the hopping frame n should be determined as a constant and be able to serve all SUs for all cases, such as the pair-wise rendezvous, any-wise rendezvous, and multi-wise rendezvous. First, we need to derive the average rendezvous delay of all cases for a given n: i.e., $E(pair.)$ in (1), $E(any.)$ in (3), and $E(multi.)$ in (5). Then, the probability distribution over all cases, $\boldsymbol{Pr}$, should be concluded from the statistic study. For example, $\boldsymbol{Pr}(1)$ is the percentage of the pair-wise rendezvous over all the rendezvous activities of a SU during its network life. Finally, the optimal n in terms of minimizing the total rendezvous delay $\overline{\tau}$ in MWB-CRNs can be obtained from:

$$\begin{aligned} \underset{d}{\text{Minimize}} \quad & \overline{\tau} = \boldsymbol{Pr}[E(pair.)E(any.)E(multi.)] \\ \text{subject to} \quad & (1),\ (3),\ (5),\ n \geq M, \text{ and } n \in \mathbb{Z}^+, \end{aligned} \tag{5.6}$$

under which follows the local optimal $\{n_i\}$, $\{n_i'\}$, d, and $\{n_i''\}$.

5.3.2 Protocol Details

Figure 5.7 shows the complete framework of our proposed protocol. Each time when a SU joints or wakes up in the network, it performs the any-wise rendezvous. Then, according to the number of packets in the queue (0, 1, and > 1), it takes corresponding actions such as turning into the listening SU, constructing the proper hopping frame for the pair-wise rendezvous, or doing it for the multi-wise rendezvous. All the parameters shown in the framework can be either directly obtained from the network, the sensing results, and other SUs, or derived by the given methods in previous sections. In addition, the dynamically obtained or generated parameters $\boldsymbol{p_d}$

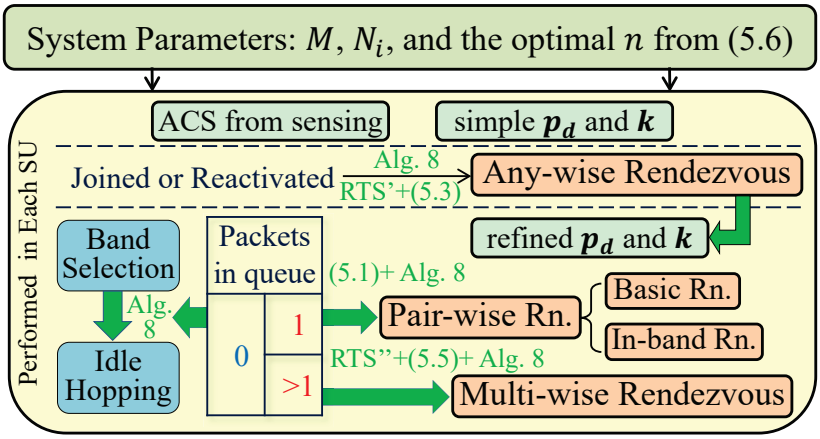


Figure 5.7: The framework of the proposed protocol.

and $\boldsymbol{k}$ can assist to update the local optimal parameters $\{n_i\}$, $\{n_i'\}$, and d. In this way, our proposed 2D-HR scheme can adapt with the changing network conditions all the time.

Moreover, in our protocol, the hopping mechanism for the listening SU has two more advantages. One of the advantages is that tight time synchronization is not required. Most existing efforts either require the time slots of each SU to be strictly synchronized [91] or expand the slot size to guarantee the handshake [5, 74]. It is proved in [32] that our hopping structure of the listening SU satisfies the asynchronous-rendezvous requirement. Therefore, our rendezvous delay is further reduced by maintaining the basic slot size in asynchronous scenarios. Another advantage is the energy saving due to less operation. As mentioned before, the SU during the listening period has fewer operations than its active period. The saved energy extends a SU's battery lifetime in the network.

5.4 Performance Evaluation

In our simulation, PUs and SUs are randomly distributed in the simulation area. Each PU is randomly assigned a channel from its belonging spectrum band when a new packet needs to be transmitted. Source SUs are randomly selected and the number of packets in the queue is randomly assigned. The destination SU is randomly chosen from the source SU's transmission range. The parameters used in our

simulation are listed in Table 5.2. In the following simulation results shown, the unit for $ETTR$, $MTTR$, and $\overline{t_w}$ is the number of the original time slots.

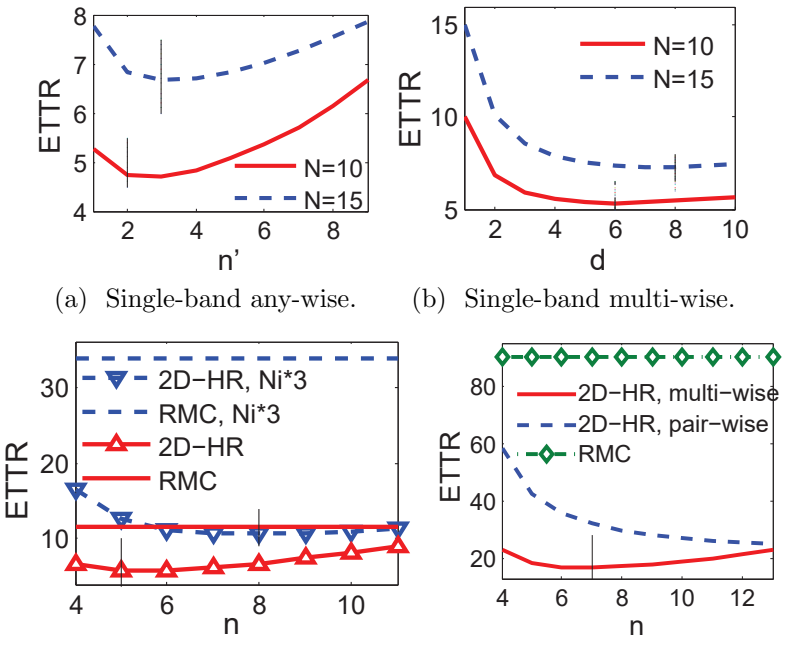
Table 5.2: Simulation parameters for 2D-HR

Simulation time	100000 slots
Simulation area	50 m × 50 m
PU/SU sensing radius	10/7 m
Average PU packet arrival rate	50 pkt/s
Number of PUs/SUs	100/100
PU packet size	50 slots
The size of (RTS+CTS)	160 + 112 bits (802.11 b/g)

5.4.1 Optimization Validation

Since the optimization of the pair-wise rendezvous is analyzed and validated, we only analyze other types of rendezvous here.

5.4.1.1 Any-wise Rendezvous



(a) Single-band any-wise. (b) Single-band multi-wise.

(c) MWB any-wise. (d) MWB multi-wise.

Figure 5.8: The optimization and performance of 2D-HR.

Figure 5.8a shows the performance over different n' with the number of neighboring SUs $k = 2$ in the single band. As we can see, the optimal n' from simulation accords

with our derivation based on (3) (vertical lines) and differs with N ($n' = 2$ when $N = 10$ and $n' = 3$ when $N = 15$). Moreover, the ETTR of our designs outperforms existing efforts ($\frac{N}{k}$) even in the single band scenario. On the other hand, the scheme is compared with the RMC scheme [18] in MWB scenarios, as shown in Figure 5.8c. Two networks are considered: $M = 4$ with $N_i = 10i$ and $M = 4$ with $N_i = 30i$. As we can see, 1) the optimal n from simulation for both scenarios accords with our derivation based on (3) (vertical lines); and 2) the ETTR under 2D-HR (at the optimal n) is almost $\frac{1}{3}$ of that under RMC for both cases, which is a great feature especially in a large-scale MWB-CRN.

5.4.1.2 Multi-wise Rendezvous

Figure 5.8b shows the performance of multi-wise rendezvous over different d, the number of receiver addresses in RTS, with the number of destination SUs $m = 10$ and $n = 10$ in a single band. As we can see, the optimal d accords with our derivation and increases with N ($d = 6$ when $N = 10$ and $d = 8$ when $N = 15$). Moreover, the first ETTR of our scheme for multi-wise rendezvous is almost two times faster than that of existing ways (N) even in the single band. On the other hand, we also compared our scheme with RMC [18] in MWB scenarios, as shown in Figure 5.8d. A small-scale MWB scenario ($M = 4$ with $N_i = 10i$) is considered here. As we can see, our scheme outperforms significantly (near $\frac{1}{5}$ of RMC) even in the small-scale scenario. Moreover, note that even using the pair-wise algorithm with the traditional one-by-one rendezvous strategy, our proposed multi-wise scheme still triumphs, since the multi-wise scheme uses a totally different RTS to hop with multiple SUs simultaneously.

5.4.2 Performance Comparison

5.4.2.1 Average Waiting Time

We compare $\overline{t_w}$ with the EJS scheme [19] over the number of queued packets in a $M = 4$ network with $N_i = 10$. As shown in Table 5.2a, the average waiting time under both algorithms increases with the longer queue. However, packets in the queue under 2D-HR enjoys a much less $\overline{t_w}$ ($\frac{1}{4}\sim\frac{1}{5}$) with a slower increasing speed as compared with that under EJS, because 2D-HR seizes every rendezvous opportunity for each of its packets all the time.

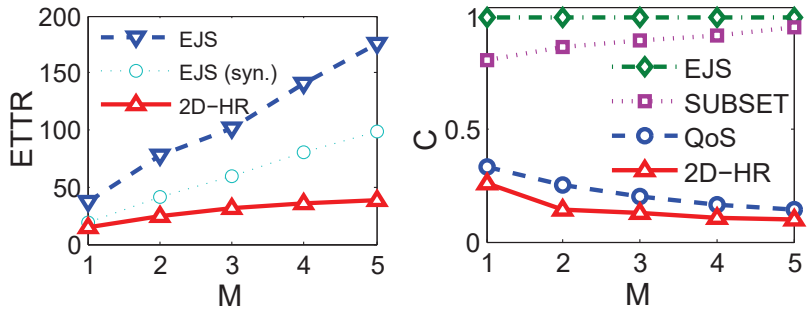
Table 5.3: The average packet waiting time comparison

m	1	2	3	4	5	6	7
EJS	98	145	192	237	282	325	368
2D-HR	26.3	32.8	39.6	46.8	54.3	62.1	70.1

5.4.2.2 Overall Rendezvous Delay

Next, we compare the overall rendezvous delay between 2D-HR and EJS including all types of rendezvous appearing during the network life, as shown in Figure 5.9a. The comparison is performed in different networks ($N_i = 20$ but M varies) and conditions (has a global clock or not). As we can see, 1) even in the single-wideband CRN where $M = 1$, 2D-HR enjoys the shorter rendezvous delay due to our fractal design for such cases; 2) 2D-HR does not degrade in the slot-asynchronous condition; 3) EJS performs worse in the slot-asynchronous scenario since it has to enlarge the slot size to guarantee the handshake; and 4) 2D-HR outperforms in all cases, especially when the MWB-CRN is larger. The reason 2D-HR maintaining a lower ETTR even in a large-scale MWB-CRN is because the any-wise and multi-wise schemes keep updating the accurate network information and cleaning congested packets. With a precise $\boldsymbol{p_d}$, sometimes the rendezvous even turns into the in-band rendezvous. Such mechanism is especially useful for MWB-CRNs.

The MTTR comparison is concluded in Table 5.4. Even though the simulation is



(a) Overall rendezvous delay. (b) Consumption rate

Figure 5.9: The overall performance comparison in MWB-CRNs.

conducted in a moderate MWB scenario with $N_i = 10$, the MTTR under existing algorithms goes tremendously high when the number of bands increases. In addition, the results are satisfied with our theoretical inference: the MTTR of 2D-HR is at $O(\frac{N^2}{M})$, which shows great advantage when the network is large as compared to existing ones ($O(N^2)$).

Table 5.4: The MTTR comparison ($N_i = 10$)

M	1	2	3	4	5	6	7
EJS	51	204	459	816	1275	1836	2401
2D-HR	41	103	148	214	245	301	358

5.4.2.3 Energy Consumption

As defined in Section II-B, we use C to evaluate the energy consumption of a listening SU. Figure 5.9b shows the impact of the network size on C ($N_i = 20$ but M varies). We add two more existing schemes (QoS in [32] and SUBSET in [92]) that showed prominent features in this aspect to compare with our design. It is excited to see that 2D-HR outperforms all of them and more importantly, it decreases with the network size. This feature allows SUs under our protocol to enjoy a longer battery life due to less activity in larger MWB-CRNs.

CHAPTER 6: PROPOSED PCH PROTOCL FOR IOT/5G SCENARIOS

In this chapter, we propose PCH, a priority-based spectrum access protocol for cognitive D2D, which can be integrated with any existing CH algorithm. PCH can support priority transmissions with a significantly reduced CH delay, as compared with non-priority transmissions. More importantly, PCH can work under practical scenarios such as the D2D both with priority packets, the D2D handoff, and the overhead/energy constraint IoT. The merits of PCH are proved theoretically and validated against extensive simulations. To the best of our knowledge, this is the first work that investigates priority communications in cognitive IoT/5G.

6.1 Problem Description

6.1.1 System Model

The system considered in this research divides time evenly into time slots with homogeneous size and consists of a finite number of CRDs and other usual devices in other coexisting networks which can operate on a set of orthogonal channels denoted as $\boldsymbol{C} = \{c_1, c_2, c_3, ..., c_M\}$.

6.1.1.1 Spectrum Access Process

Each time a CRD wants to transmit or listens for a potential transmission, all other transmission-ongoing devices within its sensing range are regarded as primary users (PUs) in CRNs. The CRD senses the primary network to get its available channel set (ACS), S_{CRD}, for the sake of interference-free to PUs, just like a secondary user (SU) in CRNs. This process is done periodically, and the ACS is updated to avoid channel status change due to PU activities. Then, each CRD generates a sequence of channels from its ACS based on the adopted CH algorithm. In each time slot, it hops

onto one channel following the sequence. The number of time slots a source CRD hopped before rendezvous on the same channel with its destination CRD is called time to rendezvous (TTR).

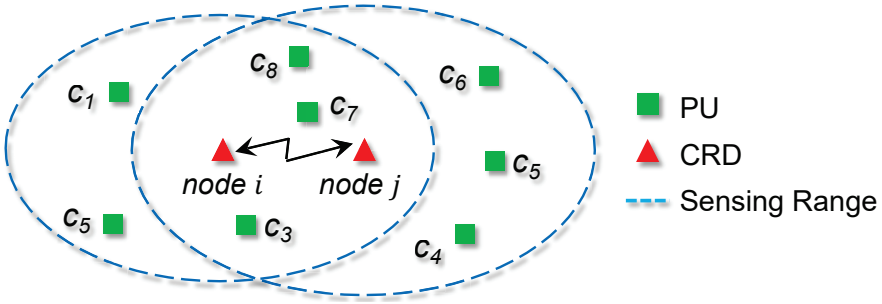


Figure 6.1: Heterogeneous ACS scenario.

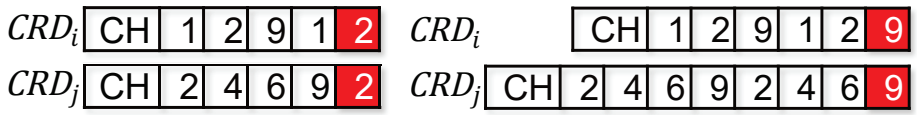
Note that the ACS of each CRD may not be homogeneous due to the diversity of each CRD's location and sensing capability. As illustrated in Figure 6.1, CRD_i and CRD_j have different sensing ranges and the PUs distributed within their sensing ranges are using channels from the set $\{c_1, c_2, c_3, ..., c_9\}$. Therefore, the ACSs of CRD_i and CRD_j in this case are $S_i = \{c_2, c_4, c_6, c_9\}$ and $S_j = \{c_1, c_2, c_9\}$, respectively. Moreover, without loss of generality, we assume that any two CRDs within the transmission range of each other have at least one common available channel.

6.1.1.2 CH Algorithm

The state-of-the-art CH algorithms for heterogeneous ACSs can guarantee the rendezvous for any two SUs who have at least one common available channel within finite time slots. Under these algorithms, the maximum TTR (MTTR) is in the order of M^2, where M is the total number of channels in the primary network, and the expected TTR (ETTR) is $O(M)$.

In our system, we do not require synchronous CH which assumes the rendezvous pair to start CH at the same time. For example, using the example in Figure 6.1 and assuming that CRD_i is a source SU and its destination SU is CRD_j, we adopt the CH method proposed in [21] which can guarantee the rendezvous for heterogeneous

ACSs in an asynchronous manner. As illustrated in Figure 6.2, CRD_i can rendezvous with CRD_j no matter their CH is synchronous or not.



(a) synchronous CH, TTR=5 (b) asynchronous CH, TTR=6

Figure 6.2: An example of synchronous and asynchronous CH.

We denote $R^d(P(S_i), P(S_j))$ as the rendezvous time of the source CRD_i with its destination CRD_j. $P(S_x)$ is the permutation function of a given ACS, which depends on the CH algorithm adopted. For example, $P(\{c_1, c_2, c_3\}) = \{3, 3, 3, 2, 2, 1\}$. The hopping sequence follows the permutation periodically. d is the offset slots of CRD_j which can also be a negative integer. Once d, $P(S_i)$, and $P(S_j)$ are determined, a CRD can have one and only one TTR. In the above example, $R^0(\{1, 2, 9\}, \{2, 4, 6, 9\}) = 5$ and $R^2(\{1, 2, 9\}, \{2, 4, 6, 9\}) = 6$.

6.1.1.3 Link Speed Setting

In emerging networks such as the 5G network, users may require different spectrum bands to serve different purposes. Since different bands are centered at different frequencies with different bandwidths, and cognitive radio usually uses different modulation techniques and protocols on its operating channel, the data rate, (or, the link speed) of different channels varies.

Denote $\boldsymbol{V} = \{v_1, v_2, v_3, ..., v_M\}$ as the link speeds of the operational channels in set $\boldsymbol{C}$. For a given packet with length L, the transmission time on each channel is $t_i = \frac{L}{v_i}$, $v_i \in \boldsymbol{V}$. In our model, we reorder the channels in $\boldsymbol{C}$ by their link speed in a **fast-to-slow** order: $S_{SU} = \{c_1, c_2, c_3, ..., c_M | t_1 \leq t_2 \leq ... \leq t_M\}$.

6.1.2 Design Challenges and Requirements

In cognitive D2D networks, different packets may have different delay requirements. One intuitive way is to give the packet with a low delay tolerance a higher priority

by sending the packet on a higher speed channel. Generally, if there are totally k priorities for different data packets, $P_{CRD} = \{p_1, p_2, ..., p_k\}$ (ranked based on priority from **high to low**), a CRD should use the corresponding k fast channels $\{c_1, c_2, ..., c_k\}$ for each priority. Such design is illustrated in Figure 6.3, where $\{\tau_1, \tau_2, ..., \tau_k\}$ is the spectrum access delay (TTR) for each priority packet. In order to fulfill the design purpose, it requires

$$\tau_i + t_i \leq \tau_j + t_j, \qquad i < j. \tag{6.1}$$

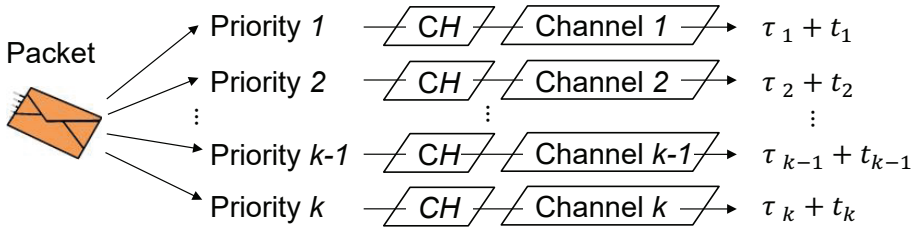


Figure 6.3: Ideal design for packets with different priorities.

However, the reality cannot support this ideal design. First, as mentioned in the system model, the ACS of each CRD changes with time and location. Thus, a packet with a fixed priority should not be associated with a fixed transmission channel. Instead, a source CRD (CRD_i) should send its highest-priority packet to its fastest available channel among all its common available channels and so forth: $c(k) = S_{i,j}(k)$, where k is the corresponding priority and $S_{i,j} = S_i \cap S_j$.

Since the rendezvous channel is usually very random (see the example in Figure 6.2) and usually not the corresponding channel of a particular priority, the rendezvous pair may need to handoff to the desired channel after the rendezvous if necessary, which causes extra delay.

More importantly, the TTR of each packet may vary a lot. Based on the research in [77], in practical CRNs, the TTR is in the same order and sometimes even longer than the packet transmission time. Such circumstances may cause the inequality in (1) invalid. Therefore, as illustrated in Figure 6.4, in order to satisfy (1), the desired

design should guarantee that *the packet with a priority enjoys a less or at least the same TTR as compared to the packet without priority.*

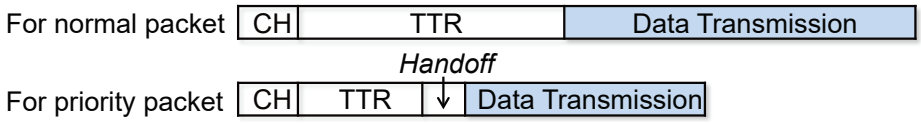


Figure 6.4: The desired CH design for priority packet.

6.2 PCH Design

6.2.1 Motivation

We first extract two facts from the analysis of the CH process in the last section. One is the homogeneous time-slot size. In all existing CH schemes, different SUs must use the equilong time slot (T) to guarantee the same unit of their hopping sequences. For example, when SU_i's hopping sequence is $\{1, 2, 1, 2, ...\}$ and SU_j's hopping sequence is $\{2, 2, 3, 3, ...\}$, it implies that the duration time on each of their hopped channel $d(channel)$ is the same. Only when $d(c_1) = d(c_2) = d(c_3) = T$, can the rendezvous-guaranteed sequences take effect. The other fact is heterogeneous transmission time. As we explained in Section II, the transmission time of a packet on different channels with different link speeds has a correspondingly different value.

Consider the source CRD in a time slot stays on a fast channel. The sojourn time of this CRD should be the same as in other time slots when it stays on other relatively slower channels. On the other hand, the packet transmission time on this fast channel is shorter than that on other slower channels. Therefore, there is ample idle time for the CRD in such a time slot. Sometimes, the idle time may be long enough for the CRD to take another transmission on another relatively fast channel.

In addition, the channel switching time and energy cost are relatively small compared with the average packet transmission time on all channels and its energy consumption, according to [93, 94]. Therefore, it is possible to ask the source CRD to hop onto more than one fast channels in one time slot. In this way, a source CRD

can keep revisiting some fast channels in some time slots during a CH process. Then, the increased visiting time of those fast channels augments their probability to become the rendezvous channel and reduces the TTR. Meanwhile, the original hopping sequence can be kept. Thus, the ETTR is reduced without worse cases (larger than original MTTR) generated. Moreover, this claim holds under any CH algorithm. We give the proof as follows.

Lemma 2. *Let CRD_i and CRD_j be the source CRD and the destination CRD, respectively. An arbitrary CH algorithm is given. Suppose under a CH process, $\tau = R^d(P(S_i), P(S_j))$. If the source CRD visits multiple channels in one time slot when feasible and the new TTR is represented by τ'. Then, $\forall d$, S_i, and S_j, $\tau' \leq R^d(P(S_i), P(S_j))$.*

Proof. Without loss of generality, let $P(S_i) = \{c_{k1}, ..., c_{ki}, ...c_{kN}\}$, where c_{ki} is the channel at the ith position in the permutation and N is the number of available channels. Denote τ^1 as the TTR when CRD_i visits one more channel, say, c_{kj}, during the ith time slot in each periodical hopping round. Then, the new sequence can be expressed as $P^1(S_i) = \{c_{k1}, ..., (c_{ki}, c_{kj}), ..., c_{kN}\}$ where $i, j = 1, 2, ..., N$ and $i \neq j$. Depending on CRD_j's hopping sequence, $P^1(S_i)$ could have the same TTR as using $P(S_i)$, or the same TTR as using $\{c_{k1}, ..., c_{kj}, ..., c_{kN}\}$. Thus, we can derive that

$$\begin{aligned}
\tau^1 =& R^d(\{c_{k1}, ..., (c_{ki}, c_{kj}), ..., c_{kN}\}, P(S_j)) \\
=& min\{\ R^d(\{c_{k1}, ..., c_{ki}, ..., c_{kN}\}, P(S_j)), \\
& \qquad R^d(\{c_{k1}, ..., c_{kj}, ..., c_{kN}\}, P(S_j))\ \} \\
\leq& R^d(\{c_{k1}, ..., c_{ki}, ..., c_{kN}\}, P(S_j)) = \tau.
\end{aligned} \tag{6.2}$$

Iteratively, we can get $\exists y > 0$,

$$\tau^y \leq \tau^{y-1} \leq ...\tau^2 \leq \tau^1. \tag{6.3}$$

Since $\tau' \in \{\tau^1, \tau^2, ..., \tau^y\}$, then, $\tau' \leq \tau$. □

In this proof, $R^d(P(S_i), P(S_j))$ represents all possible CH cases which also include the worst case. Thus, $max(\tau') \leq MTTR$. On the other hand, since $ETTR = function(\sum_d \sum_{S_i} \sum_{S_j} R^d(P(S_i), P(S_j)))$, we get $E(\tau') \leq ETTR$.

Up to now, when the source CRD proceeds CH in such a manner for packets with high priorities and follows the original CH for normal packets, the design requirement shown in (1) and the effect in Figure 6.4 can be achieved in all cases.

6.2.2 Refined Design

Next, we refine the proposed method into details to gain better performance based on some latent requirements.

6.2.2.1 Design for the Shortest Delay

From (3), when y is larger, the new CH delay decreases. In other words, the optimal PCH design should visit multiple channels in any feasible time slot as much as possible. For a source CRD, suppose $S = \{c_{k1}, c_{k2}, ..., c_{kN}\}$, and let $\boldsymbol{d} = (d_{k1}, d_{k2}, ..., d_{kN})$ represent the corresponding minimum duration time on each available channel. Since we rank the channels on a fast-to-slow order in our system model, we have

$$d_{k1} \leq d_{k2} \leq ... \leq d_{kN}. \tag{6.4}$$

For each channel c_{ki}, there is an indicator vector $\boldsymbol{x}(k_i|S)$ showing which other channel can be co-visited with c_{ki} in one time slot. $\boldsymbol{x}(k_i|S) = [l_1^{ki}, l_2^{ki}, ..., l_N^{ki}]^T$, where $l_j^{ki} \in 0, 1$, $j = 1, 2, ..., N$. If $l_j^{ki} = 1$, it means that c_{kj} can be co-visited with c_{ki} in one time slot. By default,

$$l_i^{ki} = 1. \tag{6.5}$$

As mentioned in the motivating example, the vector should satisfy

$$\boldsymbol{dx}(k_i|S) + T_s \sum_{j=1}^{N} l_j^{ki} \leq T. \tag{6.6}$$

Therefore, we can minimize the CH delay by maximizing y:

$$\begin{aligned} &\underset{\boldsymbol{X}(S)=[\boldsymbol{x}(k_1|S)\ \ldots\ \boldsymbol{x}(k_N|S)]}{\text{Maximize}} && y = \sum_{i=1}^{N}\sum_{j=1}^{N} l_j^{ki}, \\ &\text{subject to} && (4), (5), \text{and } (6). \end{aligned} \tag{6.7}$$

Since the indicator vectors of different channels are independent from each other, the problem can be simplified as getting the maximum $\sum_{j=1}^{N} l_j^{ki}$ for a given channel c_{ki}. Consider assign the value '1' to either l_j^{ki} or l_{j+1}^{ki}. Then, the combination of (l_j^{ki}, l_{j+1}^{ki}) could be either $(1, 0)$ or $(0, 1)$. Take them into (6) separately. Because of (4), the left side of (6) calculated with the $(1, 0)$ combination is less than that with the $(0, 1)$ combination. A smaller value of the left side means that we can possibly assign more '1's to $\boldsymbol{x}(k_i|S)$. Iteratively, the optimal solution is to assign '1' to the element in the vector from left to right until (6) cannot be satisfied. Finally, for a given channel c_{ki}, its co-visiting channels $Co(c_{ki}|S)$ can be expressed by the derived $\boldsymbol{x}(k_i|S)$:

$$Co(c_{ki}|S) = \boldsymbol{S} \cdot \boldsymbol{x}(k_i|S)^T. \tag{6.8}$$

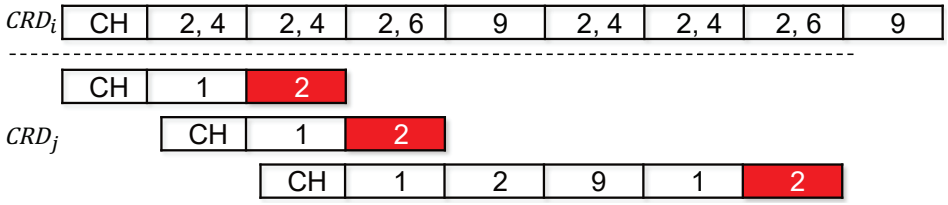


Figure 6.5: A PCH example.

Consider the same example in Figure 6.2. Suppose $d_i = i$, $Ts = 1$, and $T = 10$. Using the optimal PCH construct method given above, the result is shown in Figure

6.5. Now, we have $R^0 = 2$, $R^1 = 3$, and $R^2 = 7$. Each delay is smaller than the previous result (5, 4, and 8, respectively).

6.2.2.2 Design for Same-Role Cases

In previous discussion, we suppose that the role of the destination CRD is either a listening CRD or a source CRD with normal packets to send. However, in practical IoT, the destination CRD could also have priority packets to transmit to another CRD, namely the same-role case.

When the destination CRD is also in a PCH process, both CRDs can visit multiple channels in one time slot. If there is no proper mechanism, even when they are in the potential rendezvous time slot, they may miss each other due to different presence time on their potential rendezvous channel, as illustrated in Figure 6.6.

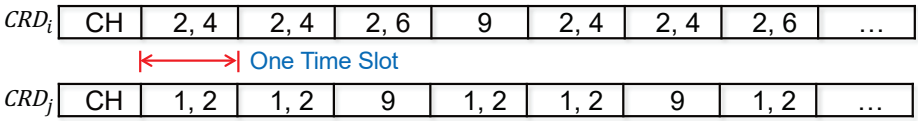


Figure 6.6: Rendezvous failure in the same-role case.

In order to guarantee their rendezvous in this case, for each channel, there should be a unified hopping moment for all CRDs in a time slot. Note that the ACS of each CRD is a subset of the total channel set in the primary network. Imagine that there is *a virtual CRD* in the network *whose ACS contains all the channels in the primary network*, i.e., $S = \boldsymbol{C}$. Let this CRD adopt our PCH design. For a given channel c_i, its co-visiting channels $Co(c_i|\boldsymbol{C})$ in a time slot can be derived by (8). Arrange the hopping sequence within a time slot by channel indexes. Then, the moment the CRD hops onto c_i in the time slot is

$$\tau_s(c_i) = \sum_{j=1}^{i-1}(T_s + d_j)l_j^i. \tag{6.9}$$

This is also the unified visiting moment of c_i in PCH. Since for any other CRD, its

ACS is a subeset of $\boldsymbol{C}$, its co-visiting channels of c_i are also a subset of $Co(c_i|\boldsymbol{C})$. In this way, a CRD can confirm the visiting moment of c_i in a time slot.

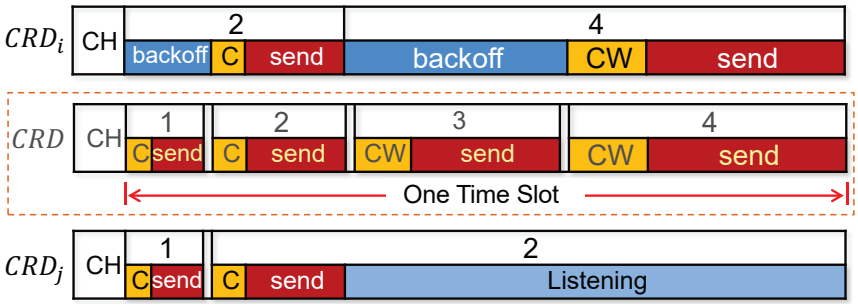


Figure 6.7: MAC for both sender soles.

After finding the right moment to hop onto each co-visiting channel in a time slot independently, the rendezvous pair in the same-role scenario can perform even better due to the destination CRD's co-visiting contribution. It is equivalent to enlarge the value of y in Lemma 1's proof, which further decreases the ETTR and MTTR of the original CH algorithm. Again, we use the same example to illustrate our design as shown in Figure 6.7. In the figure, CW refers to the *contention window* designed in the CRD MAC layer [91]. Also, note that it is *NOT* necessary to require the time slots or the mini time slots of each SUs to be *strictly synchronized* [74]. Both CRD_i and CRD_j visit channel 1, channel 2, or channel 4 at the corresponding unified visiting moment (illustrated using the virtual-ideal CRD in the middle) in their first time slot. In this case, they will rendezvous on channel 2 with $R^0 = 1$ which is further reduced as compared to the case without the destination SU's participation in PCH when $R^0 = 2$.

Note that a shorter CW can be assigned to the CRD with a higher priority packet. In this way, the packet with a higher priority has a higher chance to be sent first.

6.2.3 Protocol Details

Algorithm 1 gives the entire protocol showing our hopping-channel reconstruct design based on packet priority. As we can see, our proposed PCH is very easy for implementation for any given CH algorithm. Meanwhile, the design is also indepen-

Algorithm 9: The PCH protocol for $\mathrm{CRD_x}$

Require: ACS of $\mathrm{CRD_x}$ S_x, data packet, and adopted CH algorithm;

Rank channels in S_x according to their link speeds on a fast-to-slow order;
Generate the hopping sequence $P(S_x)$;
$\tau = 0$;
while *not rendezvous* **do**
    $\tau = \tau + 1$;
    if *no packet* **then**
        Hop onto $P(S_x)(\tau)$;
        Listening to the channel;
    if *packet without priority* **then**
        Hop onto $P(S_x)(\tau)$;
        CW; Send message; Listening;
    if *packet with priority* k **then**
        $c_j = P(S_x)(\tau)$;
        Get $Co-ch. = Co(c_j|S_x)$ using (6.8);
        i=1;
        while *not rendezvous &* $Co-ch.(i) \in Co(c_j|S_x)$ **do**
            Get $t = \tau_s(Co-ch.(i))$ using (6.9);
            Hop onto $Co-ch.(i)$ at moment t;
            $CW(k)$; Send message; Listening;
            $i = i + 1$;
        Handoff to their kth fastest common channel;
Data Transmission;

dent of the CRDs with non-priority packets.

6.3 Performance Evaluation

In our simulation, 1) CRDs and regular devices are randomly distributed in the simulation area; 2) each regular device is randomly assigned a channel when a new packet needs to be transmitted; 3) source CRDs are randomly selected, and their packets are randomly assigned with or without priority; and 4) each source CRD randomly chooses a CRD within its transmission range as its destination CRD. The parameters used in our simulation are listed in Table 6.1.

Table 6.1: Simulation parameters for PCH

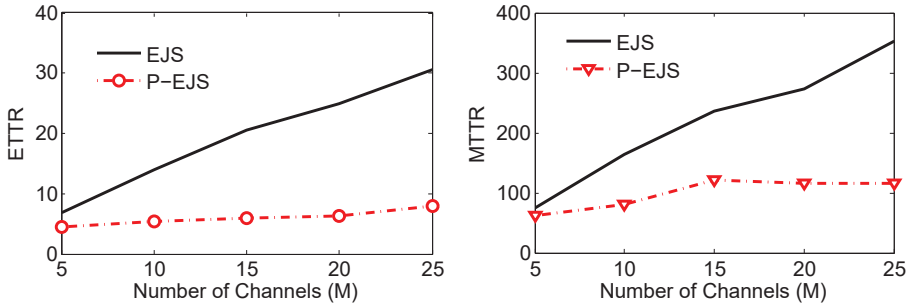
Simulation time	10000 slots
Simulation area	50 m $\times$ 50 m
Number of usual/cognitive devices	100/50
CRD sensing radius	10/7 m
Average packet arrival rate	50 pkt/s
Average packet size	50 slots
Channel data rate	$10/i$ Mbps on channel i
The transmit power	$15 - 0.5i$ dbm on channel i [94]
Channel switching time/energy	200 μs / 20 μJ [93]

6.3.1 Performance Comparison

We adopt a state-of-the-art CH algorithm, Enhanced Jump-Stay (EJS) [19], to evaluate our PCH performance. The ETTR and MTTR of both the EJS and the priority-based EJS (P-EJS) are compared in Figure 6.8. Both these protocols can achieve a 100% successful rendezvous rate. The results in Figure 6.8 validate our proof that CRDs with a priority packet can always enjoy a lower spectrum access delay including the worst case. Moreover, PCH has a better resistance to the primary network change, e.g., when the number of total channels increases.

6.3.2 Impact of Different Factors

Figure 6.9 shows the ETTR performance of both the PCH and the PCH under the same-role scenario with different common available channels. We use γ to represent



(a) ETTR performance. (b) MTTR performance.

Figure 6.8: TTR vs. M in priority-based EJS.

the common available channel rate of a CRD, $\gamma = \frac{\#\text{of common available channels}}{\#\text{of available channels}}$.

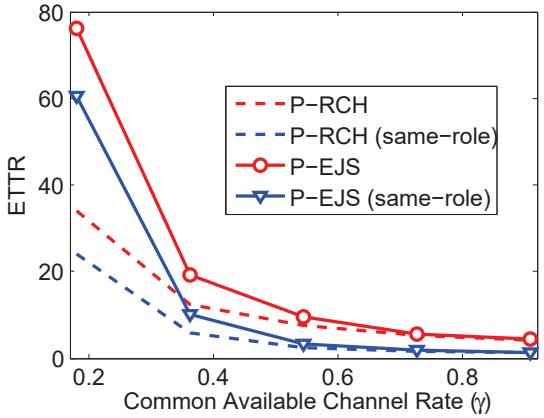


Figure 6.9: Impact of the same-role CH.

6.3.2.1 Impact of the Common Available Channel Rate

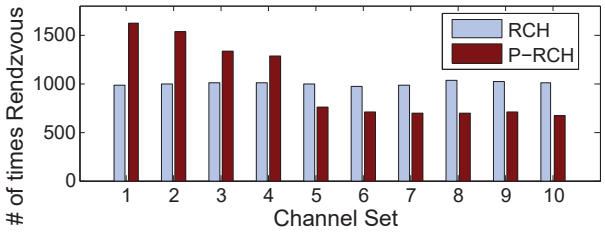
When γ increases, the ETTR of both PCHs decreases, which satisfies the initial goal of CH. However, the ETTR does not change much when $\gamma > 0.5$. This is because that the ETTR performance mainly depends on the involvement of the fast channels not the number of common available channels. When there are a sufficient number of fast channels, the ETTR performance is dominated by their re-occurrences.

6.3.2.2 Impact of Same-Role Cases

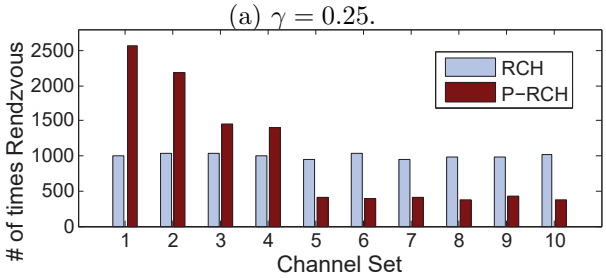
When both CRDs have priority packets to send, PCH helps them rendezvous with each other with even less delay. From Figure 6.9, PCH always performs better under the same-role scenario no matter using the Random CH (RCH) or EJS algorithm.

Since our design enables same-role CRDs to have the synchronous visiting moment on the common channel within a time slot, the CRD pair enjoys another mini-CH round in each time slot.

6.3.3 Rendezvous Channel Distribution



(a) $\gamma = 0.25$.



(b) $\gamma = 0.75$.

Figure 6.10: Rendezvous-channel distribution with different γ.

We record the number of times for each channel becoming the rendezvous channel. The rendezvous-channel distribution is shown in Figure 6.10. We summarize: *i)* the chance to be the rendezvous channel is evenly distributed over all channels with RCH under various network conditions, which accords with its random feature; *ii)* for a CRD with a priority packet to send, it is more likely to rendezvous on those fast channels (channel 1-4), especially when γ is high, which validates our analysis; and *iii)* since there is no need to handoff if the rendezvous channel is fast, PCH reduces the possible handoff delay.

6.3.4 Energy Consumption Performance

From Figure 6.11, when there are more channels in the system with a small γ, energy cost increases under the PCH since the TTR is large. Nevertheless, when the number of channels is small, the consumption of priority spectrum access is less than

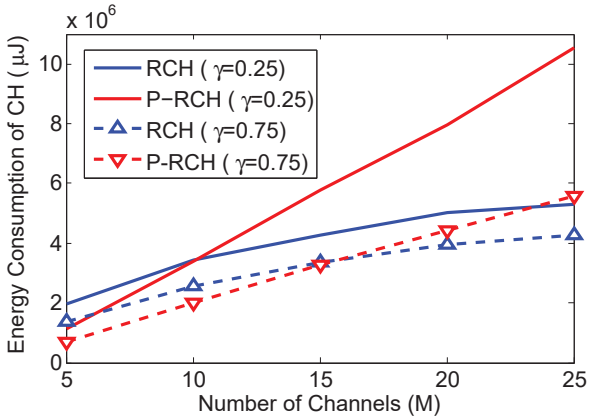


Figure 6.11: Energy consumption during CH for different packets.

that of regular access, despite the low γ. On the other hand, when γ is large, the energy consumption of PCH is very close to normal RCH due to the benefits of the reduced TTR by PCH. In this case, PCH can be used without constraint when the number of channels is less than 15.

CHAPTER 7: CONCLUSION

7.1 Completed Work

In this dissertation, four MAC schemes, practical self-adaptive rendezvous-control framework (PSA), slot-asynchronous handshake protocol, contention window-based deadlock-free (CWDF) handshake protocol, and self-adaptive optimal fragmentation protocol (SAOF), are proposed to form the fast and energy-efficient spectrum access framework for CD2DNs. Three novel schemes are proposed to realize the fast neighbor discovery, heterogeneous rendezvous, and priority-based spectrum access functions in general, MWB, and IoT/5G scenarios, respectively.

The proposed PSA scheme addressed the challenges of practical rendezvous in CRNs for the first time. Through establishing the rigorous probabilistic model to analyze all practical factors which influence the performance, a fully self-adaptive rendezvous framework PSA is proposed without imposing impractical assumptions. PSA can provide the maximum throughput with interference avoidance and rendezvous delay compensation by intelligently reasoning the environment and dynamically updating the optimal STTR as well as the size and content of the ACS. Simulation results validate our analysis and the merits of PSA in terms of overwhelming throughput against non-PAS for different CH algorithms under various network scenarios.

In the proposed slot-asynchronous handshake scheme, we developed probabilistic models for each possible factor which may influence the successful handshake during blind rendezvous. Then, according to the analysis of each factor, we proposed corresponding schemes and integrated them into a novel MAC protocol with the optimal size of a time slot in terms of the shortest TTH. Simulation results verify the optimality of the time slot and show that our design can maintain the rendezvous

performance at the theoretical level in different practical scenarios.

In the proposed CWDF handshake scheme, the impact of deadlock on blind rendezvous performance in CRNs is addressed for the first time. A contention window based deadlock-free rendezvous protocol without imposing the role-preassigned assumption is proposed which well adapts to the nature of CH systems. In addition, we developed the probabilistic model for throughput analysis in both 2-SU and multi-SU scenarios. By deriving the corresponding optimal size of the contention window, our proposed protocol can provide high throughput in both moderate and saturated networks. Additionally, our protocol does not affect the performance of the incorporated CH scheme in terms of ETTR.

The proposed SAOF protocol addresses the SU packet fragmentation issue in CRNs for the first time. Compared with traditional wireless networks, we regard PU interference and handoff delay as the unique fragmentation factors for the desired network. Probabilistic retransmission model is established based on these novel factors. In addition, we associated our model with the time-varying PU activities in a practical way in order to learn the related parameters dynamically. Then, we proposed SAOF for the optimal fragmentation in terms of maximizing the throughput. Simulation results have verified both the correctness of the proposed mathematical model and the optimality of the fragmentation. The up-to-date parameters estimated by SAOF can provide vital information for the design in higher layer functions.

SUBSET is proposed by joint design of channel selection and channel hopping. The corresponding analytical models and application models are also proposed for calculating ETTR/MTTR and adjusting with the information change throughout the network life, respectively. The TTR analytical model can support our CH algorithm to work independently for any subset ACS pairs. From both analytical and simulation results, SUBSET is more robust than other CH schemes against false alarm and can achieve fast blind rendezvous with $O(1)$ ETTR and MTTR. To the best of our

knowledge, this is the first work that can reduce TTR to $O(1)$ in guaranteed blind rendezvous design. In addition, under SUBSET, for the first time, TTR decreases with the increasing number of available channels in the network. Simulation results also show that SUBSET can achieve fast rendezvous in different spectrum and network environments with fewer operation requirements.

2D-HR is proposed which can dynamically adapt to network conditions and achieve heterogeneous rendezvous (pair-wise, any-wise, and multi-wise) in MWB-CRNs. The proposed design is based on a 2D hopping structure which gives the flexibility of altering the hopping slot size and enables rendezvous at a hopping frame level. Both mathematical proof and simulation results showed that 2D-HR can achieve all types of rendezvous with guarantees in a fast speed in MWB-CRNs. In addition, our design can also reduce the energy consumption of SUs significantly. Moreover, 2D-HR can also work on a single spectrum band with better performance than existing works, and in asynchronous scenarios with neither degrading on performance nor changing on the protocol. For the first time, heterogeneous rendezvous is achieved in MWB-CRNs.

In the proposed PCH scheme, the efficient and adaptive protocol can be integrated with any existing CH algorithm. The proposed design is based on co-visiting fast channels within each feasible time slot. Both mathematical proof and simulation results showed that PCH can achieve shorter spectrum access delay for packets with higher priority in cognitive D2D networks. Besides, we further refined and revised our design to deal with different practical issues. Simulation results validated our design goals in various IoT/5G environments. For the first time, the priority-based spectrum access is achieved in cognitive D2D networks for IoT/5G.

To sum up, since the MAC layer failures during the spectrum access in CD2DNs can be eliminated under the proposed MAC schemes, the spectrum access delay and energy consumption in CD2DNs can be reduced significantly as compared to existing CRN solutions. In addition, various functions can be achieved based on our framework

in various CD2DN application scenarios.

7.2 Future Work

Based on the proposed MAC designs, two issues can be considered in the future work:

- CD2DN Security: since our proposed protocols have some unique features against malicious behaviors such as channel hopping without a pre-shared secret, and data protection from MAC layer inherently, I plan to well utilize these features and extend our protocols to deal with unique security issues identified in IoT.

- Data Mining Enabled CD2DNs: there are lots of research holes in my current research work that can be filled with data mining. By analyzing or training data from different network layers, we can improve the performance of different applications, such as 3D routing optimization via social analysis of secondary users, battery management via spectrum and network analysis, and cooperating with body area networks for cognitive health systems.

7.3 Published and Submitted Work

1. Xingya Liu and Jiang Xie, "SUBSET: Fast and Energy-efficient Neighbor Discovery for Cognitive D2D Communications in IoT," *submitted to IEEE Transactions on Mobile Computing*, Nov 2016.

2. Xingya Liu and Jiang Xie, "PSA: A Practical Self-adaptive Rendezvous Control Framework for Cognitive Radio Networks," *submitted to IEEE Transactions on Mobile Computing*, Dec 2016.

3. Xingya Liu and Jiang Xie, "Slot Asynchronous and Synchronous MAC for Handshake in Cognitive Radio Networks," *submitted to IEEE Transactions on Wireless Communications*, Dec 2016.

4. Xingya Liu and Jiang Xie, "A 2D Channel Hopping Protocol for Heterogeneous Spectrum Access in 5G Cognitive Radio Networks," *in preparation to submit for journal publication.*

5. Xingya Liu and Jiang Xie, "PCH: A Priority-based Spectrum Access Protocol for Cognitive D2D Communications in IoT," *in preparation to submit for journal publication.*

6. Xingya Liu and Jiang Xie, "Priority-based Spectrum Access in Cognitive D2D Networks for Internet of Things," *Proc. IEEE ICC*, May 2017.

7. Xingya Liu and Jiang Xie, "A 2D Heterogeneous Rendezvous Protocol for Multi-wideband Cognitive Radio Networks," *Proc. IEEE INFOCOM*, May 2017.

8. Xingya Liu and Jiang Xie, "A Self-Adaptive Optimal Fragmentation Protocol for Multi-Channel Cognitive Radio Ad Hoc Networks," *Proc. IEEE GLOBECOM*, December 2016.

9. Xingya Liu and Jiang Xie, "Contention Window-Based Deadlock-Free MAC for Blind Rendezvous in Cognitive Radio Ad Hoc Networks," *Proc. IEEE GLOBECOM*, December 2015.

10. Xingya Liu and Jiang Xie, "SUBSET: A Joint Design of Channel Selection and Channel Hopping for Fast Blind Rendezvous in Cognitive Radio Ad Hoc Networks," *Proc. IEEE SECON*, June 2015.

11. Xingya Liu and Jiang Xie, "A Slot-Asynchronous MAC Protocol Design for Blind Rendezvous in Cognitive Radio Networks," *Proc. IEEE GLOBECOM*, December 2014.

12. Xingya Liu and Jiang Xie, "A Practical Self-Adaptive Rendezvous Protocol in Cognitive Radio Ad Hoc Networks," *Proc. IEEE INFOCOM*, April 2014.

REFERENCES

[1] FCC, "Notice of proposed rule making and order," no. 03-222, Dec. 2003.

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *COMPUTER NETWORKS*, vol. 50, pp. 2127–2159, 2006.

[3] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, pp. 810–836, 2009.

[4] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *Proc. IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2008.

[5] K. Bian, J.-M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proc. ACM MobiCom*, 2009.

[6] L. Jiao and F. Li, "A single radio based channel datarate-aware parallel rendezvous MAC protocol for cognitive radio networks," in *Proc. IEEE LCN*, 2009.

[7] C.-F. Shih, T. Y. Wu, and W. Liao, "DH-MAC: A dynamic channel hopping MAC protocol for cognitive radio networks," in *Proc. IEEE ICC*, 2010.

[8] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pp. 755–760, 2010.

[9] J. Shin, D. Yang, and C. Kim, "A channel rendezvous scheme for cognitive radio networks," *IEEE Communications Letters*, vol. 14, no. 10, pp. 954–956, 2010.

[10] M. Altamimi, K. Naik, and X. Shen, "Parallel link rendezvous in ad hoc cognitive radio networks," in *Proc. IEEE Global Telecommunications Conference (Globecom)*, 2010.

[11] C. Xin, M. Song, L. Ma, and C.-C. Shen, "Performance analysis of a control-free dynamic spectrum access scheme," *IEEE Trans. Wireless Communications*, vol. 10, no. 12, pp. 4316–4323, 2011.

[12] K. Bian and J.-M. Park, "Asynchronous channel hopping for establishing rendezvous in cognitive radio networks," in *Proc. IEEE INFOCOM*, pp. 236–240, 2011.

[13] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Trans. Mobile Computing*, vol. 10, no. 2, pp. 216–227, 2011.

[14] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE INFOCOM*, pp. 2444–2452, 2011.

[15] M. Abdel Rahman, H. Rahbari, and M. Krunz, "Adaptive frequency hopping algorithms for multicast rendezvous in DSA networks," in *IEEE International Symposium on DYSPAN*, 2012.

[16] G.-Y. Chang, W.-H. Teng, H.-Y. Chen, and J.-P. Sheu, "Novel channel-hopping schemes for cognitive radio networks," *IEEE Trans. Mobile Computing*, 2012.

[17] J. Kim, Y. Baek, J. Yun, K. Cho, K. Lee, and J. Han, "A repeated group sequence rendezvous scheme for cognitive radio networks," in *Proc. Spring Congress on Engineering and Technology (S-CET)*, 2012.

[18] R. Gandhi, C.-C. Wang, and Y. C. Hu, "Fast rendezvous for multiple clients for cognitive radios using coordinated channel hopping," in *Proc. IEEE SECON*, 2012.

[19] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1742–1745, 2013.

[20] I. Chuang, H. Wu, and Y. Kuo, "A fast blind rendezvous method by alternate hop-and-wait channel hopping in cognitive radio networks," *IEEE Transcations on Mobile Computing*, vol. 13, no. 10, pp. 2171–2184, 2014.

[21] L. Chen, K. Bian, C. Liu, J.-M. J. Park, and X. Li, "A group-theoretic framework for rendezvous in heterogeneous cognitive radio networks," in *Proc. MobiHoc*, pp. 165–174, 2014.

[22] H. Su and X. Zhang, "Channel-hopping based single transceiver MAC for cognitive radio networks," in *Proc. Conference on Information Sciences and Systems (CISS)*, pp. 197–202, 2008.

[23] B. Hamdaoui and K. Shin, "OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks," *IEEE Trans. on Mobile Computing*, vol. 7, no. 8, p-p. 915–930, 2008.

[24] S. Jha, U. Phuyal, M. Rashid, and V. Bhargava, "Design of OMC-MAC: An opportunistic multi-channel MAC with QoS provisioning for distributed cognitive radio networks," *IEEE Trans. on Wireless Communications*, vol. 10, no. 10, pp. 3414–3425, 2011.

[25] S. Debroy, S. De, and M. Chatterjee, "Contention based multichannel MAC protocol for distributed cognitive radio networks," *IEEE Trans. on Mobile Computing*, vol. 13, no. 12, pp. 2749–2762, 2014.

[26] Y. Song, "Optimal secondary user packet size in mobile cognitive radio networks under fading channels," in *Proc. IEEE INFOCOM*, pp. 163–171, 2015.

[27] C. Luo, F. R. Yu, H. Ji, and V. C. M. Leung, "Cross-layer design for TCP performance improvement in cognitive radio networks," *IEEE Trans. on Vehicular Technology*, vol. 59, pp. 2485–2495, 2010.

[28] A. Jamal, C. K. Tham, and W. C. Wong, "Dynamic packet size optimization and channel selection for cognitive radio sensor networks," *IEEE Trans. on Cognitive Communications and Networking*, vol. PP, 2016.

[29] L. Kleinrock, ed., *Queueing Systems Volume I: Theory.* New York: A Wiley-Interscience Publication, 1975.

[30] K. Bian and J.-M. Park, "Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks," *IEEE Transcations on Mobile Computing*, vol. 12, no. 7, pp. 1294–1307, 2013.

[31] J.-P. Sheu, C.-W. Su, and G.-Y. Chang, "Asynchronous quorum-based blind rendezvous schemes for cognitive radio networks," *IEEE Trans. Commun.*, vol. 64, no. 3, pp. 918–930, 2016.

[32] Y. Song and J. Xie, "QB^{2}IC: A QoS-based broadcast protocol under blind information for multihop cognitive radio ad hoc networks," *IEEE Trans. Vehicular Technology*, vol. 63, no. 3, pp. 1453–1466, 2013.

[33] W. Ren, Q. Zhao, and A. Swami, "Power control in cognitive radio networks: How to cross a multi-lane highway," *IEEE Transcations on Selected Area in Communications*, vol. 27, no. 7, pp. 1283–1296, 2009.

[34] J. Li and J. Xie, "A power control protocol to maximize the number of common available channels between two secondary users in cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2013.

[35] Q. Chen, Y.-C. Liang, M. Motani, and W.-C. Wong, "A two-level MAC protocol strategy for opportunistic spectrum access in cognitive radio networks," *IEEE Transcations on Vehicular Techology*, vol. 60, pp. 2164–2180, 2011.

[36] S.-Y. Lien, C.-C. Tseng, and K.-C. Chen, "Carrier sensing based multiple access protocols for cognitive radio networks," in *Proc. IEEE ICC*, pp. 3208–3214, 2008.

[37] X. Zhang and H. Su, "Opportunistic spectrum sharing schemes for CDMA-based uplink MAC in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 716–730, 2011.

[38] J. Xiang, Y. Zhang, and T. Skeie, "Medium access control protocols in cognitive radio networks," *Wireless Communication and Mobile Computing*, vol. 10, no. 1, pp. 31–49, 2010.

[39] A. De Domenico, E. Strinati, and M. Di Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Communications Surveys and Tutorials*, vol. 14, pp. 21–44, 2012.

[40] Y. Zhao, M. Song, and C. Xin, "FMAC: A fair MAC protocol for coexisting cognitive radio networks," in *Proc. IEEE INFOCOM*, 2013.

[41] D. Hu and S. Mao, "Design and analysis of a sensing error-aware MAC protocol for cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2009.

[42] M. Felegyhazi, M. Cagalj, and J.-P. Hubaux, "Efficient MAC in cognitive radio systems: A game-theoretic approach," *IEEE Transcations on Wireless Communications*, vol. 8, pp. 1984–1995, 2009.

[43] G. Joshi and S. W. Kim, "An enhanced synchronized MAC protocol for cognitive radio networks," in *Proc. IEEE WiCOM*, 2011.

[44] M. Shah, S. Zhang, and C. Maple, "An analysis on decentralized adaptive MAC protocols for cognitive radio networks," in *Proc. IEEE ICAC*, 2012.

[45] L. T. Tan and L. B. Le, "Distributed MAC protocol for cognitive radio networks: Design, analysis, and optimization," *IEEE Transcations on Vehicular Techology*, vol. 60, pp. 3990–4003, 2011.

[46] M. Kalil, A. Puschmann, and A. Mitschele-Thiel, "SWITCH: A multichannel MAC protocol for cognitive radio ad hoc networks," in *Proc. IEEE Vehicular Technology Conference (VTC)*, 2012.

[47] H. Hussein, H. Elsayed, and S. Elramly, "Performance evaluation of cognitive radio network predictive MAC (P-MAC) access algorithm and its enhancement," in *Proc. IEEE ICOIN*, 2013.

[48] S. Jha, U. Phuyal, M. Rashid, and V. Bhargava, "Design of OMC-MAC: An opportunistic multi-channel MAC with QoS provisioning for distributed cognitive radio networks," *IEEE Transcations on Wireless Communications*, vol. 10, pp. 1984–1995, 2011.

[49] S. K. Timalsina, S. Moh, llyong Chung, and M. Kang, "A concurrent access MAC protocol for cognitive radio ad hoc networks without common control channel," *EURASIP Journal on Advances in Signal Processing*, vol. 69, 2013.

[50] M. Altamimi, K. Naik, and X. Shen, "Parallel link rendezvous in ad hoc cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2010.

[51] M. C. Oto and O. B. Akan, "Energy-efficient packet size optimization for cognitive radio sensor networks," *IEEE Trans. on Mobile Computing*, vol. 11, pp. 1544–1553, 2012.

[52] A. E. Shafie and T. Khattab, "Throughput maximization via adjusting packet size of a buffered cognitive radio user," in *Proc. IEEE PIMRC*, pp. 1024–1029, 2014.

[53] C. Majumdar, N. Sridhar, and S. N. Merchant, "Variable rate m-QAM assisted packet size optimization for cognitive radio and mimo cognitive radio based sensor networks," in *Proc. IEEE MILCOM*, pp. 422–427, 2014.

[54] D. Chen, H. Ji, and V. C. M. Leung, "Distributed best-relay selection for improving TCP performance over cognitive radio networks: A cross-layer design approach," *IEEE Journal on Selected Areas in Communications*, vol. 30, pp. 716–730, 2012.

[55] E. Askari and S. Assa, "Full-duplex cognitive radio with packet fragmentation," in *IEEE WCNC*, pp. 1502–1507, 2014.

[56] A. Giannoulis, P. Patras, and E. W. Knightly, "Mobile access of wide-spectrum networks: Design, deployment and experimental evaluation," in *Proc. IEEE INFOCOM*, 2013.

[57] G. Hattab and M. Ibnkahla, "Multiband spectrum access: Great promises for future cognitive radio networks," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 282–306, 2014.

[58] H. Sun, A. Nallanathan, C.-X. Wang, and Y. Chen, "Wideband spectrum sensing for cognitive radio networks: a survey," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 74–81, 2013.

[59] Y. Song and J. Xie, "Optimal power control for concurrent transmissions of location-aware mobile cognitive radio ad hoc networks," in *Proc. IEEE GLOBECOM*, 2009.

[60] 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, 2013.

[61] P. B. Nayak, S. Verma, and P. Kumar, "Multiband fractal antenna design for cognitive radio applications," in *Proc. ICSC*, 2013.

[62] J. Li and J. Xie, "A new communication framework for wide-band cognitive radio networks," in *Proc. IEEE SECON*, 2014.

[63] M. Shah, S. Zhang, and C. Maple, "Cognitive radio networks for internet of things: Applications, challenges and future," in *Automation and Computing (ICAC)*, pp. 1–6, 2013.

[64] E. Hossain and M. Hasan, "5g cellular: key enabling technologies and research challenges," *IEEE Instrumentation and Measurement Magazine*, vol. 18, no. 3, pp. 11–21, 2015.

[65] C. Cordeiro, K. Challapali, and M. Ghosh, "Cognitive PHY and MAC layers for dynamic spectrum access and sharing of tv bands," in *Proc. First Intl Workshop Technology and Policy for Accessing Spectrum (TAPAS)*, 2006.

[66] L.-C. Wang, C.-W. Wang, and C.-J. Chang, "Optimal target channel sequence design for multiple spectrum handoffs in cognitive radio networks," *IEEE Trans. Communications*, vol. 60, September 2012.

[67] Y. Song and J. Xie, "ProSpect: A proactive spectrum handoff framework for cognitive radio ad hoc networks without common control channel," *IEEE Trans. Mobile Computing*, vol. 11, no. 7, 2012.

[68] S.-T. Sheu and T.-F. Sheu, "DBASE: a distributed bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LAN," in *Proc. IEEE INFOCOM*, vol. 3, 2001.

[69] M. Juang, K.-C. Wang, and J. Martin, "A measurement study on link capacity of a high stress IEEE 802.11b/g network," in *Proc. International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, 2008.

[70] T. Sugimoto, N. Komuro, H. Sekiya, S. Sakata, and K. Yagyu, "Maximum throughput analysis for RTS/CTS-used IEEE 802.11 DCF in wireless multi-hop networks," in *Proc. International Conference on Computer and Communication Engineering (ICCCE)*, pp. 1–6, 2010.

[71] M. Chowdhury, A. Asaduzzaman, M. F. Kader, and M. O. Rahman, "Design of an efficient MAC protocol for opportunistic cognitive radio networks," in *International Journal of Computer Science and Information Technology (IJCSIT)*, 2012.

[72] S. Yin, D. Chen, Q. Zhang, and M. Liu, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Trans. Mobile Computing*, 2012.

[73] G. Woo, P. Kheradpour, D. Shen, and D. Katabi, "Beyond the bits: cooperative packet recovery using physical layer information," in *Proc. ACM MobiCom*, pp. 147–158, 2007.

[74] X. Liu and J. Xie, "A slot-asynchronous MAC protocol design for blind rendezvous in cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2014.

[75] Y. Song and J. Xie, "A novel unified analytical model for broadcast protocols in multi-hop cognitive radio ad hoc networks," *IEEE Trans. Mobile Computing*, 2013.

[76] P. Wang, L. Xiao, S. Zhou, and J. Wang, "Optimization of detection time for channel efficiency in cognitive radio systems," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 111–115, 2007.

[77] X. Liu and J. Xie, "A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks," in *Proc. IEEE INFOCOM*, 2014.

[78] Y. Song and J. Xie, "Performance analysis of spectrum handoff for cognitive radio ad hoc networks without common control channel under homogeneous primary traffic," in *Proc. IEEE INFOCOM*, 2011.

[79] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks," in *Proc. IEEE GLOBECOM*, 2002.

[80] J. Shim, Q. Cheng, and V. Sarangan, "Cooperative sensing with adaptive sensing ranges in cognitive radio ad-hoc networks," in *Proc. IEEE CROWNCOM*, 2010.

[81] T. S. Rappaport, *Wireless Communications: Principles and Practice*. New Jersey: Prentice Hall, 1996.

[82] J. Unnikrishnan and V. Veeravalli, "Cooperative sensing for primary detection in cognitive radio," *IEEE Journals of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 18–27, 2008.

[83] Y. Zhang, L. Lazos, K. Chen, B. Hu, and S. Shivaramaiah, "FD-MMAC: Combating multi-channel hidden and exposed terminals using a single transceiver," in *Proc. IEEE INFOCOM*, 2014.

[84] S. Gollakota and D. Katabi, "ZigZag decoding: combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM*, pp. 159–170, 2008.

[85] J. Deng, B. Liang, and P. Varshney, "Tuning the carrier sensing range of IEEE 802.11 MAC," in *Proc. IEEE GLOBECOM*, 2004.

[86] F.-Y. Hung and I. Marsic, "Effectiveness of physical and virtual carrier sensing in IEEE 802.11 wireless ad hoc networks," in *Proc. IEEE WCNC*, 2007.

[87] USC/ISI, "Network Simulator 2 (NS2)," http://www.isi.edu/nsnam/ns/.

[88] X. Liu and J. Xie, "A self-adaptive optimal fragmentation protocol for multi-channel cognitive radio ad hoc networks," in *Proc. IEEE GLOBECOM*, 2016.

[89] C. W. Wang, L. C. Wang, and F. Adachi, "Modeling and analysis for reactive-decision spectrum handoff in cognitive radio networks," in *Proc. IEEE GLOBECOM*, 2010.

[90] D. Wang, Z. Yang, and Y. Song, "A novel spectrum sensing scheduling scheme for PU and SU signal differentiation in CR networks," in *Proc. IEEE ICC*, 2015.

[91] X. Liu and J. Xie, "Contention window-based deadlock-free mac for blind rendezvous in cognitive radio ad hoc networks," in *Proc. IEEE GLOBECOM*, 2015.

[92] X. Liu and J. Xie, "SUBSET: A joint design of channel selection and channel hopping for fast blind rendezvous in cognitive radio ad hoc networks," in *Proc. IEEE SECON*, pp. 426–434, 2015.

[93] S. Agarwal and S. De, "Impact of channel switching in energy constrained cognitive radio networks," *IEEE Communications Letters*, vol. 19, no. 6, pp. 977–980, 2015.

[94] Cisco, "MX60/64W and Z1." https://documentation.meraki.com/MR/Radio$_S ettings$.