Rendezvous for cognitive radio networks

Lu Yu
Hong Kong Baptist University

Follow this and additional works at: https://repository.hkbu.edu.hk/etd_oa

Recommended Citation
https://repository.hkbu.edu.hk/etd_oa/214

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at HKBU Institutional Repository. It has been accepted for inclusion in Open Access Theses and Dissertations by an authorized administrator of HKBU Institutional Repository. For more information, please contact repository@hkbu.edu.hk.
Rendezvous for Cognitive Radio Networks

Lu YU

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

Doctor of Philosophy

Principal Supervisor: Prof. Yiu Wing Leung

Hong Kong Baptist University

August 2015
Declaration

I declare that this thesis represents my own work which has not previously includ-
ed in any thesis, dissertation or report submitted to any institution for a degree,
diploma or other qualification.

Signature:____________________

Date: Aug 2015
Abstract

With the traditional static spectrum management, a significant portion of the licensed spectrum is underutilized in most of time while the unlicensed spectrum is over-crowded due to the growing demand for wireless radio spectrum from exponential growth of various wireless devices. Dynamic Spectrum Access utilizes the wireless spectrum in a more intelligent and flexible way. Cognitive radios are a promising enabler for Dynamic Spectrum Access because they can sense and access the idle channels. With cognitive radios, the unlicensed users (SUs) can opportunistically identify and access the vacant portions of the spectrum of the licensed users (PUs). In cognitive radio networks (CRNs), multiple idle channels may be available to SUs. If two or more SUs want to communicate with each other, they must select a channel which is available to all of them. The process of two or more SUs to meet and establish a link on a commonly-available channel is known as rendezvous.

1) Multiple Radios for Fast Rendezvous in CRNs: The existing works on rendezvous implicitly assume that each cognitive user is equipped with one radio (i.e., one wireless transceiver). As the cost of wireless transceivers is dropping, this feature can be exploited to significantly improve the rendezvous performance at low cost. We investigate the rendezvous problem in CRNs where cognitive users are equipped with multiple radios and different users may have different numbers of radios. We first study how the existing rendezvous algorithms can be generalized to use multiple radios for faster rendezvous. We then propose a new rendezvous algorithm, called role-based parallel sequence (RPS), which specifically exploits multiple radios for more efficient rendezvous. Our basic idea is to let the cognitive users stay in a specific channel in one dedicated radio and hop on the available channels with parallel sequences in the remaining general radios. We prove that RPS provides guaranteed rendezvous (i.e., rendezvous can be completed within a finite time) and
derive the upper bounds on the maximum time-to-rendezvous (TTR) and the expected TTR. The simulation results show that i) multiple radios can cost-effectively improve the rendezvous performance, and ii) the proposed RPS algorithm performs better than the ones generalized from the existing algorithms.

2) Efficient Channel-Hopping Rendezvous Algorithm Based on Available Channel Set: All the existing rendezvous algorithms that provide guaranteed rendezvous (i.e., rendezvous can be achieved within finite time) generate channel-hopping (CH) sequences based on the whole channel set. However, some channels may be unavailable (e.g., being used by the licensed users) and these existing algorithms would randomly replace the unavailable channels by the available ones in the CH sequence. This random replacement is not effective, especially when the number of unavailable channels is large. We design a new rendezvous algorithm, called Interleaved Sequences based on Available Channel set (ISAC), that attempts rendezvous on the available channels only for faster rendezvous. ISAC constructs an odd subsequence and an even subsequence and interleaves these two subsequences to compose a CH sequence. We prove that ISAC provides guaranteed rendezvous. We derive the upper bounds on the maximum time-to-rendezvous to be $O(|C^1||C^2|)$ ($|C^1| \leq Q, |C^2| \leq Q$), where $|C^1|$ and $|C^2|$ are the numbers of available channels of two users and $Q$ is the total number of channels. Extensive experiments are conducted to evaluate ISAC.

3) Adjustable Rendezvous in Multi-Radio CRNs: We propose an Adjustable Multi-Radio Rendezvous (AMRR) algorithm which exploits multiple radios for fast rendezvous based on available channels only. Suppose that a cognitive user is equipped with $R$ radios. Our basic idea is to partition the radios into two groups: $k$ stay radios and $(R - k)$ hopping radios. The user stays on specific channels in the stay radios while hops on its available channels parallely in the hopping radios. We prove that the maximum time-to-rendezvous (MTTR) of AMRR is upper-bounded by $O\left(\frac{|C^1||C^2|}{R_1R_2}\right)$, where $|C^1|$ and $|C^2|$ are the numbers of available channels of two users and $R_1$ and $R_2$ are the numbers of radios of the two users. This bound meets the
lower bound of MTTR of any deterministic rendezvous algorithm when two users are equipped with the same number of radios (i.e., $R_1 = R_2$). AMRR is adjustable in giving its best performance on either MTTR or $E(TTR)$ by adjusting value of $k$. Simulation results show that AMRR performs better than the state-of-the-art.

4) **Cooperative Rendezvous in Multi-User CRNs**: The existing rendezvous algorithms implicitly assume that only one pair of users send handshaking messages for rendezvous at a time. In practice, more than one pair of users may go through the rendezvous process at about the same time and their handshaking messages may collide with each other. As a result, the rendezvous performance would degrade. We propose to turn this disadvantage (i.e., multiple users cause multiple access interference to each other) into an advantage (i.e., multiple users cooperate with each other for fast rendezvous). Specifically, we propose a *Cooperative Rendezvous Protocol* (CRP) where multi-pair users are cooperative to relay the channel information to speed the rendezvous efficiently. If one user gets the channel information of its intended rendezvous user, it can change the channel set and only hop on the commonly-available channels between itself and its intended rendezvous user. CRP can be applied in conjunction with any existing rendezvous algorithm. We prove that CRP can decrease the expected time-to-rendezvous ($TTR$). The simulation results show that: i) When there are multiple pairs of users, the rendezvous performance is significantly degraded because of collision, and ii) CRP can effectively counteract the collision and significantly improve the rendezvous performance.
Acknowledgements

I would like to express my special appreciation and thanks to my supervisor, Prof. Yiu Wing Leung, for his expertise on inspiring guidance and constructive suggestions in my studies and research works in these years. He has brought me into this challenging research area and shared insightful experiences with me. When I was in dilemma, he always shares constructive idea with me, like lighthouse over the sea. He encourages me to explore terra incognita instead of following others’ step. I would also like to thank my co-supervisors, Dr. Hai Liu, for his great patience in guiding me to shape good research topics, and improve skills of presentation and paper writing. He cares about my progress and gives me prompt feedback positively all the time. I would also like to thank Dr. Xiaowen Chu and Dr. Zhiyong Lin for offering useful information and assistance when I have difficulty in mathematical proof. Without their supervision and constant help, it would not have been possible for me to complete this dissertation.

I would like to thank my friends for their direct and indirect help. In particular, I should mention Ms. Yuanyuan Bian, Ms. Ruijiao Liang, Ms. Qing Bao, Mr. Qian Chen, Ms. Xinxin Mei, Ms. Jingzhu He, Ms. He Chang, Mr. Fusang Zhang among many others. I would like to express my special appreciation and thanks to Qian Chen for his help to solve my programming problems. Finally, I take this special occasion to thank my father Hui Yu, my mother Huaiai Tang and my brother Cheng Yu for supporting me for so many years. I also wish to thank my mentor Mr. Stephen Lai and Mr. Zhanmin Li. They taught me what is the right attitude to handle problems and overcome difficulties. Without them, I would never go so far.
Table of Contents

Declaration i

Abstract ii

Acknowledgements v

Table of Contents vi

List of Tables x

List of Figures xi

List of Symbols xiv

List of Abbreviation xv

Chapter 1 Introduction 1

1.1 Cognitive Radio Networks 1

1.2 Rendezvous in Cognitive Radio Networks 2

1.3 Contributions 4

1.3.1 Multiple Radio for Fast Rendezvous in Cognitive Radio Networks 4

1.3.2 Channel-Hopping Based on Available Channel Set for Rendezvous of Cognitive Radios 8

1.3.3 Adjustable Rendezvous in Multi-Radio Cognitive Radio Networks 9

1.3.4 Cooperative Rendezvous in Multi-User Cognitive Radio Networks 10

1.4 Dissertation Outline 12
## Chapter 2  Literature Review

2.1 Taxonomy of Existing Rendezvous Algorithms .......................... 13  
   2.1.1 Architecture: centralized/decentralized .......................... 13  
   2.1.2 Common control channel (CCC): used/not used .................. 13  
   2.1.3 Applicable models: symmetric/asymmetric ..................... 14  
   2.1.4 Maximum Time-To-Rendezvous (MTTR): finite/infinite ........ 15  
   2.1.5 Based on Available/Whole Channel Set .......................... 16  

2.2 Algorithms for Blind Rendezvous ........................................ 17  
   2.2.1 Algorithms that require time-synchronization .................. 17  
   2.2.2 Algorithms that do not require time-synchronization .......... 18  
   2.2.3 Other additional requirements ................................. 19  

## Chapter 3  Multiple Radios for Fast Rendezvous in Cognitive Radio  

3.1 System Model and Problem Formulation ............................... 21  

3.2 Solutions .............................................................. 23  
   3.2.1 Generalized Random Algorithm ................................. 23  
   3.2.2 Generalized Existing Rendezvous Algorithms ................ 25  
   3.2.3 New Algorithm .................................................. 32  

3.3 Simulation ............................................................. 39  
   3.3.1 Effectiveness of Proposed Approach ............................ 41  
   3.3.2 Property of Proposed Approach: Influence of Radio Allocation 44  
   3.3.3 Property of Proposed Approach: Influence of $G$ for Fixed $\theta$ 50  
   3.3.4 Performance of Generalized Algorithms and Proposed Algorithm 51  

3.4 Chapter Summary ...................................................... 55  

## Chapter 4  Efficient Channel Hopping Rendezvous Algorithm Based 

on Available Channel Set .................................................. 57  

4.1 System Model And Problem Definition ............................... 57  

4.2 Algorithm Design ...................................................... 58
4.3 Algorithm Analysis .................................................. 62
4.4 Simulation ............................................................. 68
  4.4.1 Effectiveness of ISAC ........................................... 68
  4.4.2 Influence of m or n for fixed Q .............................. 73
  4.4.3 Influence of G for fixed m .................................... 73
4.5 Chapter Summary ..................................................... 75

Chapter 5 Adjustable Rendezvous in Multi-Radio Cognitive Radio Networks

5.1 Adjustable Multi-Radio Rendezvous Algorithm .................. 77
5.2 Algorithm Analysis .................................................. 80
5.3 Simulation ............................................................. 87
  5.3.1 Performance of different allocations of stay radios (k) .... 88
  5.3.2 Comparisons of different algorithms ......................... 89
5.4 Chapter Summary ..................................................... 91

Chapter 6 Cooperative Rendezvous in Multi-User Cognitive Radio Networks

6.1 System Model ........................................................ 94
6.2 Protocol Description ............................................... 96
6.3 Protocol Analysis ................................................... 100
  6.3.1 The expectation of G .......................................... 101
  6.3.2 The fitting of the expected latency of P1 and P2 (L1) .... 101
  6.3.3 The analysis of Random Algorithm ......................... 102
  6.3.4 The analysis of a general existing algorithm which is based on available channel set .......... 105
6.4 Simulation ............................................................ 108
  6.4.1 Collision evaluation .......................................... 108
  6.4.2 Performance of Cooperative Rendezvous Protocol ........ 109
6.5 Chapter Summary ................................................... 114
List of Tables

1.1 Comparing the Upper Bounds of MTTR and E(TTR) . . . . . . . . . 5
1.2 Summary of Representative CH Algorithms . . . . . . . . . . . . . 7
## List of Figures

1.1 An illustration example of rendezvous in CRNs. . . . . . . . . . . . . 4

2.1 Taxonomy of the existing rendezvous algorithms. . . . . . . . . . . . . 17

3.1 Sequence structure when user $i$ has 3 radios and user $j$ has 2 radios. . 22

3.2 Rendezvous achieved by Jump-Stay with Independent Sequence. . . . 29

3.3 Six cases under the symmetric model. Remarks: We say $r_i = r_j$

   (Figure 3.3(b) and Figure 3.3(f)) if all step-lengths of CH sequences

   with user $i$ are identical and are equal to those with user $j$, and $r_i \neq r_j$

   (Figure 3.3(a) and Figure 3.3(e)) otherwise. . . . . . . . . . . . . . 30

3.4 Rendezvous of an existing algorithm with Parallel Sequence when

   $R_i = R_j = 2$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

3.5 CH sequence of a user with 3 radios and 7 channels. . . . . . . . . . . 35

3.6 Rendezvous of two users by performing RPS. . . . . . . . . . . . . . . 36

3.7 Four cases of RPS under the symmetric model. . . . . . . . . . . . . 37

3.8 Four cases of RPS under the asymmetric model. . . . . . . . . . . . . 40

3.9 Comparison of multi-radio and single-radio under the symmetric mod-

el (Random). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

3.10 Comparison of multi-radio and single-radio under the symmetric mod-

el (Jump-Stay). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

3.11 Comparison of multi-radio and single-radio under the asymmetric

   model (Random). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3.12 Comparison of multi-radio and single-radio under the asymmetric

   model (Jump-Stay). . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3.13 Comparison of different allocation of $(R_i, R_j)$ when $R_i + R_j = 6$ under

   the symmetric model (Random). . . . . . . . . . . . . . . . . . . . 47

3.14 Comparison of different allocation of $(R_i, R_j)$ when $R_i + R_j = 6$ under

   the symmetric model (Jump-Stay). . . . . . . . . . . . . . . . . . . 47
3.15 Comparison of different allocation of \((R_i, R_j)\) when \(R_i + R_j = 6\) under the symmetric model (RPS).

3.16 Comparison of different allocation of \((R_i, R_j)\) when \(R_i + R_j = 6\) under the asymmetric model (Random).

3.17 Comparison of different allocation of \((R_i, R_j)\) when \(R_i + R_j = 6\) under the asymmetric model (Jump-Stay).

3.18 Comparison of different allocation of \((R_i, R_j)\) when \(R_i + R_j = 6\) under the asymmetric model (RPS).

3.19 Comparison of different \(G\) when \(\theta = 0.5\) (Jump-Stay).

3.20 Comparison of different \(G\) when \(\theta = 0.5\) (RPS).

3.21 Comparison of different algorithms under the symmetric model when \((R_i, R_j) = (3, 3)\).

3.22 Comparison of different algorithms under the symmetric model when \((R_i, R_j) = (3, 4)\).

3.23 Comparison of different algorithms under the asymmetric model when \(\theta = 0.8\).

3.24 Comparison of different algorithms under the asymmetric model when \(\theta = 0.4\).

4.1 Sender-role Sequence.

4.2 Structure of Receiver-role Sequence.

4.3 Receiver-role Sequence when \(C^2 = \{1, 3, 4\}\).

4.4 Rendezvous of two users performing ISAC when \(C^1 = \{1, 2\}\) and \(C^2 = \{1, 3, 4\}\).

4.5 Comparison of different algorithms under the asymmetric model when \(\theta = 0.1\) and \(G = 1\).

4.6 Comparison of different algorithms under the asymmetric model when \(\theta = 0.4\) and \(G = 0.25\theta Q\).

4.7 Comparison of different algorithms under the asymmetric model when \(\theta = 0.8\) and \(G = 0.75\theta Q\).
List of Symbols

$Q$     The number of all channels
$P$     The smallest prime number which is not smaller than $Q$
$r$     Step-length
$C$     The whole channel set
$C^i$   The available channel set of user $i$
$|C^i|$  The number of available channels of user $i$
$G$     The number of commonly-available channels
$\rightarrow S_i^t$ The hopping channels of user $i$ in time slot $t$ when it is equipped with multiple radios
$c_i$   Channel $i$
$m$     An variable
$m_p$   The smallest prime number which is not smaller than $m$
$R_i$   The number of radios of user $i$
$k_i$   The number of stay radios of user $i$
$\theta$ The ratio of the number of available channels to the number of all channels
$K$     The number of users
$L_1$   The latency in stage 1
## List of Abbreviation

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRN</td>
<td>Cognitive Radio Network</td>
</tr>
<tr>
<td>SU</td>
<td>Secondary User</td>
</tr>
<tr>
<td>PU</td>
<td>Primary User</td>
</tr>
<tr>
<td>CH</td>
<td>Channel Hopping</td>
</tr>
<tr>
<td>TTR</td>
<td>Time To Rendezvous</td>
</tr>
<tr>
<td>MTTR</td>
<td>Maximum Time To Rendezvous</td>
</tr>
<tr>
<td>E(TTR)</td>
<td>Expected Time To Rendezvous</td>
</tr>
<tr>
<td>VTTR</td>
<td>Variance of TTR</td>
</tr>
<tr>
<td>CCC</td>
<td>Common control channel</td>
</tr>
<tr>
<td>RPS</td>
<td>Role-based Parallel Sequence</td>
</tr>
<tr>
<td>ISAC</td>
<td>Interleaved Sequences based on Available Channel set</td>
</tr>
<tr>
<td>ES</td>
<td>Even Subsequence</td>
</tr>
<tr>
<td>OS</td>
<td>Odd Subsequence</td>
</tr>
<tr>
<td>AMRR</td>
<td>Adjustable Multi-Radio Rendezvous</td>
</tr>
<tr>
<td>CRP</td>
<td>Cooperative Rendezvous Protocol</td>
</tr>
<tr>
<td>JS</td>
<td>Jump-Stay</td>
</tr>
<tr>
<td>EJS</td>
<td>Enhanced Jump-Stay</td>
</tr>
<tr>
<td>MC</td>
<td>Modular Clock</td>
</tr>
<tr>
<td>MMC</td>
<td>Modified Modular Clock</td>
</tr>
<tr>
<td>DRSEQ</td>
<td>Deterministic Rendezvous Sequence</td>
</tr>
<tr>
<td>CRSEQ</td>
<td>Channel Rendezvous Sequence</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>ACH</td>
<td>Asynchronous Channel Hopping</td>
</tr>
<tr>
<td>QCH</td>
<td>Quorum-based Channel Hopping</td>
</tr>
<tr>
<td>DRDS</td>
<td>Disjoint Relaxed Different Set</td>
</tr>
<tr>
<td>HH</td>
<td>Heterogeneous Hopping</td>
</tr>
<tr>
<td>EAR</td>
<td>Enhanced Adaptive Rendezvous</td>
</tr>
<tr>
<td>GCR</td>
<td>General Construction for Rendezvous</td>
</tr>
<tr>
<td>MTP</td>
<td>Moving Traversing Pointers</td>
</tr>
<tr>
<td>CBH</td>
<td>Conversion Based Hopping</td>
</tr>
<tr>
<td>RW</td>
<td>Ring-Walk</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Cognitive Radio Networks

With the traditional static spectrum management, a significant portion of the licensed spectrum is under-utilized most of time while the unlicensed spectrum is over-crowded due to the growing demand for wireless radio spectrum from exponential growth of various wireless devices [3]. Dynamic Spectrum Access utilizes the wireless spectrum in a more intelligent and flexible way. Cognitive radios are a promising enabler for Dynamic Spectrum Access because they can sense and access the idle channels. Cognitive radio networks (CRNs) have been proposed to allow unlicensed users (or secondary users, SUs) to access the spectrum of licensed users (or primary users, PUs) opportunistically. A CRN consists of a group of SUs who coexist with PUs in the same geographic area. SUs are equipped with cognitive radios, which are capable of identifying and accessing the vacant portions of the licensed spectrum (i.e., idle channels) without causing interference to PUs. To protect the transmission of PUs, SUs are required to vacate the channels once PUs reclaim them. Therefore, available channels of SUs are usually not static and change dynamically due to unpredictable activities of PUs, which distinguishes CRNs from traditional wireless networks significantly.

Channelization of spectrum is usually regulated by the radio communications agencies of different countries or word-wide regions under supervision of International Telecommunication Union. Channelization of spectrum is done and fixed before PUs are licensed to use the spectrum. For example, according to the Second Memorandum Opinion and Order (FCC 10-174), in the US a TV channel has a bandwidth of 6Mhz and TV stations operate from channels 2 to 69 in the Very High Frequency and the Ultra High Frequency spectrum. In a CRN, users (We use user and SU in-
terchangeably to refer to secondary user) are assumed to be aware of channelization of the licensed spectrum in advance. Each user should sense the spectrum to identify available channels before using them. A channel is determined to be available to the user if the user can operate on the channel without causing interference to PUs. In dynamic environment, when a PU becomes active and reclaims its spectrum, SUs should vacate corresponding channels and these channels consequently become unavailable. Thus, duration of an available channel is the duration up to the time when a PU reclaims this channel.

1.2 Rendezvous in Cognitive Radio Networks

In cognitive radio networks (CRNs), multiple idle channels may be available to SUs. To establish a direct communication link, two users should operate on a commonly-available channel at the same time. The process of two or more users to meet (on a commonly-available channel at the same time) and establish a link on this channel is referred to as rendezvous. Rendezvous is a fundamental and essential operation in CRNs because data communication is impossible without rendezvous of users. However, implementation of rendezvous is nontrivial in CRNs due to the following reasons: i) Two users are even not aware of the presence of each other before rendezvous; ii) Available channels of users might be different and change dynamically; iii) Users may join the network and start the rendezvous operation at different instants of time. We define the starting time of a rendezvous process as the starting time of the later user.

A number of rendezvous algorithms have been proposed in the literature [28]. There are basically two approaches to implement rendezvous: using common control channel (CCC) and using channel-hopping (CH) technique. With the former approach, rendezvous is achieved via either global CCC shared by all users, or local CCC shared by the users in the same group. Both global CCC and local CCC suffer from the problems of low scalability and high maintenance cost, and CCC easily becomes a single point of failure to jamming attack. Channel-hopping (CH)
is another representative technique for rendezvous. With CH technique, each SU selects a set of available channels and hops among these channels. A rendezvous is said to be achieved if two SUs hop on the same channel simultaneously. Using CH technology can avoid the aforementioned problems and thus is widely adopted in almost all existing algorithms.

With the CH technique, the CRN is assumed to be time-slotted. Communications of users are assumed to follow a specific standard (e.g., IEEE 802.22) which specifies the creation and management of time slots. For example, IEEE 802.22 suggests that the length of a time slot for users is 10ms [25]. In each time slot, a user selects one of its available channels and hops on it. Each user independently performs a pre-designed CH algorithm and hops on channels according to the CH sequence produced by the CH algorithm. It is not necessary for users to follow the same CH sequence. Even performing the same CH algorithm, different users may obtain different CH sequences and may begin their channel-hopping at different time. A rendezvous is said to be achieved if the users hop on the same commonly-available channel in the same time slot, during which the users can discover each other and establish the communication link. It is worth pointing out that rendezvous of two users is most crucial and rendezvous of multiple users in the whole network can be implemented by pairwise rendezvous of two neighboring users [25].

Taking example in Figure 1.1, SUs opportunistically access three licensed channels with indices of 1, 2, and 3. Suppose that users A and B have the same set of available channels \{1, 2, 3\} and apply the CH algorithm in [38]. User A generates a CH sequence \{1, 2, 3, 3, ...\}. That is, user A attempts rendezvous on channel 1 in its first time slot, hops on channel 2 in its second time slot, and so on. User B starts its channel-hopping two time slots later than A and generates a sequence \{1, 2, 3, 3, ...\}. As shown in the top right corner of Figure 1.1 using the CH algorithm in [38], A and B can achieve rendezvous on channel 3 in B’s third time slot. Notice that not all CH algorithms can guarantee rendezvous of users. If each user applies the simplest random CH algorithm, i.e., randomly hopping on an available channel
in each time slot, the rendezvous of users cannot be guaranteed, as demonstrated in the bottom right corner of Figure 1.1.

In the literature, there are two models to describe the channel availability: i) symmetric model in which all users have the same available channels; and ii) asymmetric model in which different users may have different available channels. Both the symmetric and asymmetric models are important in practice. For example, the symmetric model is suitable for SUs who are located in a relatively small area (compared with their distance to PUs) while the asymmetric model is applicable if geographical locations of SUs are far. Rendezvous is carried out to establish a communication link. It is desirable that this process can be completed as soon as possible so that users can start data transmission. TTR is an ideal performance metric to measure how fast the rendezvous process is.

1.3 Contributions

1.3.1 Multiple Radio for Fast Rendezvous in Cognitive Radio Networks

Many effective rendezvous algorithms have been proposed in the literature. To the best of our knowledge, all the existing rendezvous algorithms implicitly assume that each user is equipped with one radio (i.e., one wireless transceiver). As the cost of
Table 1.1: Comparing the Upper Bounds of MTTR and E(TTR)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Symmetric Model</th>
<th>Asymmetric Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Bounds of MTTR of Existing Algoritms generalized to multiple radios</td>
<td>Random $3P$</td>
<td>$3MP(P-G) + 3P$</td>
</tr>
<tr>
<td>Upper Bounds of E(TTR) of Existing Algorithms generalized to multiple radios</td>
<td>RPS $2 \times \left\lceil \frac{P}{\max(R_i,R_j)} - 1 \right\rceil$</td>
<td>$\left\lceil \frac{P}{\min(R_i,R_j)} - 1 \right\rceil \times (Q - G)$ when $R_i = R_j$</td>
</tr>
<tr>
<td>Upper Bounds of E(TTR) of New Algorithm</td>
<td>Random $\left\lceil \frac{P}{Q} - \frac{A_{Q,R_i}^{R_j}}{A_{Q}^{R_i} - A_{Q}^{R_j}} \right\rceil$</td>
<td>$\left\lceil \frac{P}{Q} \right\rceil \times (Q - G + 1)$ when $R_i = R_j$</td>
</tr>
<tr>
<td>JS/Independent</td>
<td>JS/Parallel $\left\lceil \frac{3P}{R_i} \right\rceil$ when $R_i = R_j$</td>
<td>$\left\lceil \frac{3P}{R_i} \right\rceil$ when $R_i = R_j$</td>
</tr>
<tr>
<td>JS/Parallel</td>
<td>$\left\lceil \frac{5P-4}{3} \right\rceil \times (2P + 1 + \frac{1}{P})$</td>
<td>$\left\lceil \frac{2QP(P-G)+(Q+5-P-(2G-1)/Q)}{R_i} \right\rceil$ when $R_i = R_j$</td>
</tr>
</tbody>
</table>

Remarks: i) $R_i$, $R_j$ are the numbers of radios of two users; $Q$ is the number of all channels; $P$ is the smallest prime number which is not smaller than $Q$; $G$ is the number of commonly-available channels of two users; $|C^1|$, $|C^2|$ are the numbers of available channels of two users, respectively; $A_j^i$ is the number of possible permutations of $i$ objects from a set of $j$. ii) We select the Jump-Stay (JS) algorithm [25] for comparison since it was recently proposed and was shown to have a very good performance [25]. The upper bounds of MTTR and E(TTR) of the existing algorithms are derived in Section 3.2. iii) In a CRN, different SUs may be equipped with different numbers of radios.
wireless transceivers is dropping, this feature can be exploited to significantly improve the rendezvous performance at low cost. In particular, when a SU is equipped with multiple radios, the time-to-rendezvous (TTR, i.e., the time required by the rendezvous operation) can potentially be reduced by a large amount while the additional cost (i.e., cost of the extra radios) is low. In addition, the energy consumption can also be reduced (if the number of radios is increased from 1 to \( n \), \( n \) radios would consume energy but the time spent on rendezvous could be reduced by more than \( n \) times, so the total energy consumption can be reduced). When rendezvous has been done, these radios can be utilized for parallel and fast transmission.

In this part, we study the rendezvous problem in CRNs where each SU is equipped with multiple radios and different SUs may have different numbers of radios. Table 1.1 summarizes the differences between: i) the proposed algorithm, and ii) the existing rendezvous algorithms (e.g., Jump-Stay \[25\]) after they are generalized to use multiple radios.

We make three contributions.

1. We investigate a new approach (i.e., exploiting multiple radios per user) to significantly improving the rendezvous performance at low cost.

2. We generalize the Random algorithm and the existing rendezvous algorithms in order to use multiple radios for faster rendezvous.

3. We propose a new rendezvous algorithm, called \textit{role-based parallel sequence} (RPS), which specifically exploits multiple radios for more efficient rendezvous. We derive upper bounds on the maximum TTR (MTTR) and the expected TTR (E(TTR)) of this algorithm. We conduct extensive simulation to demonstrate that its MTTR and E(TTR) decrease significantly with the increase of the number of radios.
Table 1.2: Summary of Representative CH Algorithms

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Channel Set based on (Available or Whole)</th>
<th>Guaranteed rendezvous</th>
<th>Upper bounds on MTTR under the symmetric model</th>
<th>Upper bounds on MTTR under the asymmetric model</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISAC</td>
<td>Available Channel Set</td>
<td>YES</td>
<td>$2(n m_p - n + 1)$</td>
<td>$2(n m_p - G + 1)$</td>
</tr>
<tr>
<td>MC/MMC [35]</td>
<td>Available Channel Set</td>
<td>NO</td>
<td>Infinity</td>
<td>Infinity</td>
</tr>
<tr>
<td>Random [35]</td>
<td>Available Channel Set</td>
<td>NO</td>
<td>Infinity</td>
<td>Infinity</td>
</tr>
<tr>
<td>Jump-Stay [25]</td>
<td>Whole Channel Set</td>
<td>YES</td>
<td>$3P$</td>
<td>$6Q P( P - G)$</td>
</tr>
<tr>
<td>Enhanced Jump-Stay  [26]</td>
<td>Whole Channel Set</td>
<td>YES</td>
<td>$4P$</td>
<td>$4P( P + 1 - G)$</td>
</tr>
<tr>
<td>CRSEQ [33]</td>
<td>Whole Channel Set</td>
<td>YES</td>
<td>$\geq 3P^2 - 4P + 1$ and $\leq P(3P - 1)$</td>
<td>$P(3P - 1)$</td>
</tr>
<tr>
<td>ARMMC [13]</td>
<td>Whole Channel Set</td>
<td>NO</td>
<td>Infinity</td>
<td>Infinity</td>
</tr>
<tr>
<td>MTP [16]</td>
<td>Available Channel Set</td>
<td>YES</td>
<td>$2m^2 \times 32(\log \log Q + 1)$</td>
<td>$2(\max{m, n})^2 \times 32(\log \log Q + 1)$</td>
</tr>
<tr>
<td>CBH [14]</td>
<td>Available Channel Set</td>
<td>YES</td>
<td>$O(m^2)$</td>
<td>$O((\max{m, n})^2)$</td>
</tr>
</tbody>
</table>

Remarks: $m$, $n$ are the numbers of available channels of two users, respectively; $m_p$ is the smallest prime number which is not smaller than $m$; $Q$ is the total number of channels (i.e. all potentially available channels); $P$ is the smallest prime number which is not smaller than $Q$; $G$ is the number of commonly-available channels of the two users; ISAC [40] is proposed in 2013 while CBH [14] and MTP [16] are proposed in 2014 and 2015.
1.3.2 Channel-Hopping Based on Available Channel Set for Rendezvous of Cognitive Radios

For the existing algorithms that provide guaranteed rendezvous \cite{25, 33, 6, 26, 24}, they generate the CH sequences based on the whole channel set (i.e., set of all channels). In practice, some channels may not be available because: i) these channels are being used by the licensed users, or ii) the channels are not being used but the cognitive users cannot identify these idle channels because of the practical limitations of the spectrum sensing methods \cite{36}. In fact, the available channel set is usually a small subset of the whole channel set. For example, work in \cite{36} optimizes the detection (sensing) time for channel efficiency. It shows that when the SNR (signal-to-noise ratio) is $-3$dB, the SUs can only detect one idle channel in 15% of the total idle period. In other words, on average, at most 15% of channels can be correctly detected as “available” for data transmission of SUs. If a cognitive user follows a CH sequence which includes both the available and the unavailable channels, this user would waste time in attempting rendezvous on the unavailable channels. This would increase the time to complete rendezvous, especially when the available channel set is a small subset of the whole channel set.

In this part, we propose a new rendezvous algorithm, called *interleaved sequences based on available channel set* (ISAC), for CRNs. Our contributions are three-fold.

1. *Algorithm design:* We design ISAC to realize two desirable properties: i) CH sequences are generated based on the available channel set for better rendezvous performance, and ii) rendezvous can certainly be completed within finite time.

2. *Algorithm analysis:* We prove that ISAC provides guaranteed rendezvous. In particular, we derive upper bounds on the maximum time-to-rendezvous (MT-TR) for two popular models of channel availability (namely, symmetric model and asymmetric model). These upper bounds are expressed in terms of the number of available channels instead of the total number of channels. Table 1.2
shows the upper bounds on MTTR of the existing algorithms, especially when
the available channel set is a small subset of the whole channel set.

3. Algorithm evaluation: We conduct extensive simulation for performance e-
valuation and demonstrate that ISAC gives significantly smaller MTTR than
the existing algorithms. ISAC is suitable to many QoS-concern applications
of CRNs where communications are required to be realized within a specific
duration (i.e., small MTTR is desired).

1.3.3 Adjustable Rendezvous in Multi-Radio Cognitive Ra-
dio Networks

In this part, we study the rendezvous problem in CRNs where each user is equipped
with multiple radios and different users may have different numbers of radios. We
propose an Adjustable Multi-Radio Rendezvous (AMRR) algorithm which partitions
$R$ radios of a user into two groups: $k$ stay radios and $(R - k)$ hopping radios
($1 \leq k < R$). The user stays on specific channels in the stay radios while hops on its
available channels parallely in the hopping radios. The contributions of this work
are summarized below.

1. Fast Rendezvous: We propose a new rendezvous algorithm, called Adjustable
Multi-Radio Rendezvous (AMRR). AMRR exploits multiple radios and gener-
ates CH sequences based on available channels for fast rendezvous. We conduct
extensive simulation to demonstrate that AMRR can significantly shorten MT-
TR and $E(TTR)$. For example, AMRR reduces the MTTR up to 85% when
the average number of radios is 4 and the ratio of the available channels is
10%, compared to the state-of-the-art [39, 23, 30].

2. Adjustable Rendezvous: Among all existing multi-radio rendezvous algorithms,
AMRR is the only adjustable one which can give its best performance on
either MTTR or $E(TTR)$ with different allocations of radios. Specifically, if an
application concerns MTTR other than $E(TTR)$, AMRR can give its shortest
MTTR by allocating one radio as the stay radio (i.e., the remaining radios are the hopping radios) under the symmetric model, and allocating half of the radios as the stay radios under the asymmetric model. If E(TTR) becomes more important than MTTR, AMRR gives its shortest E(TTR) by allocating one radio as the stay radio under both the symmetric model (theoretically) and the asymmetric model (experimentally).

3. **Optimal Upper-bound of MTTR**: We derive upper-bounds of MTTR and E(TTR) of AMRR. We prove that AMRR gives the lowest (i.e., optimal) MTTR in the order of magnitude when two users are equipped with the same number of radios. In other words, any deterministic rendezvous algorithm using the same number of radios cannot give MTTR smaller than that of AMRR under the asymmetric model.

1.3.4 **Cooperative Rendezvous in Multi-User Cognitive Radio Networks**

In the literature, most of existing algorithms consider the rendezvous of one-pair users [25, 6, 33, 24]. If there are multiple SU pairs in the system, even one-pair users hop on the same channel at the same time slots, the rendezvous could fail when there is another user also hops on this channel at this time slot. We call it collision. In the wireless networks, there are a large number of users. Each user has an intended rendezvous user. A good rendezvous algorithm should avoid high collision. We observed that there are two metrics to evaluate the collision: Sequence Similarity (SS) and Congestion. [1, 2] define an expected hamming distance as 

$$E \left[ \sum_{i=1}^{n} \frac{1}{n} (x_i \neq y_i) \right]$$

That is, HD is a record of difference between multiple sequences. They describe that CH sequences with higher HD result in a lower collision probability. In [37], authors defines it as degree of congestion. It defines the load of a channel $c$ as the ratio of the maximum number of users that hop on $c$ at the same time, to the number of users $K$. They use the maximum channel load of all channels to measure the degree of congestion. We do simulation to estimate the collision caused by multi-pair users.
When we consider collision, the rendezvous is achieved only when one-pair users and there are only these two users hop on one same channel in the same time slot. The results are shown in Section 6.4. When the number of users increases, the collision becomes more serious. For example in Figure 6.8, when there are 100 channels and 100 users in the system, each user has average 50 available channels, the average TTR of Random Algorithm 101.01 when we ignore the collision. However, when we consider the collision, the average TTR becomes 231.24. It is more than double of the original one. Therefore, the collision should not be ignored.

It is hard to solve the collision problem by designing an rendezvous algorithm with high HD and low congestion degree. Moreover, we think there is a trade-off. A higher HD implies a larger TTR. If one algorithm try to let multiple users hop on different channels, the pair users intended to rendezvous would also try to hop on different channels. The TTR will be larger. Instead of solving the collision caused by multi-pair users, we try to make use of multi-pair users. All users are cooperative to transmit the channel information of each other even when one user achieve rendezvous with an user which is not the intended rendezvous user. For example, user 1 and user 2 are one-pair users intended to rendezvous. User 3 and user 4 are another pair. If user 1 and user 3 achieve rendezvous, user 1 can get the channel information of user 2 from user 3 if user 3 ever achieved rendezvous with user 2. User 1 can narrow the channel area to be the commonly-available channels after it get the channel information of user 2. The rendezvous will be much faster. This cooperative policy is simple and effective.

In this part, we study the rendezvous of one-pair users under multi-pair users system. The contributions of this work are summarized below.

1. **Collision Evaluation:** We evaluate the collision caused when there are multi-pair users in the CRNs. When the number of users increases, the collision becomes more serious. The collision can enlarge the expected TTR up to 213.13% when we implement pure Random Algorithm and there are 100 channels and 200 users.
2. *Cooperative Rendezvous Protocol*: We propose a new protocol, called *Cooperative Rendezvous Protocol* (CRP). CRP makes use of the cooperativity of all users to relay the channel information. Each user may get the channel information of its intended rendezvous user from another user. All users are cooperative and try to narrow the channel set to speed up the rendezvous process. There are three cases for rendezvous of one-pair users.

3. *Theoretical Analysis*: We theoretically analyze that the improvement of the expected TTR when we apply CRP in conjunction with Random Algorithm and a general existing rendezvous algorithm which generates CH sequence based on available channels. There is improvement when we apply CRP in conjunction with an existing algorithm when the ratio of the number of available channel to the number of all channels is smaller to a specific value.

4. *Protocol evaluation*: We conduct extensive simulation for performance evaluation. We demonstrate that serious collision could be caused when there are multi-pair users in the system. We also show that CRP can counteract the collision and decrease the expected TTR in terms of any ratio of the number of available channel to the number of all channels.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows. Chapter 2 introduces the literature review. Chapter 3 presents generalized Random algorithm and the existing rendezvous algorithms to use multiple radios and a new rendezvous algorithm which specifically exploits multiple radios for more efficient rendezvous. Chapter 4 presents a rendezvous algorithm, called *interleaved sequences based on available channel set* (ISAC), for CRNs, Chapter 5 presents Adjustable Multi-Radio Rendezvous (AM-RR) algorithm and Chapter 6 presents a *Cooperative Rendezvous Protocol* (CRP) to makes use of all users to transmit the channel information. Finally, Chapter 7 concludes the dissertation.
Chapter 2

Literature Review

2.1 Taxonomy of Existing Rendezvous Algorithms

The existing rendezvous algorithms (systems) can be classified according to different criteria. We discuss the criteria in the following.

2.1.1 Architecture: centralized/decentralized

Under centralized system, such as DSAP [8] and DIMSUMNet [9], a centralized server is operated to schedule the data exchanges among users. With a centralized server for global coordination, this approach eases the rendezvous process. However, centralized systems may suffer from several issues such as low scalability and low robustness (With increased network size, the server easily becomes the bottleneck of communications). Thus, decentralized systems are usually more scalable than centralized ones. However, without the centralized controller, the design and implementation of rendezvous algorithms becomes very challenging. In decentralized system, users have to attempt rendezvous without the aid of any server. A channel is preselected as a CCC. In [19] and [29], a global CCC is preselected and known to all users. In [43] and [22], a cluster-based control-channel method was proposed, in which a local CCC is selected for each cluster. However, the extra costs in establishing and maintaining the global/local CCCs are considerable.

2.1.2 Common control channel (CCC): used/not used

CCC usually refers to a dedicated channel for users to mutually exchange control information. Such a channel may be commonly available to all users (global CCC)
or a group of users (local CCC) in the network.

Many systems assume the existence of CCC to facilitate rendezvous between users. Technically, the desired CCC can be simply allocated in unlicensed spectrum or dynamically established in licensed spectrum. With CCC, users can easily achieve rendezvous, as they just need to hop on the CCC and exchange control information over the channel. However, using CCC may cause new problems including: i) Allocating CCC in unlicensed spectrum will definitely aggravate the congestions; ii) Constructing and maintaining CCC in licensed spectrum is hard or even impractical since channel availability at users usually changes over time and different users may have different available channels at any time; iii) CCC is vulnerable to jamming attack and easily becomes a single point of failure.

To avoid the aforementioned problems, most current works focus on the rendezvous systems without using CCC. In particular, intensive efforts have been devoted to blind rendezvous systems in which neither CCC nor centralized controller is required [35]. Compared with those systems using CCC, blind rendezvous systems are more feasible and reliable, but also may be more complicated and less efficient. We notice that almost all blind rendezvous systems are CH-technique based. With this desirable feature, this approach has drawn significant attention in the literature and some effective algorithms for blind rendezvous have been proposed (e.g., Jump-Stay [25], M-/L-QCH [7] and ASYNC-ETCH [41]).

### 2.1.3 Applicable models: symmetric/asymmetric

In CRNs, users periodically sense spectrum of PUs to identify available channels. Due to the imperfect sensing operation and the difference in geographic locations, the users usually obtain different sensing results and thus identify different sets of available channels. It is called *symmetric model* if all users involved in rendezvous are assumed to have the same set of available channels. It is called *asymmetric model* if users may have different available channels [27]. Taking example in Figure 1.1 again, users A and B have the identical set of available channels since they are
both located outside the interference area of any PU. Users $A$ and $B$ can attempt rendezvous under the symmetric model. In contrast, users $D$ and $E$ have different sets of available channels and the asymmetric model is applicable. In reality, most cases should fall into the asymmetric model. It is usually more difficult to achieve rendezvous under the asymmetric model than under the symmetric model because the users have fewer commonly-available channels under the asymmetric model (only one in the extreme case). Therefore, some works \cite{38, 42} studied rendezvous under the symmetric model, which is a simplification of the asymmetric model. Many notable algorithms that are applicable to the asymmetric model (and thus applicable to the symmetric model as well) have been proposed \cite{35, 26, 33, 27, 13, 5, 11, 31}.

2.1.4 Maximum Time-To-Rendezvous (MTTR): finite/infinite

To evaluate the performance of rendezvous algorithms, particularly CH rendezvous algorithms, the most important metric is time-to-rendezvous (TTR), which is defined as the number of time slots that it takes for users to achieve rendezvous once all the users have begun their channel-hopping. Generally speaking, TTR of an algorithm is a random variable and it cannot be directly used to evaluate the algorithm. Instead, two TTR-related metrics are usually adopted in practice to serve this aim: MTTR and $E(TTR)$. Specifically, the MTTR of an algorithm is the TTR in the worst scenario when applying this algorithm, while $E(TTR)$ is the expected value of TTR when taking TTR as a random variable. It should be point out that, in practice, the TTR is affected by not only rendezvous algorithms but also many other factors, including delay for spectrum sensing, delay for handshaking, and delay caused by communication failure. Because these factors are independent of rendezvous algorithms, definition of MTTR ($E(TTR)$) actually concerns only the maximum (average) time it takes for users to achieve rendezvous once all users have begun their channel-hopping in ideal communication scenario. An algorithm with finite MTTR means that the algorithm is with guaranteed rendezvous, i.e., the time it takes for users to achieve rendezvous is upper-bounded if the users perform the al-
algorithm. Design of CH algorithms with guaranteed rendezvous is difficult, since the users may not be aware of available channels of each other and time-synchronization is not ensured. Recently, several algorithms with guaranteed rendezvous have been proposed to address the blind rendezvous problem under only the symmetric model [38, 42] or under both the symmetric and asymmetric models [33, 51].

2.1.5 Based on Available/Whole Channel Set

In most of the existing works, CH schemes are always based on the whole channel set. However, some channels may be unavailable due to the PU activities or sensing limitations. In practice, the available channel set is usually a small subset of the whole channel set. To improve the overall performance of rendezvous, some previous works based on the whole channel set would randomly replace the unavailable channels in the CH sequence [25, 26]. This random replacement is not effective, especially when the number of unavailable channels is large. In view of this observation, some algorithms based on available channel set are proposed recently. For example, CSAC [40], MC/MMC [35], CBH [14], MTP [16] and HH [37]. All of them derived an upper-bound on MTTR which is an expression of the number of available channels instead of all channels. MTP is the latest rendezvous algorithm which is based on available channel set [16]. It can guarantee rendezvous within $O(\max\{|V_a|, |V_b|\}^2 \log \log Q)$ where $|V_a|, |V_b|$ are the number of available channels of two users. It designs a rendezvous scheme for two channels first. Each user selects two channels in one period. Then try to move the pointer period by period. When there is one commonly-available channel between the selected two channels in one period, the rendezvous is achieved. Table 1.2 summarizes the characteristics of existing representative CH algorithms.

Based on the above discussion, we present a possible taxonomy of the existing rendezvous algorithms in Figure 2.1. We focuses on blind rendezvous systems which draw most of attentions of the current research. We detail the blind rendezvous of our taxonomy in next section.
2.2 Algorithms for Blind Rendezvous

Blind rendezvous systems have drawn more attention of researchers due to their attractive characteristics such as feasibility and reliability. A typical approach for blind rendezvous is CH technique. We next survey CH based algorithms for blind rendezvous according to the taxonomy in Figure 2.1.

2.2.1 Algorithms that require time-synchronization

Bahl et al. presented a pioneer work in [4] and proposed a link-layer protocol, in which each user is allowed to select multiple (channel, seed) pairs and the CH sequence is determined based on these pairs. Though this protocol was designed for increasing the capacity of IEEE 802.11 networks, it guarantees rendezvous between users under the symmetric model. In [11], Zhang et al. proposed a protocol named SYNC-ETCH (synchronous efficient channel hopping). Given $M$ potentially available channels, via some modulo operations SYNC-ETCH constructs $2M$ different CH sequences with the same length of $2M - 1$ such that any two CH sequences can overlap exactly once.
2.2.2 Algorithms that do not require time-synchronization

There are CH algorithms which do not require time-synchronization. According to existing works [25, 38], let $T$ be the duration for exchanging the necessary control messages in rendezvous. Since the cognitive users may not be time synchronized, each time slot should have a duration of $2T$. The overlap of two time slots of two users is sufficient to send the message.

In [24], they presented a ring-walk (RW) algorithm which guarantees the rendezvous under both models. In RW, each channel is represented as a vertex in a ring. Users walk on the ring by visiting the vertices (channels) with different velocities and rendezvous is guaranteed since users with lower velocities will be caught by users with higher velocities. However, RW requires that each user has a unique ID and knows the upper bound of network size. Recently, a notable work by Theis et al. presented two CH algorithms: modular clock algorithm (MC) and its modified version MMC for the symmetric model and the asymmetric model, respectively [35]. The basic idea of MC and MMC is that each user picks a proper prime number and randomly selects a rate less than the prime number. Based on the two parameters, the user generates its CH sequence via pre-defined modulo operations. Although MC and MMC are shown to be effective, both algorithms cannot guarantee the rendezvous if the selected rates or the prime numbers of two users are identical. Yang et al. proposed two significant algorithms, namely deterministic rendezvous sequence (DRSEQ) [38] and channel rendezvous sequence (CRSEQ) [33], which provide guaranteed rendezvous for the symmetric model and the asymmetric model, respectively. In CRSEQ, the sequence is constructed based on triangle numbers and modular operations. In terms of MTTR, CRSEQ is quite good under the asymmetric model but it does not perform well under the symmetric model. In [25], authors proposed a Jump-Stay (JS) algorithm in which a user alternately “jumps” on available channels and “stays” on a specific channel. They derived the lower bounds of MTTR which are $3P$ and $3QP(P - G) + 3P$ under the symmetric model and the asymmetric model, respectively. Bian et al. [6] presented an asynchronous channel hopping (ACH)
algorithm which aims to maximize rendezvous diversity. It assumes that each user has a unique ID and ACH sequences are designed based on the user ID. Though the length of user ID is a constant, it may result in a long TTR in practice given that a typical MAC address contains 48 bits. In [15], an efficient rendezvous algorithm based Disjoint Relaxed Difference Set (DRDS) was proposed while DRDS only costs a linear time to construct. They proposed a distributed asynchronous algorithm that can achieve and guarantee fast rendezvous under both the symmetric and the asymmetric models. They derived the lower bounds of MTTR which are $3P$ and $3P^2 + 2P$ under the symmetric model and the asymmetric model, respectively. They showed that it is nearly optimal. In [1], the authors addressed the pairwise as well as the multicast rendezvous problems under fast primary user (PU) dynamics. They considered the issue of adaptively adjusting the channel hopping (CH) sequences to account for the spectrum heterogeneity and PU dynamics, which are the two main challenges in DSA systems. Wu et al. proposed an effective rendezvous algorithm called Heterogeneous Hopping (HH) [37] which is based on the available channel set only. There are other algorithms in this category such as synchronous QCH [6], SYNCETCH and ASYNC-ETCH [11], AMRCC [13], C-MAC [12], and MtQS-DSrdv [32]. Most of the existing rendezvous algorithms use one single radio per user for rendezvous. The authors in [30] studied the multi-interface rendezvous problems for CRNs. However, they assumed that all nodes have the same number of radios (say, $R$), the channels are divided into $R$ partitions, and each radio performs rendezvous over the channels in one partition.

2.2.3 Other additional requirements

There are also some algorithms need additional information which is reasonable. ACH [6], [21] and RW [24] need user ID and network size. CSAC [40] need the assumption of two roles of users: sender and receiver. Asymmetric approaches in [10] (i.e., RCCH and ARCH) require role preassignment (i.e., every user is preassigned as either sender or receiver). Only SARCH [10] is for blind rendezvous. The MTTR
is $8N^2 + 8N$. However, the additional requirement is that $2N + 1$ should be prime.
Chapter 3

Multiple Radios for Fast Rendezvous in Cognitive Radio Networks

The existing works on rendezvous implicitly assume that each cognitive user is equipped with one radio (i.e., one wireless transceiver). As the cost of wireless transceiver is dropping, this feature can be exploited to significantly improve the rendezvous performance at low cost. In this chapter, we investigate the rendezvous problem in CRNs where cognitive users are equipped with multiple radios and different users may have different numbers of radios. Specifically, system model and problem formulation are presented in Section 3.1. In Section 3.2, we generalize the Random algorithm and the existing rendezvous algorithms to use multiple radios for faster rendezvous. Then we propose a new rendezvous algorithm which specifically exploits multiple radios for more efficient rendezvous. We present simulation results in Section 3.3 for performance evaluation and conclude this work in Section 3.4.

3.1 System Model and Problem Formulation

We consider a CRN consisting of $K$ ($K \geq 2$) users. Time is divided into slots of equal duration. The licensed spectrum is divided into $Q$ non-overlapping channels $c_1, c_2, ..., c_Q$, where $c_i$ is called channel $i$. Let $C$ be the channel set \{ $c_1, c_2, ..., c_Q$ \}. Let $C^i \subset C$ be the set of available channels of user $i$ ($i = 1, 2, ..., K$), where a channel is said to be available to a user if the user can communicate on this channel without causing interference to any PUs. The available channels can be identified by any spectrum sensing method (e.g., [9]). Without loss of generality, we consider the rendezvous of a pair of users, say user $i$ and user $j$, $i \neq j$ and $i, j = 1, 2, ..., K$.

User $i$ is equipped with $R_i$ ($R_i \geq 1$) radios and user $j$ is equipped with $R_j$ ($R_j \geq 1$) radios. Note that $m$ may not be equal to $n$ in CRNs. Let $G$ be the number of
Figure 3.1: Sequence structure when user $i$ has 3 radios and user $j$ has 2 radios.

commonly-available channels of user $i$ and user $j$. The CH sequence of user $i$ is denoted by $\{S^i_{11}, S^i_{21}, S^i_{31}, \ldots\}$, where vector $\vec{S}^i_t = \{S^i_{1t}, S^i_{2t}, S^i_{3t}, \ldots, S^i_{R_i t}\}$ represents that user $i$ hops on channels $S^i_{th}$ on radio $h$ in time slot $t$. Figure 3.1 shows the sequence structure where user $i$ has 3 radios and user $j$ has 2 radios.

In this study, we do not require time-synchronization in the networks. Without time synchronization, the slot time could be doubled so as to ensure that the overlap of two time slots is long enough to complete all necessary steps of rendezvous [33, 33]. In this sense, the CH sequences of two users are slot-aligned even without time-synchronization [28]. In each time slot, user $i$ hops on $R_i$ channels and user $j$ hops on $n$ channels to attempt rendezvous. We say that a rendezvous is achieved if user $i$ and user $j$ hop on the same channel on any of the radios in the same time slot. Typically the time-to-rendezvous (TTR) is usually in the order of tens of milliseconds, which is very small compared with the PU dynamic. To illustrate, let us consider an example. Suppose it is necessary to send a handshaking packet of 100 bytes (e.g., containing information of user IDs) for rendezvous and the data rate of the wireless channel is 10 Mbps. That is, The duration of each frame is $(100 \times 8)/(10 \times 10^6) = 0.08$ms. Suppose that the SIFS (Short Inter Frame Spacing) is $10\mu$s (e.g., in IEEE 802.11b or IEEE 802.11n [18]) and the three-way handshaking is adopted. We can estimate the time necessary for rendezvous as $3 \times (0.08 \times 10^{-3} + 10 \times 10^{-6}) = 0.27$ms. Thus, the duration of each time slot is $0.54$ms around (i.e., the overlap of two time slot is no less than $0.27$ms). If it takes 10 to 100 time slots to achieve rendezvous (see the numerical results in our paper), the time-to-rendezvous is only $5.4$ms to $54$ms. On
the other hand, a common type of primary users quoted in the literature is the TV station which only uses its TV channels at certain time of each day (say, evenings and nights), and the activity of this PU changes very slowly compared with the time-to-rendezvous. Therefore, channels availabilities are assumed to be static in the process of rendezvous. We define the rendezvous problem as follows.

**Rendezvous problem for two users:** Suppose two users have $R_i$ and $R_j \ (R_i, R_j \geq 1)$ radios respectively and they may start the rendezvous process at different time. The problem is to determine a CH sequence for each radio of each user, such that these users will hop on a commonly-available channel in the same time slot.

### 3.2 Solutions

In this section, we generalize the Random algorithm and the existing algorithms to use multiple radios for faster rendezvous. Then we design a new rendezvous algorithm, called *role-based parallel sequence* (RPS), which specifically exploits multiple radios for more efficient rendezvous.

#### 3.2.1 Generalized Random Algorithm

When there is a single radio, the Random algorithm randomly selects an available channel in each time slot and attempts to achieve rendezvous on this channel in this time slot. When there are multiple radios, this Random algorithm can be generalized as follows: each radio randomly and independently selects an available channel in each time slot and attempts to achieve rendezvous on this channel in this time slot. When two or more radios happen to select the same channel, these radios will randomly select again until they select different channels (suppose that for each user the number of available channels is no less than that of radios). Obviously, the Random algorithm cannot guarantee the rendezvous within finite time and hence the MTTR is infinity. The algorithm is formally presented as follows.

In line 2, $\overrightarrow{S_i} = \{S_{i1}^i, S_{i2}^i, S_{i3}^i, \ldots, S_{iR_i}^i\}$ denotes the set of channels that user $i$ hops
Algorithm 1: Random Algorithm

Require: \( Q, R_i, C^i \) // for user \( i \)

1: \( t = 1 \)
2: \( \vec{S}_t^i = \{S_{t1}^i, S_{t2}^i, S_{t3}^i, \ldots, S_{tR_i}^i\} \)
3: while not rendezvous do
4: for \( k = 1 \) to \( R_i \) do
5: \( S_{tk}^i = \text{RandomSelect}(C^i) \)
6: for \( j = 1 \) to \( (k - 1) \) do
7: if \( S_{tk}^i == S_{tj}^i \) then
8: \( S_{tk}^i = \text{RandomSelect}(C^i) \)
9: \( j = 1 \)
10: end if
11: end for
12: end for
13: \( t = t + 1 \)
14: Attempt rendezvous on \( \vec{S}_t^i \)
15: end while
on \( m \) respective radios in time slot \( t \). In lines 4-12, user \( i \) randomly selects an available channel for each radio. In line 14, the rendezvous is achieved when one channel in \( S^k_i \) is equal to one channel of another user. The following theorems give the performance properties of the generalized Random algorithm.

**Theorem 1.** The \( E(TTR) \) of the generalized Random algorithm is equal to \( \frac{A^R_i A^R_j}{A^Q_i A^Q_j - A^{R_i + R_j}_Q} \) under the symmetric model where \( Q \) is the number of all channels and \( A^i_j \) is the number of possible permutations of \( i \) objects from a set of \( j \).

**Proof.** In time slot \( k \), user \( i \) hops on \( R_i \) different channels \( S^k_i \) and user \( j \) on \( R_j \) different channels \( S^k_j \) (Figure 3.1). If any channel in \( S^k_i \) is the same as one channel in \( S^k_j \), the rendezvous is achieved. In any time slot, each channel has \( Q \) choices. The probability that all \( m \) channels of user \( i \) are not equal to any channel of user \( j \) is \( \frac{A^{R_i + R_j}_Q}{A^Q_i A^Q_j} \). So \( p(TTR = t) = \left( \frac{A^{R_i + R_j}_Q}{A^Q_i A^Q_j} \right)^{t-1} \left( 1 - \frac{A^{R_i + R_j}_Q}{A^Q_i A^Q_j} \right) \). The \( E(TTR) = \sum_{i=1}^{+\infty} t \times p(TTR = t) = \sum_{i=1}^{+\infty} t \times \left( \frac{A^{R_i + R_j}_Q}{A^Q_i A^Q_j} \right)^{t-1} \left( 1 - \frac{A^{R_i + R_j}_Q}{A^Q_i A^Q_j} \right) = \frac{A^{R_i}_Q A^{R_j}_Q}{A^Q_i A^Q_j - A^{R_i + R_j}_Q} \). \( \square \)

**Theorem 2.** The \( E(TTR) \) of the generalized Random algorithm is equal to \( \frac{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|}}{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|} - A^{R_i}_{|C^i| - G} A^{R_j}_{|C^j| - G}} \) under the asymmetric model where \( |C^i| \) and \( |C^j| \) are the numbers of available channels of two users and \( G \) is the number of commonly-available channels of the two users.

**Proof.** In time slot \( k \), the probability that all \( R_i \) channels of user \( i \) are not equal to any channel of user \( j \) is \( \frac{A^{R_i}_{|C^i| - G} A^{R_j}_{|C^j| - G}}{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|}} \). So \( p(TTR = t) = \left( \frac{A^{R_i}_{|C^i| - G} A^{R_j}_{|C^j| - G}}{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|}} \right)^{t-1} \left( 1 - \frac{A^{R_i}_{|C^i| - G} A^{R_j}_{|C^j| - G}}{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|}} \right) \). The \( E(TTR) = \sum_{i=1}^{+\infty} t \times p(TTR = t) = \sum_{i=1}^{+\infty} t \times \left( \frac{A^{R_i}_{|C^i| - G} A^{R_j}_{|C^j| - G}}{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|}} \right)^{t-1} \left( 1 - \frac{A^{R_i}_{|C^i| - G} A^{R_j}_{|C^j| - G}}{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|}} \right) = \frac{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|}}{A^{R_i}_{|C^i|} A^{R_j}_{|C^j|} - A^{R_i}_{|C^i| - G} A^{R_j}_{|C^j| - G}} \). \( \square \)

### 3.2.2 Generalized Existing Rendezvous Algorithms

In the literature, several rendezvous algorithms have been proposed for CRNs. They implicitly assume one radio per user. In this subsection, we generalize these existing...
rendezvous algorithms such that each user can use multiple radios for faster rendezvous. We consider two strategies to generalize these algorithms to use multiple radios:

1. **Independent Sequence**: Apply an existing rendezvous algorithm to independently generate a CH sequence for each radio. If this algorithm always generates the same CH sequence using a deterministic method, then the sequence is rotated by $x$ positions where $x$ is a randomly generated integer. For example, suppose a user is equipped with two radios and an existing rendezvous algorithm generates two CH sequences $\{s_1, s_2, \ldots \}$ and $\{r_1, r_2, \ldots \}$. In the first time slot, the two radios hop on channels $s_1$ and $r_1$ respectively. In the second time slot, the two radios hop on channels $s_2$ and $r_2$ respectively.

2. **Parallel Sequence**: Apply an existing algorithm to generate a CH sequence and apply this CH sequence on all radios in parallel. For example, suppose a user is equipped with three radios and an existing rendezvous algorithm generates the CH sequence $\{s_1, s_2, s_3, s_4, s_5, s_6, \ldots \}$. In the first time slot, the three radios hop on channels $s_1$, $s_2$ and $s_3$, respectively. In the second time slot, the three radios hop on channels $s_4$, $s_5$ and $s_6$, respectively.

We propose two schemes (namely, independent sequence and parallel sequence) to generalize the existing single-radio algorithms to use multiple radios. While the ideas of these schemes are simple, their theoretical analysis is not trivial. When different nodes have different number of radios, only the independent sequence scheme works. Nevertheless, the parallel sequence scheme also has its own merit: when the nodes have the same number of radios, the parallel sequence scheme works and it gives better performance than the independent sequence scheme. Therefore, we report both schemes in our paper.

When the Independent Sequence strategy is applied to generalize an existing rendezvous algorithm to use multiple radios for faster rendezvous, the steps are given in Algorithm 2.
Algorithm 2: Independent Sequence

Require: $Q$, $R_i$, $A$, $C_i$ // an existing algorithm denoted by $A$ and user $i$

1: $t = 1$
2: $\vec{S}_t^i = \{S_{t1}^i, S_{t2}^i, S_{t3}^i, ..., S_{tR_i}^i\}$
3: for $k = 1$ to $R_i$ do
4: if $A$ is a deterministic method and the sequence is $S_{At}$ then
5: $S_{Akt} = S_{At}$ rotated by random positions
6: else
7: $S_{Akt}$ is generated by algorithm $A$
8: end if
9: end for
10: while not rendezvous do
11: for $k = 1$ to $R_i$ do
12: $S_{tk}^i = S_{Akt}$
13: end for
14: $t = t + 1$
15: Attempt rendezvous on $\vec{S}_t^i$
16: end while
In line 3, since the user has $R_i$ radios, it generates $R_i$ independent CH sequences by an existing algorithm. In lines 4-8, $k$-th sequences are generated by this algorithm. In line 12, the $k$-th sequence is performed in the $k$-th radio. Let $MTTR_0$ be the MTTR of the existing rendezvous algorithm using one radio. The following theorem gives the MTTR of Algorithm 2.

**Theorem 3.** If two users perform an existing algorithm on multiple radios with Independent Sequence, the maximum time-to-rendezvous (MTTR) is equal to $MTTR_0$ under both the symmetric model and the asymmetric model.

**Proof.** When the users run an existing algorithm independently on multiple radios, each pair of radios of the two users will achieve rendezvous on or before $MTTR_0$. As a result, the two users will achieve a guaranteed rendezvous on or before $MTTR_0$ by any pair of radios. So $MTTR \leq MTTR_0$. Next we prove $MTTR \geq MTTR_0$. Consider a worst case in which all the $m$ radios of a user use the same sequence (e.g., the existing algorithm happens to independently generate $m$ identical sequences). In this case, the rendezvous is the same as the original algorithm. So $MTTR \geq MTTR_0$. Overall, the MTTR is equal to $MTTR_0$. \hfill \Box

Among the existing rendezvous algorithms, the Jump-Stay algorithm performs well in terms of MTTR and $E(TTR)$ [25]. When the Jump-Stay algorithm is generalized by the Independent Sequence strategy, we have the following results.

**Corollary 1.** Under the symmetric model, any two users performing the Jump-Stay algorithm on multiple radios with Independent Sequence achieve rendezvous in at most $3P$ time slots which is an upper bound of MTTR. The $E(TTR)$ is not greater than $\frac{5P-4}{3} + \frac{1}{3Q^{R_i+R_j-1}} \times (2P + 1 + \frac{1}{P})$, where $P$ is the smallest prime number which is not smaller than $Q$.

**Proof.** Figure 3.2 illustrates how the sequences are generated. Suppose that $Q = 6$ and $P = 7$. User $i$ has 3 radios with $(i_1 = 1, r_1 = 1)$, $(i_2 = 1, r_2 = 2)$, and $(i_3 = 1, r_3 = 3)$. User $j$ has 2 radios with $(i_1 = 2, r_1 = 1)$ and $(i_2 = 2, r_2 = 2)$. The
Rendezvous is achieved in time slot 2 when user \( i \) hops on channels \((4, 1, 3)\) while user \( j \) hops on channels \((3, 4)\).

Figure 3.3 shows the six cases when users perform the Jump-Stay algorithm independently on multiple radios. Step-length \( r \) takes integer value in \([1, Q]\); two users select different step-lengths, \( r_k_1 \neq r_k_2 \), with probability \((1 - \frac{1}{Q^{R_i + R_j - 1}})\) while select the same step-length with probability \(\frac{1}{Q^{R_i + R_j - 1}}\). Thus, we compute the occurrence probabilities of six cases when all users have multiple radios are

- \(2 \times \frac{P + 1}{2P} \times (1 - \frac{1}{Q^{R_i + R_j - 1}})\) and \(\frac{1}{3} \times \frac{P - 1}{P} \times \frac{1}{Q^{R_i + R_j - 1}}\), respectively. So the upper bound of MTTR should be \(3P\) and \(E(TTR)\) should be:

\[
\frac{5P - 4}{3} + \frac{1}{3Q^{R_i + R_j - 1}}(2P + 1 + \frac{1}{P}).
\]

Similar to the proof in [9], we can prove that the upper bound of \(E(TTR)\) under the asymmetric model is smaller than \(\frac{5P - 4}{3} + \frac{1}{3Q^{R_i + R_j - 1}}(2P + 1 + \frac{1}{P}) + 3QP(P - G)\).

When the Parallel Sequence strategy is applied to generalize an existing rendezvous algorithm (say, algorithm \( A \)) to use multiple radios for faster rendezvous, the steps are given in Algorithm 3.

In line 3, an existing rendezvous algorithm is run to generate one CH sequence \( S_A \). In lines 5-7, this CH sequence \( S_A \) is applied in parallel to the \( m \) radios of the user. The following theorem gives the MTTR of Algorithm 3:

**Theorem 4.** Suppose two users are equipped with \( R_i = R_j = R \) radios. Under the
Figure 3.3: Six cases under the symmetric model. Remarks: We say $r_i = r_j$ (Figure 3.3(b) and Figure 3.3(f)) if all step-lengths of CH sequences with user $i$ are identical and are equal to those with user $j$, and $r_i \neq r_j$ (Figure 3.3(a) and Figure 3.3(e)) otherwise.
**Algorithm 3: Parallel Sequence**

**Require:** $Q, R_i, A, C_i$ // an existing algorithm denoted by $A$ and user $i$

1: $t = 1$
2: $\vec{S}_i = \{s_{i1}, s_{i2}, s_{i3}, ..., s_{iR_i}\}$
3: Generate sequence $S_{At}$ by algorithm $A$
4: while not rendezvous do
5: for $k = 1$ to $R_i$ do
6: $s_{tk}^i = S_{A((t-1) \times R_i + k)}$
7: end for
8: $t = t + 1$
9: Attempt rendezvous on $\vec{S}_i$
10: end while

Symmetric and the asymmetric models, when the Parallel Sequence strategy is applied in conjunction with an existing rendezvous algorithm with $MTTR = MTTR_0$, rendezvous can be achieved in at most $\lceil \frac{MTTR_0}{R} \rceil$ time slots.

**Proof.** Since the CH sequence is applied in parallel to the $R$ radios of the user, the user will finish the hopping sequence in time slot $\lceil \frac{MTTR_0}{R} \rceil$. Figure 3.4 shows the new rendezvous achieved when the users are equipped with the same number of radios. The channel at time slot $T$ in the hopping sequence when the user is equipped with one radio will appear in time slot $\lceil \frac{T}{R} \rceil$ when the user is equipped with $R$ radios. If user $i$ and user $j$ achieve rendezvous on channel $k$ in time slot $TTR$ before $MTTR_0$ in algorithm $A$, then the two users will hop on this channel and achieve rendezvous at $\lceil \frac{TTR}{R} \rceil$ before time slot $\lceil \frac{MTTR_0}{R} \rceil$. Therefore, rendezvous can be achieved in at most $\lceil \frac{MTTR_0}{R} \rceil$.

**Corollary 2.** Suppose all users are equipped with $R_i = R_j = R$ radios. Under the symmetric model, when the Parallel Sequence strategy is applied in conjunction with the Jump-Stay algorithm, rendezvous can be achieved in at most $\lceil \frac{3P}{R} \rceil$ time slots. The $E(TTR)$ is upper bounded by $(\frac{5P}{3} + \frac{11}{3} + \frac{1}{Q})/R$, where $P$ is the smallest prime number which is not smaller than $Q$.  

31
Figure 3.4: Rendezvous of an existing algorithm with Parallel Sequence when $R_i = R_j = 2$.

The proof of Corollary 2 is very similar to the proof of Theorem 4 and we do not repeat the details.

Overall, the Independent Sequence strategy can guarantee rendezvous even when the users have different number of radios, while the Parallel Sequence strategy can guarantee rendezvous only when the users have the same number of radios but it can better exploit these radios to achieve smaller MTTR.

### 3.2.3 New Algorithm

In this subsection, we design and analyze a new rendezvous algorithm that specifically exploits multiple radios for more efficient rendezvous. Our basic idea is to assign one of two possible roles, called *general radio* and *dedicated radio*, to each radio. The rendezvous is expected to be achieved between the general radios of one user and the dedicated radio of the other. Our RPS algorithm generates CH sequences in rounds and the length of each round is in inversely proportional to the number of general radios. The upper-bounds of MTTR and $E(TTR)$ of RPS are the expression of the length of the round (later shown in proof of Theorem 5 & 6). Therefore,
large number of general radios leads to a shorter round which consequently gives smaller upper-bounds of MTTR and E(TTR). In RPS, we use only one radio as the dedicated radio and the remaining radios as the general radios to optimize the rendezvous performance. Users hop on available channels in the general radios while stay on a specific available channel in the dedicated radio. Suppose that a user is equipped with \( R_i \) radios. In our paper, each node is equipped with multiple radios where each radio has a role: either ”dedicated” or ”general”. So we call the new algorithm \textit{Role-based Parallel Sequence} (RPS). It is described as follows.

i) All radios are divided into two groups, \((R_i-1)\text{ general radios}\) and one \textit{dedicated radio}.

ii) A starting index \( R_i \) is randomly selected from \([1, P - 1]\) and a step-length \( r \) is randomly selected from \([1, P - 1]\), where \( P \) is the smallest prime number which is not smaller than \( Q \).

iii) \((R_i - 1)\text{ general radios}\) hop on \( P \) channels with step-length \( r \) in the round-robin manner. For example, in Figure 3.5 \( P = 7 \), the starting index is 1 and step-length is 2. The first channel in the channel hopping sequence is 1 and the \( k\text{-th} \) channel is \( (i + r \cdot k) \% P \) \(( (1+2k) \% 7 \text{ in this example})\). The CH sequence is \( \{1, 3, 5, 7, 2, 4, 6, 1, 3, \ldots \} \) and two general Radios 1 and 2 hop on this sequence in parallel as follows. Radio 1 hops on subsequence \( \{1, 5, 2, 6, \ldots \} \) and Radio 2 hops on subsequence \( \{3, 7, 4, 1, \ldots \} \).

iv) \textit{Dedicated radio} stays on one channel for \( \lceil \frac{P}{R_i-1} \rceil \) time slots and switches to next channel for the same duration, where the channel is taken from \([1, Q]\) in a round-robin manner. For example, in Figure 3.5 dedicated radio 3 stays on channel 1 for 4 time slots and then switches to channel 2.

v) If the channel selected in iii) and iv) is not available to the user, randomly select an available channel.

The algorithm is formally presented as Algorithm 4.
Algorithm 4: RPS Algorithm

Require: $Q$, $R_i$, $C_i$

1: $t = 1$
2: $S^i_t = \{S^i_{t1}, S^i_{t2}, S^i_{t3}, ..., S^i_{tR_i}\}$
3: $P =$ the smallest prime number not smaller than $Q$
4: $i = \text{RandomSelect}(1, P)$
5: $r = \text{RandomSelect}(1, Q)$
6: while not rendezvous do
7:   for $k = 1$ to $R_i - 1$ do
8:       $S^i_{tk} = (i + ((t - 1) \times (R_i - 1) + k - 1) \times r - 1) \mod P + 1$
9:       if $S^i_{tk} \ge Q$ then
10:          $S^i_{tk} = S^i_{tk} \mod Q$
11:     end if
12:     if $S^i_{tk} \notin C^i$ then
13:         $S^i_{tk} = \text{RandomSelect}(C^i)$
14:     end if
15:   end for
16:   $S^i_{tR_i} = (\lceil \frac{t}{R_i - 1} \rceil - 1) \mod Q + 1$
17:   if $S^i_{tR_i} \notin C^i$ then
18:       $S^i_{tR_i} = \text{RandomSelect}(C^i)$
19:   end if
20:   $t = t + 1$
21:   Attempt rendezvous on $S^i_t$
22: end while
In line 4, starting index $i$ and step-length $r$ are preselected randomly. In lines 7-15, the $(R_i - 1)$ general radios will hop on continuous $(R_i - 1)$ channels with $i$ and $r$. In line 16, the dedicated radio will switch to the next channel after $\lceil \frac{P}{R_i - 1} \rceil$ time slots. Lines 12-14 and 17-19 ensure that the channels are available to the user.

Figure 3.6 shows rendezvous of two users by performing the RPS algorithm. Suppose that $|C| = Q = 3$, $P = 3$. User $i$ is equipped with 3 radios and the available channels are $\{1, 3\}$. It starts with channel 1. Step-length is 2. Each round consists of $\lceil \frac{P}{3 - 1} \rceil = 2$ time slots. User $j$ is equipped with 2 radios and the available channels are $\{2, 3\}$. It starts with channel 2. Step-length is 2. Each round consists of $\lceil \frac{P}{3 - 1} \rceil = 3$ time slots. In this example, there will be a random available channel in the position with underline since the channel based on the algorithm is not available to the user. In Figure 3.6(a), two users hop on channel 3 in time slot 4. Rendezvous is achieved by the dedicated radio of user $j$ and the general radios of the user $i$. User $j$ will stay on channel 3 for 3 time slots. User $i$ has a permutation of all channels in any 3 consecutive time slots. Since there is at least one commonly-available channel between them, channel 3 in this example, there must be a rendezvous between the dedicated radio of user $j$ and the general radios of user $i$. However, before this rendezvous, there is an earlier one in time slot 3. It is achieved by the general radios of user $i$ and the general radio of the user $j$. In Figure 3.6(b), user $j$ starts later than user $i$ for 1 time slot. When the dedicated radio of user $i$ (with a shorter round) stays on the commonly-available channel (channel 3) for 2 time slots, the general radios of user $j$ do not have a permutation of all available channels in 2 consecutive
time slots. Therefore, it is possible that channel 3 is not in these 2 time slots. We
cannot guarantee a rendezvous between the dedicated radio of user $i$ and the general
radios of user $j$. The guaranteed rendezvous realized in time slot 5.

Now we theoretically analyze the RPS algorithm. Specifically, we derive the up-
per bounds of MTTR of the RPS algorithm in Theorem 5 and Theorem 6 under the
symmetric model and the asymmetric model, respectively. In addition, we derive
the upper bounds of $E(TTR)$ of the RPS algorithm under both the symmetric and
asymmetric models. These upper bounds reveal two points: i) the order of perfor-
mance, and ii) whether the proposed method can achieve guaranteed rendezvous.

**Theorem 5.** Under the symmetric model, let $R_i$ and $R_j$ denote the numbers of
radios of two users, respectively. If $R_i \neq R_j$, the MTTR of the RPS algorithm is
upper bounded by $(2 \times \lceil \frac{P}{\max(R_i, R_j)} - 1 \rceil - 1)$ and the $E(TTR)$ of the RPS algorithm is
upper bounded by $\lceil \frac{P}{\max(R_i, R_j)} - 1 \rceil + \frac{1}{2} \times \lceil \frac{P}{\min(R_i, R_j)} - 1 \rceil^2$; if $R_i = R_j$, the MTTR of the
RPS algorithm is upper bounded by $\lceil \frac{P}{R_i} \rceil$ and the $E(TTR)$ of the RPS algorithm

36
Figure 3.7: Four cases of RPS under the symmetric model.

is upper bounded by $\lceil \frac{P}{R_i - 1} \rceil$.

Proof. Let user $i$ be equipped with $R_i$ radios while user $j$ be equipped with $R_j$ radios. Figure 3.7 lists the four cases of rendezvous under symmetric model. Figure 3.7(a), 3.7(b), and 3.7(c) happen when $R_i \neq R_j$. Since the results depend on which user starts hopping first, we assume $R_i < R_j$, i.e., $\lceil \frac{P}{R_i - 1} \rceil > \lceil \frac{P}{R_j - 1} \rceil$. Figure 3.7(d) happens when $R_i = R_j$. A remarkable distinction between them is whether the lengths of each round of the two users are the same.

1. Case 1: Figure 3.7(a). $l' \leq \lceil \frac{P}{R_j - 1} \rceil$ implies that there is a permutation of all channels before the dedicated radio of user $i$ transfers to the next channel. The rendezvous is achieved between general radios of user $j$ and dedicated radio of user $i$ during the first $\lceil \frac{P}{R_j - 1} \rceil$ time slots. That is, $TTR \leq \lceil \frac{P}{R_j - 1} \rceil$.

2. Case 2: Figure 3.7(b). In this case, $l' < \lceil \frac{P}{R_j - 1} \rceil$ implies that there is no enough time slots for user $j$ to have permutation of all channels before the
dedicated radio of user $i$ transfers to next channel. The rendezvous can only be guaranteed between general radios of user $i$ and dedicated radio of user $j$ during the first $2 \times \lceil \frac{P}{R_j-1} \rceil - 1$ time slots. That is, $TTR \leq 2 \times \lceil \frac{P}{R_j-1} \rceil - 1$.

3. Case 3: Figure 3.7(c). User $j$ starts firstly. User $j$ has a permutation of all channels in any $\lceil \frac{P}{R_j-1} \rceil$ consecutive time slots. When user $i$ starts, it will stay on one channel for $\lceil \frac{P}{R_i-1} \rceil$ time slots. $\lceil \frac{P}{R_i-1} \rceil > \lceil \frac{P}{R_j-1} \rceil$. Thus, a rendezvous is guaranteed before $\lceil \frac{P}{R_j-1} \rceil$ time slots.

4. Case 4: Figure 3.7(d) $R_i = R_j$. The rendezvous is achieved before the first $\lceil \frac{P}{R_i-1} \rceil$ (or $\lceil \frac{P}{R_j-1} \rceil$) time slots.

When $R_i \neq R_j$, in the above analysis, $\lceil \frac{P}{R_i-1} \rceil$ is replaced by $\lceil \frac{P}{\min(R_i,R_j)-1} \rceil$ and $\lceil \frac{P}{R_j-1} \rceil$ by $\lceil \frac{P}{\max(R_i,R_j)-1} \rceil$. According to the analysis of these cases, we prove that the MTTR is $(2 \times \lceil \frac{P}{\max(R_i,R_j)-1} \rceil - 1)$. Combining with the occurrence probabilities we derive an upper bound of $E(TTR)$ under the symmetric model. The $E(TTR) \leq \frac{1}{2} \times \left[ \frac{\frac{P}{\min(R_i,R_j)-1}}{\lceil \frac{P}{\max(R_i,R_j)-1} \rceil} + \frac{\frac{P}{\max(R_i,R_j)-1}}{\lceil \frac{P}{\min(R_i,R_j)-1} \rceil} \right] + \frac{1}{2} \left[ \frac{\frac{P}{\max(R_i,R_j)-1}}{\lceil \frac{P}{\min(R_i,R_j)-1} \rceil} \right] \leq \frac{P}{\max(R_i,R_j)-1} + \frac{\left( \frac{P}{\max(R_i,R_j)-1} - 1 \right)^2}{2 \times \lceil \frac{P}{\min(R_i,R_j)-1} \rceil}.

There is only one case when $R_i = R_j$, the MTTR and the upper bound of $E(TTR)$ are both $\lceil \frac{P}{R_i-1} \rceil$ or $\lceil \frac{P}{R_j-1} \rceil$.

\textbf{Theorem 6.} Under the asymmetric model, let $R_i$ and $R_j$ denote the numbers of radios of two users, respectively. If $R_i \neq R_j$, the MTTR of the RPS algorithm is upper bounded by $(2 \times \lceil \frac{P}{\max(R_i,R_j)-1} \rceil - 1) + \frac{P}{\min(R_i,R_j)-1} \times (Q - G)$ and the $E(TTR)$ of the RPS algorithm is upper bounded by $\left[ \frac{P}{\max(R_i,R_j)-1} \right] + \frac{\left( \frac{P}{\max(R_i,R_j)-1} - 1 \right)^2}{2 \times \lceil \frac{P}{\min(R_i,R_j)-1} \rceil} + \frac{P}{\min(R_i,R_j)-1} \times (Q - G)$; if $R_i = R_j$, the MTTR of the RPS algorithm is upper bounded by $\lceil \frac{P}{R_i-1} \rceil \times (Q - G + 1)$ and the $E(TTR)$ of the RPS algorithm is upper bounded by $\left[ \frac{P}{R_i-1} \right] \times (Q - G + 1)$.

\textbf{Proof.} Under the asymmetric model, since the available channel sets of two users are different from each other, the users may achieve many potential rendezvous (rendezvous 1 to $(Q - G)$) in Figure 3.8. In Figure 3.8(a), user $j$ has a permutation
of all channels before $\lceil \frac{P_{R_j}}{R_i-1} \rceil$ and the dedicated radio of user $i$ stays on one channel during this period. There is a potential rendezvous before $\lceil \frac{P_{R_j}}{R_i-1} \rceil$; this channel may not be a commonly-available channel to all users. The next potential rendezvous can be guaranteed in the next round of user $i$ ($\lceil \frac{P_{R_j}}{R_i-1} \rceil$ to $2 \times \lceil \frac{P_{R_j}}{R_i-1} \rceil$ in Figure 3.8) because only after these time slots the dedicated radio of user $i$ will transfer to the next channel. We can say, under asymmetric model, we expect a rendezvous between the dedicated radio of the user with less radios (user $i$) and the general radios of the user with more radios (user $j$). The worst case is repeating the rendezvous under symmetric model for $(Q - G)$ times. Above all, the MTTR should be equal to or smaller than $2 \times \lceil \frac{P}{\max\{R_i,R_j\}} - 1 \rceil - 1 + \lceil \frac{P}{\min\{R_i,R_j\}} - 1 \rceil \times (Q - G)$. We assume a uniform distribution that a commonly-available channel appears on the 1st to $(Q - G)$-th potential rendezvous. The upper bound of $E(TTR)$ when $R_i \neq R_j$ extend for $\lceil \frac{P}{\min\{R_i,R_j\}} - 1 \rceil \times (Q - G)$ in all cases. In this way, the $E(TTR) \leq \lceil \frac{P}{\max\{R_i,R_j\}} - 1 \rceil + \frac{\lceil \frac{P}{\max\{R_i,R_j\}} - 1 \rceil^2}{2 \times \lceil \frac{P}{\min\{R_i,R_j\}} - 1 \rceil} + \lceil \frac{P}{\min\{R_i,R_j\}} - 1 \rceil \times (Q - G)$.

And similarly, the MTTR and the upper bound of $E(TTR)$ is $\lceil \frac{Q}{R_i-1} \rceil \times (Q - G + 1)$ when $R_i = R_j$.

3.3 Simulation

We built a simulator in Visual Studio 2010 to evaluate the effectiveness of the proposed approach (i.e., using multiple radios for rendezvous) and the proposed rendezvous algorithm (i.e., the Role-based Parallel Sequence (RPS) algorithm). When each user has a single radio, we consider the following algorithms for comparison:

i) the Random algorithm [35] (it is the most simple rendezvous algorithm), and
ii) the Jump-Stay algorithm [25], MMC algorithm [35], HH [37] (They are recently proposed and they have good performance). When each user has multiple radios, we consider the following algorithms for comparison: i) the generalized Random algorithm (Section 3.2.1), ii) the generalized Jump-Stay/MMC/HH algorithm with the Independent Sequence strategy (Section 3.2.2), iii) the generalized Jump-Stay/MMC/HH algorithm with the Parallel Sequence strategy (Section 3.2.2), and
Figure 3.8: Four cases of RPS under the asymmetric model.
iv) the RPS algorithm (Section 3.2.3). The performance is measured in terms of the average TTR and the maximum TTR, where TTR is counted as the number of time slots required to achieve rendezvous. We consider both the symmetric model and the asymmetric model.

We use the notation \((R_i, R_j)\) to denote the case that two users are equipped with \(R_i\) and \(R_j\) radios respectively. We consider the following key parameters: the number of channels \(Q\) in the whole channel set is varied from 10 to 100. Under the symmetric model, all channels are available to all users. Under the asymmetric model, we introduce a parameter \(\theta\) \((0 < \theta \leq 1)\) and randomly select channels from the channel set, such that the average number of channels available to a user is equal to \(\theta Q\). If \(\theta \times Q < 1\), we reset \(\theta Q\) to 1. For each set of parameter values, we perform 100,000 independent runs and then compute the average TTR and the maximum TTR.

### 3.3.1 Effectiveness of Proposed Approach

In this subsection, we demonstrate that multiple radios can effectively improve the rendezvous performance.

**Under the Symmetric Model**

Figure 3.9 shows the performance of the Random algorithm under the symmetric model. It can be seen that: i) multiple radios can significantly reduce both the average TTR and the maximum TTR, and ii) the performance improvement is especially significant when the number of radios per user is increased from a small value. For example, we suppose that each time slot has duration of 20ms \([25]\). When there are 50 channels and the number of radios per user is increased from 1 to 2, the average TTR is decreased from 50.02 (1.00s) to 12.88 (0.26s) (i.e., 74.25% reduction) while the maximum TTR is decreased from 813 (16.26s) to 217 (4.34s) (i.e., 73.31% reduction). When the number of radios per user is increased from 2 to 3, the average TTR is decreased from 12.88 (0.26s) to 6.01 (0.12s) (i.e., 53.34% reduction).
reduction) while the maximum TTR is decreased from 217 (4.34s) to 88 (1.76s) (i.e., 59.45% reduction). Therefore, a cost-effective tradeoff between cost of the radios and performance is to select 2 to 4 radios per user. In addition, we find that adoption of multiple radios can reduce the overall energy cost on rendezvous. For example, when the number of radios is increased from 1 to 3, the average time spent on rendezvous (E(TTR)) is reduced by 87.98% (more than 3 times). We assume that one radio performing rendezvous in one time slot costs one unit of energy, denoted by $U$. Then the expected energy consumption is $E(TTR) \times U \times (R_i + R_j)$ when the two users are equipped with $R_i$ and $R_j$ radios, respectively. Figure 3.9(c) shows the expected consumption when each user is equipped with single radio and multiple radios. For example, when there are 50 channels, the energy consumption of users with 1, 2, 3 and 4 radios are $50.02U$, $25.76U$, $18.03U$ and $14.50U$, respectively. Figure 3.9(d) shows the average TTR under different $(R_i, R_j)$ when $M = 50$. We find that the average TTR drops significantly with the increase of number of radios.

Figure 3.10 shows the performance of the Jump-Stay algorithm under the symmetric model. We observe similar properties as those in Figure 3.9. For example, when the number of channels is 50 and the number of radios is increased from 1 to 2, the average TTR is reduced from 28.99 to 10.22 (i.e, 64.75% reduction) while the maximum TTR is decreased from 156 to 81 (i.e, 48.08% reduction). According to Theorem 3 and Corollary 1, however, the upper bound of $E(TTR)$ is decreased from 91.33 to 85.01 while the MTTR remains 265. The theoretical results are much bigger than experimental results. It is because that Jump-Stay algorithm randomly selects available channels to replace the unavailable channels on the generated CH sequence, which practically lowers TTR in simulation.

**Under the Asymmetric Model**

Figure 3.11 shows the performance of Random algorithm under the asymmetric model. We let $\theta = 0.5$ and $G = 0.3Q$ (i.e., each user has $0.5Q$ channels and each pair of users have $0.3Q$ commonly-available channels). It can be seen that multiple
Figure 3.9: Comparison of multi-radio and single-radio under the symmetric model (Random).
radios can reduce both the average TTR and the maximum TTR significantly under the asymmetric model. For example, when there are 50 channels and the number of radios per user is increased from 1 to 2, the average TTR is decreased from 41.66 to 13.67 (i.e., 61.19% reduction) while the maximum TTR is decreased from 635 to 204 (i.e., 67.87% reduction). When the number of radios per user is increased from 2 to 3, the average TTR is decreased from 13.67 to 6.50 (i.e., 52.45% reduction) while the maximum TTR is decreased from 204 to 102 (i.e., 50% reduction). Figure 3.12 shows the performance of the Jump-Stay algorithm with Independent Sequence under the asymmetric model. For example, when there are 50 channels and the number of radios per user is increased from 1 to 2, the average TTR is decreased from 32.21 to 12.11 (i.e., 62.40% reduction) while the maximum TTR is decreased from 777 to 311 (i.e., 59.97% reduction).

3.3.2 Property of Proposed Approach: Influence of Radio Allocation

In this subsection, we study a basic property of the proposed approach: when the total number of radios for two users is fixed (i.e., \( R_i + R_j \) is a constant), how does
Figure 3.11: Comparison of multi-radio and single-radio under the asymmetric model (Random).

Figure 3.12: Comparison of multi-radio and single-radio under the asymmetric model (Jump-Stay).
the allocation \((R_i, R_j)\) for different \(R_i\) and \(R_j\) affect the rendezvous performance? For example, if there are four radios, does \((2, 2)\) give better rendezvous performance than \((1, 3)\) or \((3, 1)\)? In theory, the MTTR of the RPS algorithm is decided by the minimum value of \(R_i\) and \(R_j\). The MTTR of Random algorithm \([35]\) and other existing algorithms is almost independent of \(R_i\) and \(R_j\). All results show that even allocation has the best performance.

Under the symmetric model, we let \(R_i + R_j = 6\) and \(G = Q\). Figure 3.13 shows the comparison of the average TTR and the maximum TTR of Random algorithm between the three combinations of \((R_i, R_j)\) which are \((1, 5)\), \((2, 4)\) and \((3, 3)\). We can see that the combination \((3, 3)\) has the best performance on both the average TTR and the maximum TTR. For example, when there are 50 channels, the average TTR of \((1, 5)\), \((2, 4)\) and \((3, 3)\) are 10.41, 6.71 and 6.02 while the maximum TTR are 153, 120 and 95, respectively. It is consistent with the theoretical results. We also do the same comparison when we apply other existing algorithms on multiple radios. Figure 3.14 shows the comparison of the average TTR and the maximum TTR of the Jump-Stay algorithm with Independent Sequence between the three combinations of \((R_i, R_j)\) which are \((1, 5)\), \((2, 4)\) and \((3, 3)\). Figure 3.15 shows the comparison of the average TTR and the maximum TTR of the RPS algorithm between the two combinations of \((R_i, R_j)\) which are \((2, 4)\) and \((3, 3)\).

Under the asymmetric model, we let \(R_i + R_j = 6\), \(\theta = 0.75\) and \(G = 0.5Q\). Figure 3.16 shows the results of Random algorithm between the three combinations of \((R_i, R_j)\). The combination \((3, 3)\) has the best performance on both the average TTR and the maximum TTR. For example, when there are 50 channels, the average TTR of \((1, 5)\), \((2, 4)\) and \((3, 3)\) are 11.56, 7.42 and 6.08 while the maximum TTR are 188, 135 and 121, respectively. Figure 3.17 shows the results of the average TTR and the maximum TTR of the Jump-Stay algorithm with Independent Sequence between the three combinations of \((R_i, R_j)\). Figure 3.18 shows the comparison of the average TTR and the maximum TTR of the RPS algorithm between the three combinations.
Figure 3.13: Comparison of different allocation of \((R_i, R_j)\) when \(R_i + R_j = 6\) under the symmetric model (Random).

Figure 3.14: Comparison of different allocation of \((R_i, R_j)\) when \(R_i + R_j = 6\) under the symmetric model (Jump-Stay).
Figure 3.15: Comparison of different allocation of $(R_i, R_j)$ when $R_i + R_j = 6$ under the symmetric model (RPS).

Figure 3.16: Comparison of different allocation of $(R_i, R_j)$ when $R_i + R_j = 6$ under the asymmetric model (Random).
Figure 3.17: Comparison of different allocation of $(R_i, R_j)$ when $R_i + R_j = 6$ under the asymmetric model (Jump-Stay).

Figure 3.18: Comparison of different allocation of $(R_i, R_j)$ when $R_i + R_j = 6$ under the asymmetric model (RPS).
3.3.3 Property of Proposed Approach: Influence of $G$ for Fixed $\theta$

In this subsection, we study another basic property of the proposed approach: how does the number of commonly-available channels between two users affect the rendezvous performance? We let $\theta = 0.5$, $G = 0.1Q$, $G = 0.25Q$ and $G = 0.5Q$, respectively. $R_i = 3$ and $R_j = 4$. Figure 3.19 shows the results when we apply the Jump-Stay algorithm to multiple radios with Independent Sequence. When there are 50 channels, the average TTR of three scenarios are 10.69, 5.03 and 2.56 and the maximum TTR are 242, 119 and 35. Figure 3.20 shows the results of the RPS algorithm. When there are 50 channels, the average TTR of three scenarios are 12.83, 5.54 (i.e. 56.82% reduction) and 2.65 and the maximum TTR are 274, 104 (i.e. 62.04% reduction) and 28. According to Theorem 6, however, when $G$ is increased from $0.1Q$ to $0.25Q$, the upper bound of $E(TTR)$ is decreased from 1215.72 to 1030.22 (i.e., 15.25% reduction) and the MTTR is decreased from 1200.83 to 1041.33 (i.e., 15.45% reduction). It reveals that both the average TTR and the maximum TTR drop faster than theoretical results when $G$ increases.
We now study the performance of the proposed rendezvous algorithm and the generalized versions of the existing algorithms. We want to emphasize that all following simulations are based on multiple radios. All of them have significant improvement compared with existing algorithms. In this part, we compare the average TTR and the maximum TTR under different values of $R_i$, $R_j$, $\theta$ and $G$. We compare Generalized Random Algorithm (described in Section 3.2.1), Generalized Existing Algorithms (Section 3.2.2) and RPS Algorithm (Section 3.2.3) under the scenario when all users are equipped with multiple radios. For the existing algorithms, we apply the Jump-Stay [25], MMC [35] and HH [37] algorithm with both Independent Sequence and Parallel Sequence. HH is for heterogeneous system. We only apply it under the asymmetric model and we expand it with random replacement. When the channel in the sequence is not available, we randomly select an available one to replace it.
Figure 3.21: Comparison of different algorithms under the symmetric model when \((R_i, R_j) = (3,3)\).

Under the Symmetric Model

\(R_i = R_j\): Under the symmetric model, we let \(\theta = 1\) and \(G = Q\). Figure 3.21 shows the comparison of the average TTR and the maximum TTR between the six different algorithms when \(R_i = R_j = 3\). As shown in Figure 3.21(a) when there are 50 channels, the average TTR of RPS, JS/Independent, JS/Parallel, Random, MMC/Independent and MMC/Parallel are 5.55, 5.40, 6.68, 6.02, 5.96 and 5.93 respectively. There is no large gap between different algorithms. However, the maximum TTR are 27, 88, 56, 88, 84 and 559 respectively. Since the Maximum TTR of MMC/Parallel is too large, we show the difference between other five algorithms in Figure 3.21(b). We find that the RPS algorithm has no advantage on the average TTR but has significant improvement on the maximum TTR. Its maximum TTR is almost a third of others.

\(R_i \neq R_j\): In this case, users are equipped with different numbers of radios. We let \(R_i = 3\) and \(R_j = 4\). Figure 3.22 shows the comparison of the average TTR and the maximum TTR between the six different algorithms. For example, when there are 50 channels, the average TTR of RPS, JS/Independent, JS/Parallel, Random, MMC/Independent and MMC/Parallel are 4.28, 5.38, 4.64, 4.23, 4.59 and 4.98 re-
spectively. However, the maximum TTR are 62, 103, 70, 34, 78 and 263 respectively. Same as the scenario $R_i = R_j$, Figure 3.22(b) shows the difference between five of the algorithms. The RPS algorithm has no advantage on the average TTR but has significant improvement on the maximum TTR. Another important point is that JS/Parallel has the worst performance when $R_i \neq R_j$, which is consistent with the theoretical result.

**Under the Asymmetric Model**

Large $\theta$: Firstly we study the scenario when $\theta$ is large, that is, most channels are available to users. We let $\theta = 0.8$, $G = 0.6Q$, $R_i = 3$ and $R_j = 4$. Figure 3.23 shows the comparison of the average TTR between eight different algorithms. For example, when there are 50 channels, the average TTR of RPS, JS/Independent, JS/Parallel, Random, MMC/Independent, MMC/Parallel, HH/Independent and H-H/Parallel are 4.68, 4.61, 5.69, 5.00, 4.77, 4.80, 3.33 and 3.80 respectively. However, the maximum TTR are 69, 96, 112, 80, 81, 1698, 197 and 155 respectively. HH algorithm with multiple radios has distinct advantage on the average TTR. However, both MMC and HH has very large Maximum TTR. The reason is that their guaranteed rendezvous is based on some special conditions. Therefore, in terms of
the average TTR, the generalized HH algorithm with random replacement is the best; in term of the maximum TTR, the RPS algorithm is the best when θ is large.

Small θ: Then we study the scenario when θ is small, that is, only a small part of channels are available to users. We let $\theta = 0.4$, $G = 0.4Q$, $R_i = 3$ and $R_j = 4$. For example, when there are 50 channels, the average TTR of RPS, JS/Independent, JS/Parallel, Random, MMC/Independent, MMC/Parallel, HH/Independent and HH/Parallel are 2.21, 2.24, 2.53, 2.18, 2.24, 4.57, 1.16 and 1.20 while the maximum TTR are 22, 65, 47, 28, 30, 338, 53 and 52 respectively. Figure 3.24 shows the comparison between the seven different algorithms (except MMC/Parallel). It has the same result with the scenario when θ is large.

From Figure 3.21—Figure 3.24, we observe that both the average TTR and the maximum TTR of all the algorithms increase with the increase of the number of all channels because it will take more time to identify a commonly available channel among more channels. In addition, we observe a tradeoff between the average TTR and the maximum TTR. If the maximum TTR is more important to an application (e.g., the one with QoS requirements), the RPS algorithm is the best choice because it gives the smallest maximum TTR under both the symmetric and asymmetric
models. On the other hand, if the average TTR is more important to an application, the RPS algorithm and the generalized Jump-Stay algorithm with independent sequence are the best choices under the symmetric model while the generalized HH algorithm is the best choice under the asymmetric model.

### 3.4 Chapter Summary

We investigated a new approach (using multiple radios per user) to significantly speeding up the rendezvous process in cognitive radio networks, generalized the Random algorithm and the existing algorithms in order to use multiple radios for faster rendezvous, and designed a new rendezvous algorithm (called role-based parallel sequence (RPS)) to specifically exploit multiple radios for fast rendezvous. We theoretically derived the upper bounds of E(TTR) and MTTR of these algorithms, and conducted extensive simulation studies to evaluate their performance. We observed the following properties:

- Multiple radios can significantly speed up rendezvous, especially when the number of radios per user is increased from a small value. For example, when there are 50 channels and the number of radios per user is increased from 1 to
2, the average TTR of the generalized Jump-Stay algorithm with Independent Sequence is reduced by 63.7% while its maximum TTR is reduced by 75.6% under the symmetric model.

• Given a fixed number of radios for two users, the rendezvous performance would be better if the radios are evenly allocated on the two users. For example, if there are 6 radios, the allocation (3, 3) achieves rendezvous faster than the allocation (4, 2) or (5, 1).

• In terms of the average TTR, the generalized HH algorithm performs better than other algorithms in our simulation. In terms of the maximum TTR, the RPS algorithm is the best.
Chapter 4

Efficient Channel Hopping Rendezvous Algorithm Based on Available Channel Set

In this chapter, we design a new rendezvous algorithm, called Interleaved Sequences based on Available Channel set (ISAC), that attempts rendezvous on the available channels only for faster rendezvous. Specifically, we present the system model and problem definition in Section 4.1, design the ISAC algorithm in Section 4.2 and theoretically analyze its properties in Section 4.3, present simulation results for performance evaluation in Section 4.4 and conclude this work in Section 4.5.

4.1 System Model and Problem Definition

We consider a CRN consisting of $K$ ($K \geq 2$) users. The licensed spectrum is divided into $Q$ ($Q \geq 1$) non-overlapping channels $c_1, c_2, ..., c_Q$, where $c_i$ is called channel $i$ ($i = 1, 2, ..., Q$). Let $C$ be the whole channel set $\{c_1, c_2, ..., c_Q\}$. Let $C^i \subset C$ be the set of available channels of user $i$ ($i = 1, 2, ..., K$), where a channel is said to be available to a user if the user can communicate on this channel without causing interference to the licensed users. The available channels can be identified by any spectrum sensing method (e.g., [9]).

Time is divided into slots of equal duration. Time synchronization among the users is not needed (i.e., different users may start their time slots at different time) as long as the handshaking messages for rendezvous can be exchanged in the overlapped interval of two time slots of two respective users [25].

Rendezvous Problem for two users: Consider any two users who may have different channel availability and may start the rendezvous operation at different time. The problem is to determine a CH sequence for each user, such that these users will hop on a commonly-available channel in the same time slot within finite time.
Rendezvous for multiple users in multi-hop networks: Suppose there are multiple users. They may have different channel availability, they may start the rendezvous operation at different time, or they may be in a multi-hop network. These users can achieve rendezvous as follows: iteratively solve the above rendezvous problem to achieve pairwise rendezvous for two users in order to achieve global rendezvous for all users [25, 26]. In this process, the crucial issue is to solve the above rendezvous problem for two users. In the following, we focus on solving the rendezvous problem for two users.

4.2 Algorithm Design

Consider any two users (say, users 1 and 2). The user who initiates communication assumes the role of sender while the other user assumes the role of receiver. For the sender (say, user 1), let there be \( m = |C^1| \) available channels and the available channel set be \( C^1 = \{C_1^1, C_2^1, C_3^1, ..., C_m^1\} \). For the receiver (user 2), let there be \( n = |C^2| \) available channels and the available channel set be \( C^2 = \{C_1^2, C_2^2, C_3^2, ..., C_n^2\} \). Under the symmetric model, \( C^1 \) is same with \( C^2 \); under the asymmetric model, \( C^1 \) may not be same with \( C^2 \). Let \( G \) be the number of channels commonly available to the sender and the receiver (i.e., \( G \) is equal to the number of elements in \( C^1 \cap C^2 \)).

Using ISAC, the sender generates a CH sequence based on the available channel set \( C^1 \) while the receiver generates another CH sequence based on the available channel set \( C^2 \). We describe how ISAC generates these CH sequences in the following.

**Sender-role Sequence:** Given \( C^1 = \{C_1^1, C_2^1, ..., C_m^1\}, m_p \) be the smallest prime number which is not smaller than \( m \). We first expand \( C^1 \) to get a total of \( m_p \) channels and represent these channels in the following multiset (i.e., a set in which the elements may appear more than once [20]):

\[
C^1_* = \{C_1^1, C_2^1, ..., C_m^1, C_{m+1}^1, C_{m+2}^1, ..., C_{m_p}^1\}
\]

where \( C_h^1 (h = m + 1, ..., m_p) \) are randomly selected from \( \{C_1^1, C_2^1, ..., C_m^1\} \). The sender-role sequence is generated in rounds and each round contains \( m_p \) time slots.
A starting channel is randomly selected from $C_1$. Then, sender moves to the next channel index in $C_1$ and keeps hopping on $m_p$ channels in the round-robin fashion. Figure shows the structure of sender-role sequence. Algorithm 5 shows the detailed steps for generating a sender-role sequence. For example, in Figure 4.1(b), $m = 2$, $m_p = 2$, $C_1 = \{1, 2\}$ and $C_1^r = \{1, 2\}$. When starting channel is channel 2, the CH sequence is $\{2, 1, 2, 1, \ldots\}$.

**Receiver-role Sequence:** Given $C^2 = \{C^2_1, C^2_2, \ldots, C^2_n\}$, it randomly generates a permutation of $C^2 : \{C^2_{l_1}, C^2_{l_2}, \ldots, C^2_{l_n}\}$ first. The receiver-role sequence is composed of two interleaving subsequences: Odd Subsequences (OS) and Even Subsequences (ES). That is, the channels in time slot 1, 3, 5, ..., $(2n - 1)$ follows OS and the channels in time slot 2, 4, 6, ..., $(2n)$ follows ES. Figure 4.2(a) shows the structure of odd subsequence of receiver-role. $OS_i$ means the $i$-th time slot in OS. OS is generated in rounds and each round contains $n$ time slots where $n$ is the number of channels in the available channel set of the receiver. Receiver keeps hopping on $n$ channels with the order $\{C^2_{l_1}, C^2_{l_2}, \ldots, C^2_{l_n}\}$ in round-robin fashion alternately. Figure 4.2(b) shows the structure of Even Subsequence of receiver-role. ES is generated in rounds and each round contains $n$ time slots. $ES_i$ means the $i$-th time slot in ES. The sequence in the first round is $\{C^2_{l_1}, C^2_{l_2}, \ldots, C^2_{l_n}\}$. In the next round, all channel indices are left-shifted by 1 to generate a new permutation of $C^2$. That is $\{C^2_{l_2}, C^2_{l_1}, \ldots, C^2_{l_n}, C^2_{l_1}\}$. ES in the remaining rounds are determined in the same way. Then the two sub-sequences interleave with each other. All channels in the odd time slots of CH
sequence follow Odd Subsequence (OS) while all channels in the even time slots of CH sequence follow Even Subsequence (ES). Figure 4.2(c) shows the interleaved final CH sequence of receiver-role. The last column means the channel in time slot $T$. If $T$ is odd, then the channel should be $OS_{T/2+1}$. If $T$ is even, then the channel should be $ES_{T/2}$. Figure 4.3(a) and Figure 4.3(b) show the OS and ES when $C^2 = \{1, 3, 4\}$ and $\{C^2_{i_1}, C^2_{i_2}, \ldots, C^2_{i_n}\} = \{3, 4, 1\}$. Figure 4.3(c) shows the interleaved CH sequence. The channels with underline are from OS. Others are from ES.

The CH sequence of sender and the OS of receiver can guarantee the rendezvous when $n = km_p$ ($k \geq 1$). The CH sequence of sender and the ES of receiver can guarantee the rendezvous when $n \neq km_p$ ($k \geq 1$). The ISAC algorithm is formally presented as Algorithm 5 and Algorithm 6.

In the algorithm for sender, line 3 randomly selects a starting channel. Line 7 calculates the corresponding channel in time slot $t$ in round-robin fashion. In the algorithm for receiver, line 5 indicates whether the current time slot is odd time slot or not. Line 6 calculates the corresponding channel in time slot $t$ if $t$ is an odd. Line
Algorithm 5: ISAC algorithm (for generating sender-role sequence)

Require: sender’s available channel set $C^1 = \{C^1_1, C^1_2, ..., C^1_m\}$

1: $m_p =$ the smallest prime number not smaller than $m$
2: Randomly select $m_p - m$ channels from $C^1$ and denote these channels by $\{C^1_{m+1}, C^1_{m+2}, ..., C^1_{m_p}\}$
3: Randomly select an integer $k$ in $[1, m_p]$
4: $t = 0$
5: while not rendezvous do
6: $t = t + 1$
7: $T = (t - 2 + k) \% m_p + 1$
8: Attempt rendezvous on channel $C^1_T$
9: end while

Figure 4.3: Receiver-role Sequence when $C^2 = \{1, 3, 4\}$. 
Algorithm 6: ISAC algorithm (for generating receiver-role sequence)

Require: receiver’s available channel set $C^2 = \{C^2_1, C^2_2, ..., C^2_n\}$

1: Randomly generate a permutation of $C^2 : \{C^2_{l_1}, C^2_{l_2}, ..., C^2_{l_n}\}$

2: $t = 0$

3: while not rendezvous do

4: $t = t + 1$

5: if $t \% 2 == 1$ then

6: $T = (\frac{t}{2}) \% n + 1$ //OS

7: else

8: $T = ((\frac{t-1}{2n}) \% n + (\frac{t}{2}) \% n - 1) \% n + 1$ //ES

9: end if

10: Attempt rendezvous on channel $C^2_{l_T}$

11: end while

8 calculates the corresponding channel in time slot $t$ when $t$ is an even.

An illustration example: Available channel sets of sender and receiver are $C^1 = \{1, 2\}$ and $C^2 = \{1, 3, 4\}$, respectively. That is, $m_p = m = 2$ and $n = 3$. Figure 4.4 shows the rendezvous of the two users. Notice that time-synchronization is not available. Since the sender has only two available channels, the offset between the two users’ sequences could be 0 or 1 (as shown in Figure 4.4). Sender and receiver can achieve rendezvous regardless of the offset between their sequences (see the time slots in gray color in Figure 4.4).

4.3 Algorithm Analysis

The ISAC algorithm has an important feature that distinguishes itself from the existing algorithms using the available channel set: it guarantees that rendezvous can be completed within finite time (see Table 1.2). In this section, we prove this feature and derive the upper bounds of MTTR. We first present the following two lemmas.
Figure 4.4: Rendezvous of two users performing ISAC when $C^1 = \{1, 2\}$ and $C^2 = \{1, 3, 4\}$.

**Lemma 1.** Given two positive integers $x$ and $y$, and $y$ is a prime number. The statement “$x$ is not evenly divisible by $y$” is equivalent to that “$x$ and $y$ are co-prime”.

**Proof.** $\implies$ Suppose $z$ is a common factor of $x$ and $y$. Since $y$ is a prime number, $z$ can be only 1 or $y$ itself. However, if $z$ is identical to $y$, then $x$ must be evenly divisible by $y$, leading to a contradiction. Thus, $x$ and $y$ have no other common factors rather than 1, i.e., they are co-prime.

$\iff$ Suppose that $x$ is evenly divisible by $y$, i.e., there exists $z$ such that $x = zy$. Since $y$ is a prime number, $y$ is not equal to 1. Thus, 1 and $y$ are both common factors of $x$ and $y$, i.e., $x$ and $y$ are not co-prime, leading to a contradiction. So, $x$ cannot be evenly divided by $y$. □

**Lemma 2.** In the sender-role sequence that is generate by Algorithm 5, any consecutive $m_p$ channel indices in odd time slots (e.g., channel indices in time slots 1, 3, 5, ...) form a permutation of the channel indices in $C^1_1 = \{C^1_1, C^1_2, ..., C^1_m, C^1_{m+1}, C^1_{m+2}, \ldots, C^1_{m_p}\}$.

This statement also holds for the channel indices in even time slots.

**Proof.** Suppose the user has $m$ available channels and the $m$ channels $C^1_m$ has been expanded to $C^1_{m_p}$. The starting channel index is $k$. Then the channel index in the
time slot $T$ can be denoted as $(T - 1 + k)\%m_p$. Any consecutive $m_p$ channel indices in odd time slots are $O_1\%m_p, (O_1 + 2)\%m_p, \ldots, (O_1 + 2m_p)\%m_p$ if the first channel of these channels is $C_{O_1}$. To prove that this subsequence is a permutation of $m_p$ channel index, we refer to Lemma 1 in [5], “Given a positive integer $P$, if integer $r \in [1, P)$ is relatively prime to $P$ (i.e., the common factor between them is 1), then for any integer $x \in [0, P)$ the sequence $S = <(x \mod P) + 1, ((x + r) \mod P) + 1, \ldots, ((x + (P - 1)r) \mod P) + 1>$ is a permutation of $<1, 2, ..., P>$”. In our analysis, $r = 2$ is relatively prime to $P = m_p$. So $O_1 + 0\%m_p, (O_1 + 2)\%m_p, \ldots, (O_1 + 2(m_p - 1))\%m_p$ is a permutation of $<1, 2, ..., m_p>$.

Based on Lemma 1 and Lemma 2, we prove the correctness of ISAC and derive its upper-bound on MTTR in two cases: 1) $n$ is not divisible by $m_p$; 2) $n$ is divisible by $m_p$.

**Theorem 7.** Two users performing the ISAC algorithm can achieve rendezvous in at most $2m_p n - 2G + 2$ time slots, where $m$ and $n$ are the numbers of channels of the two users, $m_p$ is the smallest prime number which is not smaller than $m$, and $G$ is the number of commonly-available channels of the two users.

**Proof.** If $n$ is not divisible by $m_p$ and $m > 2$, we consider the CH sequence of sender and the Odd Subsequence of receiver. Based on Figure 4.2 which shows the structure of sequences of ISAC and Lemma 2 we extract the channels in the odd time slots of sender-role sequence and denote this subsequence in round 1 by $S^1 = \{C^1_k, C^1_{k+1}, \ldots, C^1_{m_p}, C^1_1, C^1_2, \ldots, C^1_{k-1}\}$ (with $m_p$ items). The receiver’s OS in round 1 is $S^2 = \{C^2_{l_1}, C^2_{l_2}, \ldots, C^2_{l_n}\}$ (with $n$ items). Furthermore, we use $S^1_h (S^2_h)$ to denote the $h$-th item of $S^1 (S^2) (h = 1, 2, \ldots)$. Then, according to ISAC, in time slot $(2T - 1)$ sender and receiver actually hop on channel $S^1_{((T-1)\%(m_p)+1}$ and channel $S^2_{((T-1)\%(n)+1}$, respectively. Since there is no time-synchronization, without loss of generality, we investigate rendezvous of the two users at a starting point in which sender and receiver are in time slots $t^1_0$ and $t^2_0$, respectively. Next, we prove that, for any pair of channels $S^1_h \in S^1$ and $S^2_g \in S^2$, there must be a same time slot at which sender and receiver hop on $S^1_h$ and $S^2_g$, respectively. Notice that this result implies
the guaranteed rendezvous of the two users since they have commonly-available channels (i.e., $S^1$ and $S^2$ share common items). To achieve this, we should prove that, for any $1 \leq h \leq m_p$ and $1 \leq g \leq n$, there exists $t$ to satisfy the following equations (i.e., sender hops on channel $S^1_h$ in its time slot $t + t^1_0$ and receiver hops on channel $S^2_g$ in its time slot $t + t^2_0$)

$$h - 1 \equiv t + t^1_0 - 1 \mod (m_p) \quad (4.3.1)$$
$$g - 1 \equiv t + t^2_0 - 1 \mod (n) \quad (4.3.2)$$

which can be equivalently rewritten as follows

$$t \equiv (h - t^1_0) \mod (m_p) \quad (4.3.3)$$
$$t \equiv (g - t^2_0) \mod (n) \quad (4.3.4)$$

Since $m_p$ is a prime number and $n$ is not evenly divisible by $m_p$, from Lemma 1, we know $m_p$ and $n$ are co-prime. Therefore, from the Chinese Remainder Theorem, there exists an integer $t$ that solves Equations (4.3.3) and (4.3.4). As mentioned earlier, this result implies the guaranteed rendezvous of the two users. We further derive an upper bound of TTR as follows. Notice that there are $m_p n$ combinations of $h$ and $g$ values (i.e., $m_p n$ channel pairs of $S^1_h$ and $S^2_g$). Hence, $t$ does not exceed $m_p n$ regardless of $h$ and $g$ values. On the other hand, since the two users have $G$ commonly-available channels (i.e., $S^1$ and $S^2$ share $G$ common items), there are $G$ pairs of commonly-available channels among all the $m_p n$ pairs of $S^1_h$ and $S^2_g$. Therefore, rendezvous will occur in at most $m_p n - G + 1$ time slots in the Odd Subsequence and the corresponded time slot in the whole CH sequence is $2m_p n - 2G + 1$. In other words, $TTR \leq 2m_p n - 2G + 1$.

Then we illustrate the case when $m = 1$ and $m = 2$. If $m = 1$, the only channel of sender must be a commonly-available channel of the two users. The sender stay on the only channel while the receiver will hop on this channel in at most $2n - 1$ time slots. If $m = 2$, all channels in the odd time slots of sender are one channel and all channels in the even time slots are another channel. No matter which one is a commonly-available channel, the receiver will hop on this channel in at most
\[ n - G + 1 \text{ time slots in its OS or ES while the corresponded time slot in the whole CH sequence is } 2n - 2G + 2. \]

We consider the next case that \( n \) is divisible by \( m_p \).

If \( n \) is divisible by \( m_p \), we consider CH of sender and the Even Subsequence of receiver. Then, we investigate any consecutive time-span of \( n \times m_p \) time slots from a starting point in which sender and receiver are in time slots \( t_0^1 \) and \( t_0^2 \).

We prove that, any pair of channels \( C_{1h}^1 \) in \( \{C_{11}^1, C_{12}^1, ..., C_{m1}^1, C_{m+1}^1, ..., C_{mp}^1\} \) and \( C_{2g}^1 \) in \( \{C_{l1}^2, C_{l2}^2, ..., C_{ln}^2\} \) appears once and only once in the time-span. That is, there exists exactly one \( t \) (\( 0 < t \leq n \times m_p \)) such that sender hops on channel \( C_{1h}^1 \) in its time slot \( t_0^1 + t \) and receiver hops on channel \( C_{2g}^2 \) in its time slot \( t_0^2 + t \). Next we prove it by contradiction.

Suppose that a pair of channels, say \( C_{1h}^1 \) and \( C_{2g}^2 \) appears more than one time. This implies that there exist two different \( t_1 \) and \( t_2 \) such that:

- sender hops on channel \( C_{1h}^1 \) in time slot \( t_0^1 + t_1 \),
- receiver hops on channel \( C_{2g}^2 \) in time slot \( t_0^2 + t_1 \),

and

- sender hops on channel \( C_{1h}^1 \) in time slot \( t_0^1 + t_2 \),
- receiver hops on channel \( C_{2g}^2 \) in time slot \( t_0^2 + t_2 \),

According to our ISAC algorithm and Lemma 2, we must have (suppose that \( t_2 > t_1 \) for convenience):

\[ t_2 = t_1 + \mu \times m_p \quad (4.3.5) \]

Sender-role hops on the \( m_p \) channels in round-robin fashion. The channel will appear after each \( m_p \) time slots. Each channel of sender will appears for \( n \) times in any continuous \( n \times m_p \) time slots. There are at most another \( n - 1 \) rounds after the first time this channel appears, so \( 1 \leq \mu < n \).

\[ t_2 = t_1 + a \times (n - 1) + b \times n \quad (4.3.6) \]
In each round of receiver’s ES, all channel indices are left-shifted by 1 from the previous round. The channel will appear after each \((n - 1)\) time slots. However, when the channel is the first one in current round, it will appear after \((n - 1) + n\) time slots (the third and fourth channel 1 in Figure 4.3), so \(b \in \{0, 1\}\). Each channel of sender will appears for \(m_p\) times in any consecutive \(n \times m_p\) time slots. There are at most another \(n - 1\) rounds after the first time this channel appears, so \(1 \leq a < m_p\).

Using Equations 4.3.5 and 4.3.6 and the fact that \(n = \delta \times m_p\) \((n\) is evenly divisible by \(m_p\)), we should have

\[
\begin{align*}
\mu \times m_p &= a \times (\delta \times m_p - 1) + b \times \delta \times m_p \quad (4.3.7) \\
a &= ((a + b) \times \delta - \mu) \times m_p \quad (4.3.8)
\end{align*}
\]

That is, \(a\) should be divisible by \(m_p\), which contradicts the requirement \(1 \leq a < m_p\). So any pair of channels \(C_1^h\) and \(C_2^g\) in any consecutive time-span of \(n \times m_p\) time slots appears once and only once.

This result implies the guaranteed rendezvous of the two users. Notice that there are \(m_p \times n\) combinations of \(h\) and \(g\) values \((i.e., m_p n\) channel pairs of \(S^1_h\) and \(S^2_g\)) in at most \(m_p n\) time slots. Hence, the combinations in \(m_p n\) time slots are different to each other. We further derive an upper bound of TTR as follows. If there are \(G\) commonly-available channels of the two users, there are \(G\) pairs of commonly-available channels among all the \(m_p n\) pairs of \(S^1_h\) and \(S^2_g\). The worst case is that the whole \(G\) times rendezvous occur in the last \(G\) time slots of channel hopping sequence. Therefore, rendezvous will occur in at most \(m_p n - G + 1\) time slots in the Even Subsequence and the corresponded time slot in the whole CH sequence is \(2m_p n - 2G + 2\). In other words, \(TTR \leq 2m_p n - 2G + 2\).

To sum up, two users performing the ISAC algorithm can achieve rendezvous in at most \((2m_p n - 2G + 2)\) times lots, where \(m\) and \(n\) are the numbers of channels of the two users, \(m_p\) is the smallest prime number which is not smaller than \(m\), \(G\) is the number of commonly-available channels of the two users.
4.4 Simulation

We built a simulator in Visual Studio 2010 to evaluate the performance of our proposed ISAC algorithm. We consider the following algorithms for comparison: i) MMC [35] (MMC generates CH sequences based on available channel set), ii) Jump-Stay [25] and iii) CRSEQ [33] (Among existing algorithms, Jump-Stay and CRSEQ have been shown to have good performance [26]). We remind that Jump-Stay and CRSEQ apply a random replacement operation to randomly replace the unavailable channels in the CH sequences by the available ones (see discussion in Section 2.1.5). We introduce a parameter $\theta$ ($0 < \theta \leq 1$) to control the ratio of the number of available channels to the total number of channels $Q$. Available channels are randomly selected from the whole channel set such that the average number of available channels is equal to $\theta Q$. Besides $\theta$, we are concerned with parameter $G$, i.e., the number of commonly-available channels of the two users involved in the rendezvous. For each $\theta$, we let $G$ properly vary in $[1, \theta Q]$. For each combination of parameter values, we perform 500,000 independent runs and compute the average TTR, the maximum TTR (MTTR) and the variance of TTR accordingly. The variance of TTR can reveal the stability of performance of rendezvous algorithms. Notice that, $n$ is the number of available channels of receiver, and $m_p$ is the smallest prime number which is not smaller than the number of available channels of sender.

4.4.1 Effectiveness of ISAC

In this subsection, we demonstrate that ISAC can effectively improve the rendezvous performance. We consider three scenarios with different number of available channels: 1) $\theta$ is small and equal to 0.1 (i.e., ratio of the number of available channels to the total number of channels is 0.1), ii) $\theta$ is moderate and equal to 0.4, and iii) $\theta$ is large and equal to 0.8.

- *Small $\theta$:* Firstly we study the scenario when $\theta$ is small, that is, only a small part of channels are available to users. We set $\theta = 0.1$, $G = 1$. Figure 4.5 shows
the average TTRs, the maximum TTRs and the variances of TTR of different algorithms against $Q$. Since there is only one commonly-available channel ($G = 1$), such case is believed to be hard for quick rendezvous. According to Figure 4.5, our ISAC algorithm significantly outperforms other algorithms in terms of both average TTR and maximum TTR. For example, when there are 50 channels, the average TTRs of ISAC, MMC, Jump-Stay and CRSEQ are 14.65, 27.00, 21.26 and 21.20 while the maximum TTRs are 98, 1023, 562 and 530, respectively. We find that ISAC has significant improvement on the maximum TTR. Its maximum TTR is almost one tenth of others. In this example, $m \leq 0.15 \times 50 = 7.5$. According to Theorem 7, $MTTR \leq 2 \times m_p n - 2G + 2 = 2 \times 7 \times 7 - 2 + 2 = 98$. This result shows that the upper bound given by Theorem 7 is tight when $G = 1$. The variances of TTR of ISAC, MMC, Jump-Stay and CRSEQ in this case are 255.06, 2080.64, 738.08 and 526.13, respectively. The performance of ISAC is the most stable one.

- Moderate $\theta$: Then we study the scenario when $\theta$ is moderate, that is, there are a moderate part of channels are available to users. We set $\theta = 0.4$, $G = 0.25\theta Q$. That is, there are 40% channels are available to the users and 25% among them are commonly-available to both of the users. Figure 4.6 shows the average TTRs, the maximum TTRs and the variances of TTR of different algorithms against $Q$. There is no large gap between different algorithms in terms of average TTR. For example, when there are 50 channels, the average TTRs of ISAC, MMC, Jump-Stay and CRSEQ are 75.94, 90.37, 83.09 and 91.20, respectively. However, ISAC significantly outperforms all the existing algorithms in terms of the maximum TTR. When there are 50 channels, the maximum TTRs are 758, 3482, 1121 and 1203, respectively. Its maximum TTR is almost one fifth of MMC. In this example, $m \leq 0.45 \times 50 = 22.5$, $G = 0.1 \times 50 = 5$. According to Theorem 7, $MTTR \leq 2 \times m_p n - 2G + 2 = 2 \times 23 \times 22 - 2 \times 5 + 2 = 1004$. This result shows that the upper bound given by Theorem 7 is not tight when $G$ is large under the asymmetric model.
Figure 4.5: Comparison of different algorithms under the asymmetric model when $\theta = 0.1$ and $G = 1$. 
Figure 4.6: Comparison of different algorithms under the asymmetric model when \( \theta = 0.4 \) and \( G = 0.25\theta Q \).

The variances of TTR of ISAC, MMC, Jump-Stay and CRSEQ in this case are 5175.16, 16475.76, 7156.27 and 9463.17, respectively. The performance of ISAC is the most stable one.

- Large \( \theta \): We now study the scenario when \( \theta \) is large, that is, most channels are available to users. We set \( \theta = 0.8, G = 0.75\theta Q \). That is, there are 80% channels are available to the users and 75% of them are commonly-available to both of the users. Figure 4.7 shows the average TTRs, the maximum TTRs and the variances of TTR of different algorithms against \( Q \). The performance is
similar to the scenario when \( \theta = 0.4 \). For example, when there are 50 channels, the average TTRs of ISAC, MMC, Jump-Stay and CRSEQ are 59.53, 57.46, 53.76 and 87.60 while the maximum TTRs are 654, 10190, 993 and 1747, respectively. Maximum TTR of ISAC is almost one twentieth of MMC. The variances of TTR of ISAC, MMC, Jump-Stay and CRSEQ in this case are 2723.60, 11594.98, 2682.32 and 7363.49, respectively. The performance of ISAC and Jump-Stay are more stable than other two algorithms.
4.4.2 Influence of $m$ or $n$ for fixed $Q$

Jump-Stay and CRSEQ generate CH sequence based on the whole channel set while Random and ISAC generate CH sequence based on the available channel set only. When we fix the total number of channels $Q$, we will get the same average TTR and maximum TTR for different $(m, n)$ if we apply Jump-Stay or CRSEQ. However, we will get different results if we apply ISAC or MMC. Figure 4.8 shows both the average TTRs and maximum TTRs of different algorithms against $m$ (or $n$) when we fix $Q = 60$ and $G = m$. We adopt eight values which are $m = 0.1Q, 0.2Q, ..., 0.8Q$. According to Figure 4.8, all algorithms get very different results when the number of available channels increases but the number of all channels is fixed. This is because there is a random-replacement in Jump-Stay and CRSEQ. When the channel in CH sequence is not available to the user, it will randomly select an available channel to replace. The chance to achieve rendezvous successfully in such time slots will increase significantly when the number of available channels is large. In terms of our ISAC algorithm, the eight simulated results are 12, 24, 36, 57, 60, 72, 84 and 92, respectively. According to Theorem 7, the eight theoretical results are 13, 25, 37, 57, 61, 73, 85 and 105, respectively. These results show that the upper bound given by Theorem 7 is tight.

4.4.3 Influence of $G$ for fixed $m$

In this subsection, we study another basic property of the proposed approach: how does the number of commonly-available channels of the two users affect the rendezvous performance? We let $\theta = 0.5$ and take five values of $G$ which are $G = 0.2\theta Q$, $G = 0.4\theta Q$, $G = 0.6\theta Q$, $G = 0.8\theta Q$, $G = \theta Q$. Figure 4.9 shows the results when we apply the ISAC algorithm. When there are 60 channels, the average TTRs of five scenarios are 157.79, 84.76, 58.03, 42.82 and 29.91 while the maximum TTRs are 1747, 891, 659, 415 and 60, respectively. In this example, $Q = 60$, $m \leq 0.55 \times 60 = 33$, $m_p = 37$. According to Theorem 7, however, the upper bound of MTTR in five
Figure 4.8: Influence of $m$ on rendezvous performance when $Q = 60$. 
scenarios are 2428, 2416, 2404, 2392 and 73, respectively. We note that our upper bound under the symmetric model is tight while the upper bound under the asymmetric model is less tight especially when $G$ is large.

4.5 Chapter Summary

We designed a new rendezvous algorithm, called ISAC, for cognitive radio networks. ISAC is the first one in the literature that generates CH sequences based on the available channel set instead of the whole channel set (the available channel set is subset of the whole channel set) while providing guaranteed rendezvous. We proved that ISAC provides guaranteed rendezvous and derived upper bounds on the maximum TTR (MTTR) under the asymmetric model. These upper bounds are expressed in terms of the number of available channels instead of the total number of channels ($2m_p n - 2G + 2$). We conducted extensive computer simulation for performance evaluation and observed the following properties:

- In terms of the maximum TTR, ISAC significantly outperforms other algorithms in almost all scenarios. For example, ISAC reduces the maximum TTR up to 97% when $Q = 100$ and $G = 10$ in Figure 4.5(b). ISAC is suitable to
many QoS-concern applications of CRNs which desire small MTTR.

- In terms of the average TTR, performance of ISAC is better than those of other algorithms when the available channel set is a small subset of the whole channel set (e.g., it reduces the average TTR up to 45.8% in Figure 4.5(a)).

- In terms of the variance of TTR, ISAC is the most stable algorithm among all the rendezvous algorithms in our simulation.

- The upper bound on MTTR under the asymmetric model is not tight especially when $G$ is large (e.g., greater than $0.2\theta Q$ in Figure 4.9).
Chapter 5

Adjustable Rendezvous in Multi-Radio Cognitive Radio Networks

We exploited multiple radios in Chapter 3 and available channel set in Chapter 4 for faster rendezvous. In this chapter, we exploit both ideas (i.e., multiple radios based on available channel sets) for more effective rendezvous and design an Adjustable Multi-Radio Rendezvous (AMRR) algorithm. It combines the desirable features in previous two chapters. Suppose that a cognitive user is equipped with $R$ radios. Our basic idea is to partition the radios into two groups: $k$ stay radios and $(R - k)$ hopping radios. The user stays on specific channels in the stay radios while hops on its available channels parallelly in the hopping radios. Specifically, we present our rendezvous algorithm AMRR in Section 5.1. In Section 5.2, we analyze the upper-bounds of MTTR and E(TTR) of AMRR. Simulation results are given in Section 5.3. We conclude this work in Section 5.4.

5.1 Adjustable Multi-Radio Rendezvous Algorithm

We focus on rendezvous of two users in multi-radio CRNs. The rendezvous of multiple users could be achieved by pairwise rendezvous of two users [1]. Time is divided into slots of equal duration. The licensed spectrum is divided into $|C|$ non-overlapping channels $C^1, C^2, ..., C^{|C|}$, where $C^i$ is called channel $i$. Let $C$ be the whole channel set $\{C_1, C_2, ..., C_{|C|}\}$. Let $C_1 = \{C^1_1, C^2_1, ..., C^{|C_1|}_1\} \subseteq C$ and $C_2 = \{C^1_2, C^2_2, ..., C^{|C_2|}_2\} \subseteq C$ be the sets of available channels of user 1 and user 2, respectively, where $|C_1|$ and $|C_2|$ are the numbers of available channels of user 1 and user 2, respectively. A channel is said to be available to a user if the user can communicate on this channel without causing interference to any PUs. User 1 is
equipped with \( R_1 \) \((R_1 > 1)\) radios and user 2 is equipped with \( R_2 \) \((R_2 > 1)\) radios. Note that \( R_1 \) may not be equal to \( R_2 \) in multi-radio CRNs. Let \( G \) be the number of commonly-available channels of user 1 and user 2. The CH sequence of user 1 is denoted by \( \{S_1^1, S_2^1, S_3^1, \ldots\} \), where vector \( S_t^1 = \{S_{t1}^1, S_{t2}^1, S_{t3}^1, \ldots, S_{tR_1}^1\} \) represents that user 1 hops on channel \( S_{tj}^1 \) in radio \( j \) in time slot \( t \). We define the rendezvous problem as follows.

**Rendezvous problem in multi-radio CRNs:** Suppose two users have \( R_1 \) and \( R_2 \) \((R_1, R_2 > 1)\) radios respectively and they may start the rendezvous process at different time (i.e., time-synchronization is not needed). The problem is to determine a CH sequence for each radio of each user, such that these users will hop on a commonly-available channel in the same time slot.

In this work, we design a new rendezvous algorithm, called *Adjustable Multi-Radio Rendezvous* (AMRR), which exploits multiple radios and available channels for more efficient rendezvous. Our basic idea is to partition the radios into two groups: \( k \) stay radios and \((R - k)\) hopping radios. Users stays on specific available channels in the stay radios while hops on its available channels parallelly in the hopping radios. Our AMRR algorithm generates CH sequences in rounds. The rendezvous is expected to be achieved between the hopping radios of one user and the stay radios of the other. The upper-bounds of MTTR and E(TTR) of AMRR are the expressions of the length of a round and the number of rounds needed to guarantee rendezvous (later shown in Theorems 8 and 9), which are related to the number of available channels of two users (i.e., \(|C^1|\) and \(|C^2|\)). Our AMRR is adjustable in giving its best performance on either MTTR or E(TTR) by adjusting value of \( k \) (i.e., the number of stay radios). This is why we call the new algorithm *Adjustable Multi-Radio Rendezvous* (AMRR). AMRR works as follows.

1) All radios are divided into two groups, \( k \) stay radios \((1 \leq k < R)\) and \((R - k)\) hopping radios.

2) A starting index \( i \) is randomly selected from \([1, |C^u|]\) \((u = 1, 2)\).

3) \((R - k)\) hopping radios parallely hop on \(|C^u|\) channels in the round-robin
(a) 3 radios and \( k = 1 \)

(b) 3 radios, \( k = 1 \) and \( C^u = \{1, 3, 4, 5, 7\} \)

(c) 3 radios and \( k = 2 \)

(d) 3 radios, \( k = 2 \) and \( C^u = \{1, 3, 4, 5, 7\} \)

Figure 5.1: CH sequence of a user with 3 radios.
manner. Figure 5.1(a) shows the sequence of a user with \( R = 3 \) radios and \(|C_u|\) available channels when \( k = 1 \). The starting index is randomly selected as \( C_u^2 \). For example, in Figure 5.1(b) \( C_u^u = \{1, 3, 4, 5, 7\}, |C_u| = 5 \). The starting index \( i \) is \( C_u^2 = 3 \) and the \( q\)-th channel is \( C_u^{u(q-1)%|C_u|+1} \) (i.e., \( C_u^{u(q-1)%5+1} \) in this example). The CH sequence is \( \{3, 4, 5, 7, 1, 3, 4, 5, 7, 1, \ldots\} \) and two hopping radios 1 and 2 implement this sequence parallelly as follows. Radio 1 implement subsequence \( \{3, 5, 1, 4, 7, \ldots\} \) and Radio 2 implement subsequence \( \{4, 7, 3, 5, 1, \ldots\} \). Radios 1 and 2 have a permutation of all channels in any \( \lceil \frac{|C_u|}{3-1} \rceil = 3 \) continuous time slots. Figure 5.1(c) and Figure 5.1(d) show the sequences when \( k = 2 \). Radio 1 has a permutation of all channels in any \( \lceil \frac{|C_u|}{3-2} \rceil = 5 \) consecutive time slots.

4) \( k \) stay radios stay on \( k \) channels for \( \lceil \frac{|C_u|}{R-k} \rceil \) time slots and switch to next \( k \) channels for the same duration, where the \( k \) channels are taken from \([C_1^u, C_{|C_u|}]\) in the round-robin manner. For example, in Figure 5.1(b) Radio 3 stays on channel 1 for \( \lceil \frac{5}{3-1} \rceil = 3 \) time slots and then switches to channel 3. Figure 5.1(c) and Figure 5.1(d) show the sequences when \( k = 2 \). Radio 2 and Radio 3 stay on channels 1 and 3, respectively, in the first \( \lceil \frac{|C_u|}{R-2} \rceil = 5 \) time slots and then switch to channels 4 and 6, respectively, in the next 5 time slots.

The algorithm is formally presented as Algorithm 7.

In line 2, starting index \( i \) is preselected randomly. In lines 5-8, the \( (R - k) \) hopping radios hop on continuous \( (R - k) \) available channels. In line 9-12, the \( k \) stay radios switch to the next \( k \) channels after each \( \lceil \frac{|C_u|}{R-k} \rceil \) time slots.

5.2 Algorithm Analysis

In this section, we theoretically analyze the AMRR algorithm. Specifically, we derive the upper-bounds of MTTR of the AMRR algorithm under the symmetric model and the asymmetric model in Theorem 8 and Theorem 9, respectively. In Theorem 8 we also derive the upper-bound of E(TTR) of the AMRR algorithm.
Algorithm 7: AMRR Algorithm

Require: \( R_u, k, C^u \)

1: \( t = 1 \)

2: \( i = \) a number randomly selected from \([1, |C^u|]\)

3: \( \overrightarrow{S}_t^u = \{S_{t1}^u, S_{t2}^u, S_{t3}^u, \ldots, S_{tR_u}^u\} \)

4: \textbf{while} not rendezvous \textbf{do}

5: \hspace{1em} \textbf{for} \( j = 1 \) to \( R - k \) \textbf{do}

6: \hspace{2em} \( h = i + ((t - 1) \times (R - k) + j - 2)\%|C^u| + 1 \)

7: \hspace{2em} \( S_{tj}^u = C_h^u \)

8: \hspace{1em} \textbf{end for}

9: \hspace{1em} \textbf{for} \( j = R - k + 1 \) to \( R \) \textbf{do}

10: \hspace{2em} \( h = ([t |C^u|/(R-k)] \times k + (j - R + k) - 1)\%|C^u| + 1 \)

11: \hspace{2em} \( S_{tj}^u = C_h^u \)

12: \hspace{1em} \textbf{end for}

13: \( t = t + 1 \)

14: \hspace{1em} Attempt rendezvous on \( \overrightarrow{S}_t^u \) (i.e., radio \( j \) access channel \( S_{tj}^u, j = 1, 2, \ldots, R \))

15: \textbf{end while}
under the symmetric model.

**Theorem 8.** Under the symmetric model \((C^1 = C^2)\), let \(k_1 (1 \leq k_1 < R_1)\) and \(k_2 (1 \leq k_2 < R_2)\) denote the numbers of stay radios of two users. We define \(\beta = \max\{\lceil \frac{|C^1|}{R_1-k_1} \rceil, \lceil \frac{|C^2|}{R_2-k_2} \rceil \}\) and \(\gamma = \min\{\lceil \frac{|C^1|}{R_1-k_1} \rceil, \lceil \frac{|C^2|}{R_2-k_2} \rceil \}\). The MTTR and the \(E(TTR)\) of AMRR are upper-bounded by 
\[
\beta + \frac{(\beta - 1)^2}{2\gamma} = O\left(\frac{|C_1|}{\max\{R_1-k_1, R_2-k_2\}}\right)
\]
and 
\[
2 \beta - 1 = O\left(\frac{|C_1|}{\max\{R_1-k_1, R_2-k_2\}}\right),
\]
respectively, which are optimized (i.e., minimized) when \(k_1 = k_2 = 1\).

**Proof.** Figure 5.2 lists the four cases of rendezvous under the symmetric model. The rendezvous in Figure 5.2(a), 5.2(b), and 5.2(c) occur when \(R_1 - k_1 \neq R_2 - k_2\). Since \(C^1 = C^2\), without loss of generality, we assume that \(\lceil \frac{|C^1|}{R_1-k_1} \rceil > \lceil \frac{|C^2|}{R_2-k_2} \rceil\). The rendezvous in Figure 5.2(d) occurs when \(R_1 - k_1 = R_2 - k_2\) and \(\lceil \frac{|C^1|}{R_1-k_1} \rceil = \lceil \frac{|C^2|}{R_2-k_2} \rceil\). A remarkable difference between them is whether the lengths of each round of the two users are the same.

1. **Case 1:** Figure 5.2(a), \(l' \geq \lceil \frac{|C^1|}{R_2-k_2} \rceil\) implies that there is a permutation of
all channels before the stay radios of user 1 transfer to next $k_1$ channels. The rendezvous is achieved between the hopping radios of user 2 and the stay radios of user 1 during the first $\lceil \frac{|C^2|}{R_2-k_2} \rceil$ time slots. That is, $TTR \leq \lceil \frac{|C^2|}{R_2-k_2} \rceil$.

2. Case 2: Figure 5.2(b) In this case, $l' < \lceil \frac{|C^2|}{R_2-k_2} \rceil$ implies that there is no enough time slots for user 2 to have permutation of all channels before the stay radios of user 1 transfer to next $k_1$ channels. The rendezvous can only be guaranteed between the hopping radios of user 2 and the stay radios of user 1 during the next $\lceil \frac{|C^2|}{R_2-k_2} \rceil$ time slots after the stay radios of user 1 transfers to next $k_1$ channel. That is, $TTR \leq 2 \times \lceil \frac{|C^2|}{R_2-k_2} \rceil - 1$.

3. Case 3: Figure 5.2(c) User 2 starts firstly. User 2 has a permutation of all channels in any $\lceil \frac{|C^2|}{R_2-k_2} \rceil$ consecutive time slots. When user 1 starts, it stay on one channel for $\lceil \frac{|C^1|}{R_1-k_1} \rceil$ time slots. $\lceil \frac{|C^1|}{R_1-k_1} \rceil > \lceil \frac{|C^2|}{R_2-k_2} \rceil$. Thus, a rendezvous is guaranteed before $\lceil \frac{|C^2|}{R_2-k_2} \rceil$ time slots. That is, $TTR \leq \lceil \frac{|C^2|}{R_2-k_2} \rceil$.

4. Case 4: Figure 5.2(d) $R_1 - k_1 = R_2 - k_2$. The rendezvous is achieved during the first $\lceil \frac{|C^1|}{R_1-k_1} \rceil$ (or $\lceil \frac{|C^2|}{R_2-k_2} \rceil$) time slots.

When $R_1 - k_1 \neq R_2 - k_2$, in the above analysis, $\lceil \frac{|C^1|}{R_1-k_1} \rceil$ is replaced by $\beta = max\{\lceil \frac{|C^1|}{R_1-k_1} \rceil, \lceil \frac{|C^2|}{R_2-k_2} \rceil\}$ and $\lceil \frac{|C^2|}{R_2-k_2} \rceil$ by $\gamma = min\{\lceil \frac{|C^1|}{R_1-k_1} \rceil, \lceil \frac{|C^2|}{R_2-k_2} \rceil\}$. According to the analysis of these cases, we prove that the MTTR is $(2\beta - 1)$. Combining with the occurrence probabilities we derive an upper-bound of $E(TTR)$ under the symmetric model. The $E(TTR) \leq \frac{1}{2} \times [\frac{\beta-\gamma+1}{\gamma} \times \beta + \frac{\beta-1}{\gamma} \times (2\beta - 1)] + \frac{1}{2} \times \beta \leq \beta + \frac{(\beta-1)^2}{2\gamma}$.

There is only one case when $R_1 - k_1 = R_2 - k_2$. The upper-bounds of MTTR and $E(TTR)$ are both $\lceil \frac{|C^1|}{R_1-k_1} \rceil$ or $\lceil \frac{|C^2|}{R_2-k_2} \rceil$. They are both smaller than the results when $R_1 - k_1 \neq R_2 - k_2$.

Above all, the upper-bounds of MTTR and $E(TTR)$ are $(2\beta - 1)$ and $\beta + \frac{(\beta-1)^2}{2\gamma}$, respectively, which can be optimized (minimized) when we set $k_1 = k_2 = 1$ under the symmetric model. Theorem 8 is proved.
Theorem 9. Under the asymmetric model \((C^1 \neq C^2)\), let \(k_1 (1 \leq k_1 < R_1)\) and \(k_2 (1 \leq k_2 < R_2)\) denote the numbers of stay radios of two users. \(G\) is the number of commonly-available channels of the two users. The MTTR of AMRR is upper-bounded by \(\max\{\left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor, \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor\} \times (F(|C^1|, |C^2|) + 2)\), where

\[
F(|C^1|, |C^2|) = \begin{cases} 
\left\lfloor \frac{|C^1| - G_{k_1}}{k_1} \right\rfloor, & \text{if } \left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor > \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor \\
\min\{\left\lfloor \frac{|C^1| - G_{k_1}}{k_1} \right\rfloor, \left\lfloor \frac{|C^2| - G_{k_2}}{k_2} \right\rfloor\}, & \text{if } \left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor = \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor \\
\left\lfloor \frac{|C^2| - G_{k_2}}{k_2} \right\rfloor, & \text{if } \left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor < \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor 
\end{cases}
\]

When \(k_1 = \frac{R_1}{2} \) and \(k_2 = \frac{R_2}{2}\), the MTTR of AMRR is optimized (i.e., minimized) to be \(O\left(\frac{|C_1||C_2|}{R_1 R_2}\right)\) which is lowest (i.e., optimal), in the order of magnitude, among any deterministic rendezvous algorithms when \(R_1 = R_2\) under the asymmetric model.

Proof. Similar with the symmetric model, we consider four cases based on their starting time. The rendezvous in Figure 5.3(a), 5.3(b), and 5.3(c) occur when \(\left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor < \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor\). The rendezvous in Figure 5.3(d) occurs when \(\left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor = \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor\).

1. Case 1: Figure 5.3(a) \(l' \geq \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor\) implies that user 2 has a permutation of all channels before \(\left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor\) and the \(k_1\) stay radios of user 1 stay on \(k_1\) channels during this period. There are \(k_1\) times potential rendezvous before \(\left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor\). These channels may not be commonly-available to two users. The next \(k_1\) times potential rendezvous can be guaranteed in the next round of user 1 (\(\left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor\) to \(2 \times \left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor\) in Figure 5.3(a)) because only after these time slots the stay radios of user 1 will transfer to the next \(k_1\) channels. We can say, under asymmetric model, we expect a rendezvous between the stay radios of the user with longer round (user 1) and the hopping radios of the user with shorter round (user 2). The worst case is the first potential \(\left\lfloor \frac{|C^1| - G_{k_1}}{k_1} \right\rfloor \times k_1\) times rendezvous channels are all not commonly-available to two users. There must be one stay radio of user 1 which stays on a commonly-available channel from time slot \(\left\lfloor \frac{|C^1| - G_{k_1}}{k_1} \right\rfloor \times \left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor\). Above all, the MTTR should be equal to or smaller than \(\left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor + \left\lfloor \frac{|C^1|}{R_1 - k_1} \right\rfloor\) times potential rendezvous can be guaranteed in the next round of user 1.

2. Case 2: Figure 5.3(b) \(l' < \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor\) implies that there is no enough time slots for user 2 to have permutation of all channels before the stay
radios of user 1 transfer to next $k_1$ channels. The first $k_1$ potential rendezvous can only be guaranteed between the hopping radios of user 1 and the stay radios of user 2 during the next $\lceil \frac{|C_1|}{R_1-k_1} \rceil$ time slots after the stay radio of user 1 transfers to next channel. Similar to case 1, there are $\lceil \frac{|C_1|}{k_1} \rceil$ times potential rendezvous. That is, $TTR \leq \ell' + \lceil \frac{|C_1|}{R_1-k_1} \rceil \times \lceil \frac{|C_1|}{k_1} \rceil + \lceil \frac{|C_2|}{R_2-k_2} \rceil \leq 2 \times \lceil \frac{|C_1|}{R_2-k_2} \rceil - 1 + \lceil \frac{|C_1|}{R_1-k_1} \rceil \times \lceil \frac{|C_2|}{k_1} \rceil$.

3. Case 3: Figure 5.3(c). User 2 starts firstly. The first $k_1$ potential rendezvous can be achieved before $\lceil \frac{|C_2|}{R_2-k_2} \rceil$. Thus, $TTR \leq \lceil \frac{|C_2|}{R_2-k_2} \rceil + \lceil \frac{|C_1|}{R_1-k_1} \rceil \times \lceil \frac{|C_1|}{k_1} \rceil$.

4. Case 4: Figure 5.3(d). $\lceil \frac{|C_1|}{R_1-k_1} \rceil = \lceil \frac{|C_2|}{R_2-k_2} \rceil$. Consider the worst case. If $|C_1| > |C_2|$, the first $\lceil \frac{|C_1|}{k_1} \rceil \times k_2$ times potential rendezvous are all fail since the channels that stay radios stay on are not a commonly-available channel. The $k_2$ stay radios of user 2 will absolutely hop on one commonly-available channel in the next $\lceil \frac{|C_2|}{R_2-k_2} \rceil$ time slots and there is a rendezvous. That is, $TTR \leq \lceil \frac{|C_2|}{R_2-k_2} \rceil + \lceil \frac{|C_1|}{k_1} \rceil \times \lceil \frac{|C_2|}{k_2} \rceil = \lceil \frac{|C_2|}{R_2-k_2} \rceil \times (\lceil \frac{|C_1|}{k_1} \rceil + 1)$. If $|C_1| < |C_2|$, after $\lceil \frac{|C_1|}{k_1} \rceil \times k_1$ times potential rendezvous, the stay radios of user 1 will certainly hop on one commonly-available channel in the next $\lceil \frac{|C_1|}{R_1-k_1} \rceil$ time slots and there is a rendezvous. That is, $TTR \leq \lceil \frac{|C_2|}{R_2-k_2} \rceil + \lceil \frac{|C_1|}{R_1-k_1} \rceil \times \lceil \frac{|C_2|}{k_1} \rceil \times (\lceil \frac{|C_1|}{k_1} \rceil + 1)$. Above all, $TTR \leq \lceil \frac{|C_2|}{R_2-k_2} \rceil \times (\min\{\lceil \frac{|C_1|}{k_1} \rceil, \lceil \frac{|C_2|}{k_2} \rceil\} + 1)$ in this case.

If $\lceil \frac{|C_1|}{R_1-k_1} \rceil > \lceil \frac{|C_2|}{R_2-k_2} \rceil$, we exchange the two roles of user 1 and user 2. Then we get $\lceil \frac{|C_1|}{R_1-k_1} \rceil + \lceil \frac{|C_2|}{R_2-k_2} \rceil \times \lceil \frac{|C_2|}{k_2} \rceil, 2 \times \lceil \frac{|C_1|}{R_1-k_1} \rceil - 1 + \lceil \frac{|C_2|}{R_2-k_2} \rceil \times \lceil \frac{|C_2|}{k_2} \rceil$ and $\lceil \frac{|C_1|}{R_1-k_1} \rceil + \lceil \frac{|C_1|}{R_2-k_2} \rceil \times \lceil \frac{|C_1|}{k_2} \rceil$ in the cases of Figure 5.3(a) 5.3(b) and 5.3(c). Above all, we give a conclusion of upper-bound of MTTR of AMRR under the asymmetric model.

\[
TTR \leq \max\{\lceil \frac{|C_1|}{R_1-k_1} \rceil, \lceil \frac{|C_2|}{R_2-k_2} \rceil\} \times (F(|C_1|, |C_2|) + 2),
\]

where

\[
F(|C_1|, |C_2|) = \begin{cases} \lceil \frac{|C_1|}{k_1} \rceil, & \text{if } \lceil \frac{|C_1|}{R_1-k_1} \rceil > \lceil \frac{|C_2|}{R_2-k_2} \rceil \\ \min\{\lceil \frac{|C_1|}{k_1} \rceil, \lceil \frac{|C_2|}{k_2} \rceil\}, & \text{if } \lceil \frac{|C_1|}{R_1-k_1} \rceil = \lceil \frac{|C_2|}{R_2-k_2} \rceil \\ \lceil \frac{|C_2|}{k_2} \rceil, & \text{if } \lceil \frac{|C_1|}{R_1-k_1} \rceil < \lceil \frac{|C_2|}{R_2-k_2} \rceil \end{cases}
\]

Above all, the denominator of the upper-bounds of MTTR is either $k_1(R_1-k_1)$ or $k_2(m_2-k_2)$. To optimize them, we should maximize $k_1(R_1-k_1)$ or $k_2(R_2-k_2)$.  

85
(a) \[ TTR \leq \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor + \left\lceil \frac{|C^1|}{R_1 - k_1} \right\rceil \times \left\lfloor \frac{|C^1|-G}{k_1} \right\rfloor \]

(b) \[ TTR \leq 2 \times \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor - 1 + \left\lceil \frac{|C^1|}{R_1 - k_1} \right\rceil \times \left\lfloor \frac{|C^1|-G}{k_1} \right\rfloor \]

(c) \[ TTR \leq \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor + \left\lceil \frac{|C^1|}{R_1 - k_1} \right\rceil \times \left\lfloor \frac{|C^1|-G}{k_1} \right\rfloor \]

(d) \[ TTR \leq \left\lfloor \frac{|C^2|}{R_2 - k_2} \right\rfloor \times \left\lfloor \min\{\left\lfloor \frac{|C^1|-G}{k_1} \right\rfloor, \left\lceil \frac{|C^2|-G}{k_2} \right\rceil \right\} + 1) \]

Figure 5.3: Four cases of AMRR under the asymmetric model.
It is easy to know that the upper-bounds of MTTR is minimized when \( k_1 = \frac{R_1}{2} \) or \( k_2 = \frac{R_2}{2} \). That is when user 1 assigns \( \frac{R_1}{2} \) radios as stay radios and user 2 assigns \( \frac{R_2}{2} \) radios as stay radios, \( MTTR = 2 \times \max\{\lceil \frac{|C_1|}{R_1} \rceil, \lceil \frac{|C_2|}{R_2} \rceil\} \times (2F(|C_1|, |C_2|) + 2) \), where

\[
F(|C_1|, |C_2|) = \begin{cases} 
\lfloor \frac{|C_1| - G}{R_1} \rfloor, & \text{if } \lceil \frac{|C_1|}{R_1} \rceil > \lceil \frac{|C_2|}{R_2} \rceil \\
\min\{\lceil \frac{|C_1| - G}{m_1} \rceil, \lceil \frac{|C_2| - G}{R_2} \rceil\}, & \text{if } \lceil \frac{|C_1|}{R_1} \rceil = \lceil \frac{|C_2|}{R_2} \rceil \\
\lfloor \frac{|C_2| - G}{R_2} \rfloor, & \text{if } \lceil \frac{|C_1|}{R_1} \rceil < \lceil \frac{|C_2|}{R_2} \rceil 
\end{cases}
\]

According to the Theorem 4 in [30], any deterministic rendezvous algorithms require at least \( \frac{|C_1||C_2|}{R_2} \) time slots to guarantee rendezvous when the users are equipped with \( m \) radios. That is, in the order of magnitude, the lower-bound of MTTR is \( O\left(\frac{|C_1||C_2|}{R_2}\right) \). When the two users are equipped with \( R \) radios, the upper-bound of MTTR of AMRR is \( O\left(\frac{|C_1||C_2|}{R^2}\right) \) which meets this lower-bound. Theorem 9 is proved.

5.3 Simulation

We built a simulator in Visual Studio 2010 to evaluate the proposed AMRR algorithm. Among the three existing algorithms using multiple radios, RPS [39] and GCR [23] give upper-bounded MTTRs while EAR [30] does not. In particular, MTTR of GCR has been shown to meet the lower bound [23]. Thus, we select RPS and GCR for comparison in the simulation. The performance is measured in terms of the average TTR and the maximum TTR, where TTR is counted as the number of time slots required to achieve rendezvous. We consider both the symmetric model and the asymmetric model.

We consider the following key parameters. The number of channels in the whole channel set \( |C| \) is varied from 10 to 100. Let \( \theta (0 < \theta \leq 1) \) be the ratio of number of available channels to the number of all channels. We randomly select \( \theta \times |C| \) channels for each user from the whole channel set. For each set of parameter values, we perform 1,000,000 independent runs and then compute the average TTR and the maximum TTR.
5.3.1 Performance of different allocations of stay radios ($k$)

Firstly we want to evaluate the performance of different allocations of stay radios. The theoretical results we get in Section 5.2 are the upper-bounds. Now we want to show the simulated results. We fix $R_1 = 4$ and $R_2 = 4$. $k_1$ and $k_2$ are the numbers of stay radios of two users. In this subsection, we have three combinations of $k_1$ and $k_2$ which are ($k_1 = k_2 = 1$), ($k_1 = k_2 = 2$) and ($k_1 = k_2 = 3$) ($1 \leq k < R$).

Under the Symmetric Model

Figure 5.4 shows the comparison of three combinations under the symmetric model when the numbers of available channels of two users are $|C^1| = |C^2| = 0.4|C|$ ($C^1 = C^2$). That is, about 40\% of all channels are available to users. It can be seen that: 1) When the number of available channels increases, both the $E(TTR)$ and $MTTR$ increase; 2) In terms of both $E(TTR)$ and $MTTR$, AMRR$_{(k_1=k_2=1)}$ has the best performance. For example, when there are 100 channels, $E(TTR)$s of the three combinations are 2.99, 6.09 and 11.90 while $MTTR$s of the three combinations are 13, 20 and 38 respectively. According to Theorem 8 in Section 5.2, when $|C^1| = |C^2| = 0.4 \times 100 = 40$, MTTRs of the three combinations are smaller than $(2 \times max\{\lceil \frac{|C^1|}{R_1-k_1} \rceil, \lceil \frac{|C^1|}{R_2-k_2} \rceil \} - 1) = 27, 39, 79$ respectively. This theoretical result is verified in the simulation. The result is optimized when $k_1 = 1$ and $k_2 = 1$.

Under the Asymmetric Model

Figure 5.5 shows the comparison of three combinations under the asymmetric model when the numbers of available channels of two users are $|C^1| = |C^2| = 0.4|C|$ ($C^1 \neq C^2$) and the number of available channels $G = 0.2|C|$ (half of the available channels are commonly-available to two users). We observe different properties. In terms of $E(TTR)$, in Figure 5.5(a), AMRR$_{(k_1=k_2=1)}$ has the best performance. In terms of $MTTR$, in Figure 5.5(b) AMRR$_{(k_1=k_2=2)}$ has the best performance. For example, when there are 100 channels, $E(TTR)$s of three combinations are 5.82, 8.26 and 14.48 while $MTTR$s of three combinations are 148, 84 and 152 respectively. According to
Symmetric model: \( \theta = 0.4, m_1 = 4 \) and \( m_2 = 4 \)

Number of all channels (|C|)

Average TTR

# of stay radios \( k_1 = k_2 = 1 \)
# of stay radios \( k_1 = k_2 = 2 \)
# of stay radios \( k_1 = k_2 = 3 \)

(b) Maximum TTR VS. |C|

Figure 5.4: Comparison of different allocations when \( \theta = 0.4 \).

From Figure 5.4 and Figure 5.5, we can see that: 1) The performance is optimized when the number of stay radios is 1 under the symmetric model; 2) Under the asymmetric model, AMRR has better performance on MTTR by allocating half of the radios as the stay radios while has better performance on E(TTR) by allocating one radio as the stay radio. Our algorithm is an adjustable rendezvous algorithm for different service. If the MTTR is more important than E(TTR), we allocate half of the radios as stay radios. Otherwise, we allocate one radios as stay radio. This theoretical result is verified in the simulation. The MTTR is optimized when \( k_1 = \frac{R_1}{2} \) and \( k_2 = \frac{R_2}{2} \).

5.3.2 Comparisons of different algorithms

In this subsection, we compare our algorithm with existing algorithms. We study two allocations of the number of stay radios: AMRR\(_{(k_1=k_2=1)}\) (i.e., one radio is allocated as stay radio) and AMRR\(_{(k_1=k_2=2)}\) (i.e., half of the radios are allocated as...
Under the Symmetric Model

Firstly, we study the symmetric model. Figure 5.6 shows the comparison when the numbers of available channels of two users are $|C^1| = |C^2| = 0.4|C|$ ($C^1 = C^2$). We can see that AMRR$_{(k_1=k_2=1)}$ has the smallest MTTR and E(TTR). For example, when there are 100 channels, E(TTR)s of AMRR$_{(k_1=k_2=1)}$, AMRR$_{(k_1=k_2=2)}$, GCR and RPS are 2.99, 6.09, 11.34 and 4.40 while MTTRs of AMRR$_{(k_1=k_2=1)}$, AMRR$_{(k_1=k_2=2)}$, GCR and RPS are 13, 20, 90 and 31.

Under the Asymmetric Model

Then we study the asymmetric model. Firstly we study the scenario when $\theta$ is small, i.e., only a small portion of channels are available to users. Figure 5.7 shows the comparison when $|C^1| = |C^2| = 0.1|C|$ ($C^1 \neq C^2$) and $G = 1$. That is, about 10% of all channels are available to users and there is only one commonly-available channel between two users. We can see that AMRR$_{(k_1=k_2=1)}$ has the best performance in terms of E(TTR) while AMRR$_{(k_1=k_2=2)}$ has the best performance in terms of MTTR. For example, when there are 50 channels, E(TTR)s of AMRR$_{(k_1=k_2=1)}$, AMRR$_{(k_1=k_2=2)}$, GCR and RPS are 2.99, 6.09, 11.34 and 4.40 while MTTRs of AMRR$_{(k_1=k_2=1)}$, AMRR$_{(k_1=k_2=2)}$, GCR and RPS are 13, 20, 90 and 31.
GCR and RPS are 1.85, 2.24, 2.86 and 2.89 while MTTRs of AMRR\(_{(k_1=k_2=1)}\), AMRR\(_{(k_1=k_2=2)}\), GCR and RPS are 10, 9, 14 and 35. Then we increase the numbers of available channels of two users to |\(C^1| = |\(C^2| = 0.4|C| \) \((C^1 \neq C^2)\) and the number of commonly-available channels \(G = 0.2|C|\). In Figure 5.8, AMRR\(_{(k_1=k_2=1)}\) has the best performance in terms of E(TTR) but AMRR\(_{(k_1=k_2=2)}\) has the best performance in terms of MTTR. For example, when there are 100 channels, E(TTR)s of AMRR\(_{(k_1=k_2=1)}\), AMRR\(_{(k_1=k_2=2)}\), GCR and RPS are 5.82, 8.26, 12.70 and 8.98 while MTTRs of AMRR\(_{(k_1=k_2=1)}\), AMRR\(_{(k_1=k_2=2)}\), GCR and RPS are 148, 84, 120 and 107.

5.4 Chapter Summary

We proposed a new algorithm named Adjustable Multi-Radio Rendezvous (AMRR) algorithm which exploits multiple radios for fast rendezvous. We derived an upper-bound of MTTR of AMRR and proved that AMRR gives the lowest (i.e., optimal) MTTR, in the order of magnitude, when two users are equipped with the same number of radios. We showed that AMRR is the only adjustable algorithm among existing multi-radio rendezvous algorithms. It can give its best performance on either MTTR or E(TTR) with different allocations of radios. Using AMRR, all
Asymmetric model: $\theta = 0.1$, $G = 1$, $m_1 = 4$ and $m_2 = 4$

(a) Average TTR VS. $|C|$  
(b) Maximum TTR VS. $|C|$

Figure 5.7: Comparison of different algorithms when $\theta = 0.1$ and $G = 1$.

Asymmetric model: $\theta = 0.4$, $G = 0.2|C|$, $m_1 = 4$ and $m_2 = 4$

(a) Average TTR VS. $|C|$  
(b) Maximum TTR VS. $|C|$

Figure 5.8: Comparison of different algorithms when $\theta = 0.4$ and $G = 0.2|C|$.
users could select the value of $k$ individually and central controller is not needed. By adjusting the value of $k$, users can make a tradeoff between the two conflicting performance objectives $E(TTR)$ and MTTR. Simulation results show that AMRR can shorten the MTTR up to 85% when the number of average radios is 4, compared to the state-of-the-art.
Chapter 6

Cooperative Rendezvous in Multi-User Cognitive Radio Networks

The existing rendezvous algorithms implicitly assume that only one pair of users send handshaking messages for rendezvous at a time. In practice, more than one pair of users may go through the rendezvous process at about the same time and their handshaking messages may collide with each other. As a result, the rendezvous performance would degrade. In this chapter, we propose to turn this disadvantage (i.e., multiple users cause multiple access interference to each other) into an advantage (i.e., multiple users cooperate with each other for fast rendezvous). Specifically, we propose a Cooperative Rendezvous Protocol (CRP) by which multiple users cooperate with each other to relay the available channel set information. When a user knows the available channel set of the intended user, it can generate a more efficient channel hopping sequence based on the commonly-available channels. This speeds up rendezvous. System model and problem formulation are presented in Section 6.1. Protocol description is presented in Section 6.2. In Section 6.3, we analyze the performance of our protocol. We present simulation results in Section 6.4 for performance evaluation and conclude our work in Section 6.5.

6.1 System Model

We consider a CRN consisting of $K$ ($K \geq 2$) users. Two users are one pair intended to rendezvous. There are $\frac{K}{2}$ pairs. Time is divided into slots of equal duration. The licensed spectrum is divided into $Q$ non-overlapping channels $C = \{c_1, c_2, ..., c_Q\}$. Let $C^i \subseteq C$ be the set of available channels of user $i$ ($i = 1, 2, ..., K$), where a channel is said to be available to a user if the user can communicate on this channel without causing interference to any PUs. For each user, there is a ratio $\theta$ which is
the ratio of the number of available channels to the total number of channels $Q$. For simplicity, we suppose that each user randomly selects $\theta Q$ available channels from the system. Let $G$ be the number of commonly-available channels of user 1 and user 2. We consider the rendezvous of one pair of users, say user 1 and user 2.

To record the channel information, each user has $\Sigma$ bit vectors. One bit vector records the channel information and user ID of one user. When there are total 10 channels, $\overrightarrow{BV} = \{01000101000\}$ means that the available channel set is $\{2, 5, 7\}$. If there are 100 channels and 100 users, the length of each bit vector is $100 + \log_2 100 < 128$ bits (16 bytes). The user adds $16\Sigma$ bytes information into the handshaking message. For example, in Figure 6.1, there are four users and four channels. User 1 and user 2, user 3 and user 4 are two pairs of users intended to rendezvous. Each user has $\Sigma$ bit vectors to store the channel information of users. Each bit vector has corresponding user ID. The initial bit vector only stores the channel information of itself. The available channels of user 1 are $\{1, 3\}$, user 2 are $\{1, 4\}$, user 3 are $\{3, 4\}$ and user 4 are $\{2, 3\}$. After user 1 achieves rendezvous with user 3, they update the bit vectors and store the channel information from the opposite user. For example, for user 1, $BV_2 = \{0011\}$ and the corresponding user ID is 3. It means that user 1 has the channel information of user 3. After $BV_\Sigma$ is used, the user stores the new channel information from $BV_2$ to $BV_\Sigma$ repeatedly. That is, each user always stores the latest $(\Sigma - 1)$ users’ channel information it gets from other users.
Rendezvous problem for two users: Suppose two users have average $\theta Q$ available channels and they may start the rendezvous process at different time. Any existing rendezvous can be applied to this protocol. Each user determines a CH sequence based on the protocol and one existing rendezvous algorithm. The rendezvous success only when two users and there are only these two users hop on one same channel in the same time slot.

6.2 Protocol Description

In this study, we discuss a Cooperative Rendezvous Protocol (CRP) in CRNs with multiple pairs of users. This protocol can be applied to any existing rendezvous algorithms. Our basic idea is to make use of the cooperativity of all users in the system to relay the channel information. If a user gets the channel information of intended rendezvous user from another user, it can narrow the channel set and hop on the commonly-available channels only to speed the rendezvous process. In Figure 6.1, user 1 and user 3 get the channel information from the other. If user 3 achieve rendezvous with user 2 later, in Figure 6.2, user 2 can get the channel information of user 1 from user 3. From this time slot, user 2 only hops on the commonly-available channels between it and user 1, that is, channel 1 in this example. The bit vector of channel information of user 2 is changed from $\{1001\}$ to $\{1000\}$. Figure 6.3 shows the CH sequences of user 1, user 2 and user 3. In time slot 3, user 3 gets the channel information of user 1. In time slot 4, user 2 gets the channel information of user 1 from user 3 and generates the new CH sequence based on the intersection of its available channels and user 1’s available channels $\{1, 3\} \cap \{1, 4\} = \{1\}$. Before time slot 4, CH sequence of user 2 is based on its available channels $\{1, 4\}$. In time slot 5, user 1 achieves rendezvous with user 3 again. It can get the channel information of user 2 from user 3 and generates the new CH sequence based on the intersection of its available channels and user 2’s available channels. Finally, user 2 achieves rendezvous with user 1 on channel 1 in time slot 6.

When we apply CRP in conjunction with an existing algorithm F, for simplicity,
Figure 6.2: Updated bit vectors after user 3 achieves rendezvous with user 2.

Based on \{1,3\} \{1,4\}={1}

User 1 rendezvous with
User 3 for two times

Based on \{1,3\}

User 2 rendezvous
with User 3

Based on \{1,4\}

User 1 rendezvous
with User 2

Based on \{1,3\} \{1,4\}={1}

Figure 6.3: CH sequence of user 1, user 2 and user 3.
we suppose that the CH sequence of user 1 based on channel set $C^1$ is $F(C^1)$. User 1 and user 2 is one pair of users who intend rendezvous. The process of user 1 is formally presented as Figure 6.4. At the beginning, the CH sequence is generated based on the initial available channel set $C^1$. After it rendezvous with user $w$, it judges whether user $w$ is exactly user 2. If it is, the rendezvous success. If it is not, user 1 can get the channel information of other users from user $w$ and update its bit vectors. If there is channel information of user 2, it updates its available channel set to be the intersection of $C^1$ and $C^2$. Or it continues following the CH sequence generated based on $C^1$.

Figure 6.5 shows the system model of CRP when there are $K$ users in the system. We consider the rendezvous between user 1 and user 2. There are three stages of whole rendezvous process.

1. Stage 1: User 1 generates the CH sequence based on its available channels
set $C_1$ while user 2 generates the CH sequence based on its available channels set $C_2$. The process of rendezvous of user 1 and other users (user 3 to user $K$) is defined as P1. The process of rendezvous of user 2 and other users is defined as P2. The process of rendezvous of user 1 and user 2 is defined as P3. There are two cases of stage 1. 1) If any user (user 3 to user $K$) finishes P1 and P2 before P3, stage 1 end. If the user finishes P1 before P2, user 2 gets the channel information of user 1 and changes the available channels to be the commonly-available channels of user 1 and user 2: $C_1 \cap C_2$. If the user finishes P2 before P1, user 1 will change the available channels to be the commonly-available channels of user 1 and user 2: $C_1 \cap C_2$. Then stage 2 begins. 2) If all users finish P1 and P2 later than P3, user 1 and user 2 finish rendezvous without the cooperation of other users. Stage 1 finishes and there is no need to start stage 2. The whole rendezvous process ends.

2. Stage 2: If one of user 1 and user 2 changes the available channels set, the user
generates CH sequences based on the new available channel set. We suppose that user 2 changes. There are two new processes. The rendezvous of user 1 and other users is defined as P4. The rendezvous of user 1 and user 2 is defined as P5. Same with stage 1, there are two cases. 1) If any user (user 3 to user \( K \)) finish P4 before P5, stage 2 end. User 1 also gets the channel information of user 2 and changes the available channels to be the commonly-available channels of user 1 and user 2: \( C^1 \cap C^2 \). Then stage 3 begins. 2) If all users finish P4 later than P5, user 1 and user 2 finish rendezvous. There is no need to start stage 3. The whole rendezvous process ends.

3. Stage 3: Both user 1 and user 2 generate CH sequences based on the commonly-available channels. The rendezvous of user 1 and user 2 is defined as P6. The whole rendezvous process ends until P6 finished.

### 6.3 Protocol Analysis

In this section, we theoretically analyze the advantage of our protocol. We divide the analysis into four parts. The first part is the estimation of the expectation of the number of commonly-available channels \( G \). The second part is the estimation of the latency of one user finishing P1 and P2 in stage 1. The third part is the analysis of the simplest rendezvous algorithm: Random Algorithm. The fourth part is the analysis of a general existing rendezvous algorithm which generates CH sequences based on available channels. We suppose that \( 0 < \theta < 1 \). The core of our protocol is to make use of the cooperativity of multiple users to relay the channel information so that the user can narrow the channel set and speed the rendezvous efficiently. It has no advantage on the theoretical results of existing algorithms which are based of the whole channels set. However, based on our investigation, when the number of available channels decreases, the average TTR of almost all existing rendezvous algorithms which are based of the whole channels set also decreases even there is no such behavior in theory [15, 37, 1]. We will show the performance our protocol...
applied in conjunction with this kind of algorithms in Section 6.4.

6.3.1 The expectation of $G$

When we fix the ratio ($\theta$) of number of available channels to the number of all channels, we need estimate the number of commonly-available channels ($G$) between two users.

**Lemma 3.** When two users randomly select $\theta Q$ channels from all $Q$ channels, the expectation of $G$ is $\theta^2 Q$.

**Proof.** For each channel, there are three cases: 1) Both of 2 users are available to use it. It is one of the commonly-available channels. The probability is $\theta^2$. 2) Only one of them is available to use it. The probability is $2\theta(1-\theta)$. 3) Both of 2 users are unavailable to use it. The probability is $(1-\theta)^2$. There are total $Q$ channels. The expectation of $G$ is $\theta^2 Q$.

We use another mathematical method to prove it. After user 1 fix the $\theta Q$ channels, user 2 selects another $\theta Q$ channels. The number of commonly-available channels $G$ is the number of channels which are selected from user 1’s $\theta Q$ available channels. The expectation of $G$ is $\sum_{i=0}^{\theta Q} \binom{\theta Q}{i} \cdot \binom{\theta Q}{\theta Q - i}$. According to combinatorial identity $\sum_{i=0}^{m} \binom{m}{i} \times \binom{m-i}{n} = \binom{m}{m+n}$, we have

$$\sum_{i=0}^{\theta Q} \binom{\theta Q}{\theta Q - i} \binom{i-1}{\theta Q - 1} = \binom{\theta Q - 1}{\theta Q - 1}. \quad (6.3.1)$$

Then we get

$$\sum_{i=0}^{\theta Q} \binom{\theta Q - i}{\theta Q} \cdot i = \binom{\theta Q}{\theta Q} \times \theta^2 Q. \quad (6.3.2)$$

Therefore, the expectation of $G$ is $\sum_{i=0}^{\theta Q} \binom{\theta Q}{\theta Q - i} \cdot \binom{i}{\theta Q} = \theta^2 Q$. \hfill \Box

6.3.2 The fitting of the expected latency of P1 and P2 ($L_1$)

In this part, we want to estimate the latency for the earliest user finishing P1 and P2. That is, the latency for user 1 or user 2 gets the channel information of the other from other users in the system.
For stage 1, we need to estimate the expected latency of the earliest user finishing P1 and P2 before P3. We use \( L_1 \) to denote it. It is too complicated to compute it by expression. We collect much data and fit the relationship between \( L_1, \theta \) and \( K \). We find that the distribution of \( L_1 \) is in close proximity to Gamma Distribution for each setting of \( \theta \) and \( K \). Figure 6.6 shows the distribution of \( L_1 \) when \( \theta = 0.3 \) and \( K = 160 \) where \( K \) is the number of users and \( \theta \) is the ratio of the number of available channels to the number of all channels. The parameters of Gamma Distribution are \( a = 3.67169 \) and \( b = 8.29843 \). The Probability density function of Gamma Distribution is \( f(x) = \frac{1}{\Gamma(a)b^a}x^{a-1}e^{-\frac{x}{b}} \). The expected value is \( E[x] = ab \). We fix \( Q = 100 \), vary \( K \) from 50 to 200 and \( \theta \) from 0.1 to 0.9. For each setting of \((K, \theta)\), we get an approximate \( a \) and \( b \). Then we fit one more time to get the relationship between \( a, b \) and \((K, \theta)\). Figure 6.7 shows the optimal fitting of \( a \). We get that

\[
a \approx 2.834 + 0.2688\theta + 0.003607K - 0.1982\theta^2 \tag{6.3.3}
\]

and

\[
b \approx 7.535 - 0.2895\theta + 0.01495K + 0.1241\theta^2 \tag{6.3.4}
\]

Therefore, \( E[L_1] = a \times b \leq 21.354 + 0.00005392K^2 + 0.0642K + 1.212\theta \). When \( K \leq 200 \) and \( \theta < 1 \), \( E[L_1] \leq 21.354 + 2.1568 + 12.84 + 1.212 = 37.56 \).

### 6.3.3 The analysis of Random Algorithm

To prove that there is improvement when we apply CRP in conjunction with an existing algorithm, we start from the simplest rendezvous algorithm: Random Algorithm. In this algorithm, each user randomly hops on an available channels in each time slot without any schedule.

**Theorem 10.** When we ignore collision and two users randomly selects \( \theta Q \) channels from all \( Q \) channels, the expected TTR of Random Algorithm is \( Q \), which is independent of \( \theta \).
Figure 6.6: Distribution of $L_1$.

Figure 6.7: Optimal-fitting of $a$. 
**Proof.** When two users randomly selects $\theta Q$ channels from all channels, the number of commonly-available channels is $G$. If two users achieve rendezvous, they must hop on one of these $G$ channels at the same time slot. For each channel of these $G$ channels, the probability that both of two users hop on it is $\frac{1}{\theta Q} \times \frac{1}{\theta Q} = \frac{1}{(\theta Q)^2}$. There are total $G$ commonly-available channels, so in each time slot, the probability of two users to achieve rendezvous is $\frac{G}{(\theta Q)^2} = \frac{\theta^2 Q}{(\theta Q)^2} = \frac{1}{\theta Q}$. The TTR of Random Algorithm follows Exponential Distribution where the inverse scale is $\lambda = \frac{1}{\theta Q}$. Therefore, the expected TTR of Random Algorithm is $\frac{1}{\lambda} = Q$. It is independent of $\theta$. \qed

**Theorem 11.** When $Q = 100$ and $K \leq 200$, we consider collision and apply CRP in conjunction with Random Algorithm. Two users randomly selects $\theta Q$ channels from all $Q$ channels. The expected TTRs are $Q$, $\theta Q + 37.56$ and $\theta^2 Q + \theta Q + 37.56$ when the two users achieve rendezvous in stage 1, 2 and 3, respectively.

There are three cases of the rendezvous of two users when we apply CRP in conjunction with Random Algorithm. 1) Two users achieve rendezvous in stage 1. 2) Two users achieve rendezvous in stage 2. 3) Two users achieve rendezvous in stage 3.

**Subcase 1):** In Figure 6.5, P3 is finished before both P1 and P2 are finished by any user (user 3 to user $K$). Two users achieve rendezvous without the cooperation from other users. Or $\theta = 1$, there is no benefit even the user has the channel information. In this case, the expected TTR is the original expected TTR in Theorem 10 which is $E[stage 1] = Q$.

**Subcase 2):** In Figure 6.5, both P1 and P2 are finished by one user (user 3 to user $K$) before P3 is finished. User 2 gets the channel information of user 1, the channel set is changed to the commonly-available channels. P5 is finished before P4. Two users achieve rendezvous in stage 2. The numbers of channels of two users are $G$ and $\theta Q$ respectively. The probability for two users to achieve rendezvous in each time slot is $p = \frac{G}{G} \times \frac{1}{\theta Q} = \frac{1}{\theta Q}$. The process follows Exponential Distribution. The expected TTR of stage 2 should be $\frac{1}{p} = \theta Q$. Therefore, the expected TTR in
this case is

\[ E[\text{stage 2}] = \theta Q + E[L_1] = \theta Q + 37.56 \quad (6.3.5) \]

If only \( \theta Q + 37.56 < Q \), there is improvement when we apply CRP in conjunction with Random Algorithm. The improvement is \((1 - \theta)Q - 37.56\). It decreases when \( \theta \) increases. For example, when \( Q = 100 \), there is improvement only if \( \theta < 0.6244 \) in theory.

**Subcase 3:** In Figure 6.5, P4 is finished before P5 is finished. User 1 gets the channel information of user 2. The channel sets of both user 1 and user 2 are changed to the commonly-available channels set. Two users achieve rendezvous in stage 3. This case only happened when P4 is finished before P5. The expected latency for two users to get the channel information of each other \( E[L_2] \leq E[\text{stage 2}] = \theta Q + 37.56 \). After both of two users change the channel set to be the commonly-available channel set, the probability for two users to achieve rendezvous in each time slot is \( p = \frac{G}{G} \times \frac{1}{\theta^2 Q} = \frac{1}{\theta^2 Q} \). The expected TTR of stage 3 should be \( \frac{1}{p} = \theta^2 Q \). Therefore, the expected TTR in this case is

\[ E[\text{stage 3}] = \theta^2 Q + E[L_2] \leq \theta^2 Q + \theta Q + 37.56 \quad (6.3.6) \]

If only \( \theta^2 Q + \theta Q + 37.56 < Q \), there is improvement when we apply CRP in conjunction with Random Algorithm. The improvement is \((1 - \theta^2 - \theta)Q - 37.56\). It decreases when \( \theta \) increases. For example, when \( Q = 100 \), there is improvement only if \( \theta < 0.4351 \) in theory.

Those results are estimated without considering the collision. Actually in the simulation part, we shows that CRP has improvement in terms of any \( \theta \).

### 6.3.4 The analysis of a general existing algorithm which is based on available channel set

In this subsection, we analyze the performance when we apply CRP in conjunction with a general existing algorithm. The algorithm generate CH sequences based on
available channels. Without loss of generality, we consider the rendezvous of user 1 (the available channel set is $C^1$) and user 2 (the available channel set is $C^2$).

**Theorem 12.** There is a general existing algorithm which generates CH sequences based on available channel set. The expected TTR is $F(|C^1|, |C^2|) = |C^1|^{x_1}|C^2|^{x_2}$ where $|C^1|$, $|C^2|$ are the numbers of available channels of two users. When we consider collision and apply CRP in conjunction with this algorithm, two users randomly selects $\theta Q$ channels from all $Q$ channels, the expected TTRs are $(\theta Q)^{x_1+x_2}$, $(\theta^2 Q)^{x_1}(\theta Q)^{x_2} + 37.56$ and $(\theta^2 Q)^{x_1+x_2} + (\theta^2 Q)^{x_1}(\theta Q)^{x_2} + 37.56$ when the two users achieve rendezvous in stage 1, 2 and 3, respectively.

**Proof.** The expected TTR of the existing algorithm can be expressed as $F(|C^1|, |C^2|) = (\theta Q)^{x_1+x_2}$. When we apply CRP in conjunction with this algorithm, same with previous subsection, there are three cases.

**Subcase 1:** Two users achieve rendezvous without the cooperation from other users. Or $\theta = 1$, there is no benefit even the user has the channel information. The expected TTR is the original expected TTR, which is $E[\text{stage 1}] = (\theta Q)^{x_1+x_2}$.

**Subcase 2:** Only user 1 gets the channel information of user 2 and the channel set of user 1 is changed to the commonly-available channels. The expected TTR should be

$$E[\text{stage 2}] = F(G, |C^2|) + E[L_1] = (\theta^2 Q)^{x_1}(\theta Q)^{x_2} + 37.56$$

(6.3.7)

Thus, there is improvement when we apply CRP in conjunction this algorithm if only $(\theta^2 Q)^{x_1}(\theta Q)^{x_2} + 37.56 < (\theta Q)^{x_1+x_2}$. We define

$$H(\theta) = (\theta Q)^{x_1+x_2} - (\theta^2 Q)^{x_1}(\theta Q)^{x_2} - 37.56$$

(6.3.8)

Then we try to get the extreme value of $H$ when $\theta$ vary from 0.1 to 0.9.

$$\frac{dH}{d\theta} = (x_1 + x_2) - (2x_1 + x_2)\theta^{x_1}(\theta Q)^{x_1+x_2-1}Q$$

(6.3.9)

Let $\frac{dH}{d\theta} = 0$, we get $\theta^* = (\frac{x_1 + x_2}{2x_1 + x_2})^{\frac{1}{2}}$. When $\theta < \theta^*$, $\frac{dH}{d\theta} > 0$. When $\theta > \theta^*$, $\frac{dH}{d\theta} < 0$. So it is the largest value when $\theta = \theta^*$. Based on our investigation, among
the existing algorithm, the best theoretical result of expected TTR is $O(|C^1||C^2|)$. That is, $x_1 = x_2 \geq 1$. $\theta^* = \frac{2}{3}$. $H(0.1) \geq 0.9 \times 10^2 - 37.56 = 52.44$ and $H(0.9) \geq 0.1 \times 90^2 - 37.56 = 772.44$. If only $H(0.1) > 0$ and $H(0.9) > 0$, $H > 0$ when $\theta$ vary from 0.1 to 0.9.

In this case, CRP has improvement in terms of any $\theta$ from 0.1 to 0.9. It has the most significant improvement when $\theta = 0.6667$.

Subcase 3): User 1 also gets the channel information of user 2. The channel sets of both user 1 and user 2 are changed to the commonly-available channels. The expected latency for two users to get the channel information of each other $E[L_2] \leq E[\text{stage 2}] = (\theta^2 Q)^{x_1} + (\theta^2 Q)^{x_2} + 37.56$. The expected TTR should be

$$E[\text{stage 3}] = F(G, G) + E[L_2] = (\theta^2 Q)^{x_1 + x_2} + (\theta^2 Q)^{x_1} (\theta Q)^{x_2} + 37.56 \quad (6.3.10)$$

Thus, there is improvement when we apply CRP in conjunction with this algorithm if only $(\theta^2 Q)^{x_1 + x_2} + (\theta^2 Q)^{x_1} (\theta Q)^{x_2} + 37.56 < (\theta Q)^{x_1 + x_2}$. We define

$$H(\theta) = (1 - \theta^{x_1 + x_2})(\theta Q)^{x_1 + x_2} - 37.56 \quad (6.3.11)$$

Then we try to get the extreme value of $H$ when $\theta$ vary from 0.1 to 0.9.

$$\frac{dH}{d\theta} = ((x_1 + x_2) - (2x_1 + x_2)\theta^{x_1} - (2x_1 + 2x_2)\theta^{x_1 + x_2})\theta Q^{x_1 + x_2 - 1}Q \quad (6.3.12)$$

Let $\frac{dH}{d\theta} = 0$, we get $(2x_1 + x_2)\theta^{x_1} + (2x_1 + 2x_2)\theta^{x_1 + x_2} = x_1 + x_2$. When $\theta < \theta^*$, $\frac{dH}{d\theta} > 0$. When $\theta > \theta^*$, $\frac{dH}{d\theta} < 0$. Thus, $\frac{dH}{d\theta}$ has the largest value when $\theta = \theta^*$. When $x_1 = x_2 = 1, 4\theta^* + 3\theta^* - 2 = 0$. $\theta^* \approx 0.4254$. When $Q = 100, (1 - \theta^2 - \theta)\theta^2 \geq 0.003756$. $0.0635 \leq \theta \leq 0.6136$. $H(0.1) \geq 0.89 \times 10^2 - 37.56 = 51.44, H(\theta^*) = 674.78$ and $H(0.6136) = 0$. $H > 0$ when $\theta > 0.6136$.

In this case, CRP has improvement when $\theta < 0.6136$. The improvement is $(1 - \theta^2 - \theta)(\theta Q)^{x_1} - 37.56$. It has the most significant improvement when $\theta = 0.4254$. 

\[\square\]
6.4 Simulation

We built a simulator in Matlab R2013b to evaluate the effectiveness of the proposed protocol (i.e., all users are cooperative to relay channel information). We first evaluate the collision of the existing algorithms when there are multiple pairs of users in the system. The performance of the proposed protocol is shown later. Random Algorithm [35], EJS [26] and MTP [16] are selected to be compared. Random Algorithm is the simplest algorithm. EJS is the latest rendezvous algorithm with good performance which is based on the whole channel set. MTP is the latest rendezvous algorithm which is based on available channel set. For each algorithm F, we consider three types works: 1) The traditional F which ignores collision, we denote it as F-ignore collision; 2) The traditional F which considers collision, we denote it as F-consider collision; 3) Apply our protocol in conjunction with F, we denote it as F-CRP consider collision.

We consider the following key parameters: 1) The number of all channels $Q = 100$, 2) The ratio of number of available channels to the number of all channels $\theta (0 < \theta \leq 1)$. We randomly select $\theta Q$ channels for each user from the whole channel set, 3) The number of users $K (50 < K \leq 200)$. There are $\frac{K}{2}$ pairs of users trying to achieve rendezvous. For each set of parameter values, we perform 100,000 independent runs and then compute the average TTR.

6.4.1 Collision evaluation

In this subsection, we demonstrate that serious collision is caused to the rendezvous of one pair of users when there are multi-pair users. Figure [6.8] [6.9] [6.10] show the performance of the Random Algorithm, EJS and MTP respectively when $\theta = 0.5$ and $Q = 100$. It can be seen that: i) multiple pairs of users has no influence on the average TTR when the algorithms ignore collision. The average TTR is almost a constant when we vary $K$ from 50 to 200; and ii) When the algorithm consider collision, the average TTR increases as the number of users $K$ increases.
For example, when there are 150 users, the average TTRs of Random Algorithm, EJS and MTP are 98.59, 153.30 and 3315.83 respectively when they ignore collision. However, the average TTRs of Random Algorithm, EJS and MTP are 235.35, 319.89 and 8243.33 respectively when they consider collision. The increase is more than 108%. Figure 6.11 shows the increase ratio of average TTR of three algorithms in terms of $K$. The increase ratio is at least 16.50% and up to 219.38%. The collision is significant and can not be ignored.

### 6.4.2 Performance of Cooperative Rendezvous Protocol.

We now study the performance of the proposed Cooperative Rendezvous Protocol. We want to emphasize that the rendezvous in following simulation consider collision. That is, two users achieve rendezvous when they hop on the same channel and there are only these two users hop on this channel. In this part, we compare the average TTR under different values of $\theta$ and $K$.

Figure 6.12 shows the average TTRs of two kinds of Random Algorithm in terms of the number of users $K$ when we fix $\theta = 0.5$. That is, there are average 50% channels are available to each user. It can be seen that: 1) CRP has significant improvement on average TTR when we apply it in conjunction with Random Algorithm; 2)
Figure 6.9: Average TTR of EJS VS. $K$ when $\theta = 0.5$.

Figure 6.10: Average TTR of MTP VS. $K$ when $\theta = 0.5$. 
The average TTRs of both Random-consider collision and Random-CRP consider collision increase significantly when $K$ increases. For example, when there are 50 users, the average TTRs of Random-consider collision and Random-CRP consider collision are 130.02 and 59.85 respectively. When there are 150 users, the average TTRs of Random-consider collision and Random-CRP consider collision are 235.35 and 120.57 respectively. The increase ratio are 81.01% and 101.45% respectively.

Figure 6.13 shows the average TTR of two Random Algorithms in terms of the ratio of available channels $\theta$ when we fix $K = 150$. It can be seen that: 1) CRP has significant improvement on average TTR. Especially when $\theta$ is small (only a small part of channels are available to users). For example, when $\theta = 0.2$, the average TTRs of Random-consider collision and Random-CRP consider collision are 230.84 and 68.78 respectively. 2) When $\theta$ increases, the improvement of Random-CRP consider collision decreases. For example, when $\theta = 0.6$, the average TTRs of Random-consider collision and Random-CRP consider collision are 237.06 and 138.68 respectively. According to the analysis in Section 6.3.3, the improvement decreases when $\theta$ increases. This simulation result is consistent with the theoretically result.

Figure 6.14 and Figure 6.15 show the average TTRs of two kinds of EJS. It
Figure 6.12: Average TTR VS. $K$ when $\theta = 0.5$.

Figure 6.13: Average TTR VS. $\theta$ when $K = 150$. 
has the same features with Figure 6.12 and Figure 6.13. In addition, in Figure 6.15, we find that the average TTR of EJS-consider collision decreases while the average TTR of EJS-CRP consider collision increases when $\theta$ increases. That is because there is random-replacement in EJS algorithm. Our available channels are randomly selected. When $\theta$ increase, the number of commonly-available channels $G$ also increases. The random-replacement will bring significant benefit to the whole rendezvous process. The influence of commonly-available channels is significant than the influence of available channels. However, in our protocol, one user changes the available channel set to be the intersection between its original channel set and the available channel set of intended rendezvous user. When the number of commonly-available channels increases, the updated channel set will also increase. Therefore, the average TTR of EJS-CRP consider collision increases.

Figure 6.16 and Figure 6.17 show the average TTRs of two kinds of MTP. In Figure 6.16, when number of users $K$ increases, the average TTR of MTP-consider collision increases significantly while there is only a small increase for MTP-CRP consider collision. For example, when there are 50 users, the average TTRs of MTP-consider collision and MTP-CRP consider collision are 5494.55 and 1849.34 respectively. When there are 150 users, the average TTRs of MTP-consider collision
and MTP-CRP consider collision are 8242.33 and 2036.20 respectively. The increase ratio are 3456.90% and 10.10% respectively. Figure 6.17 has the same features with Figure 6.13 when the number of available channels increases, the average TTR of MTP-CRP consider collision increases significantly while there is only a small increase for MTP-consider collision. For example, when $\theta = 0.2$, the average TTRs of MTP-consider collision and MTP-CRP consider collision are 7006.49 and 1341.28 respectively. When we increase the $\theta = 0.9$, the average TTRs of MTP-consider collision and MTP-CRP are 7783.85 and 4568.64 respectively. The increase ratio are 11.10% and 240.62% respectively.

6.5 Chapter Summary

We evaluated the collision caused by multiple pairs of users. When there are 100 channels and the number of users varies from 50 to 200, the average TTR of Random Algorithm, EJS and MTP increases at least 16.50% and up to 219.38% compared with the average TTR when we ignore collision. It is significant and should not be ignored. We proposed a new protocol named Cooperative Rendezvous Protocol (CR-P) by which multiple users cooperate with each other to relay the available channel.
Figure 6.16: Average TTR VS. $K$ when $\theta = 0.5$.

Figure 6.17: Average TTR VS. $\theta$ when $K = 150$. 
set information. We theoretically analyze the average TTR when we apply CRP in conjunction with Random Algorithm and a general rendezvous algorithm which is based on available channel set. Simulation results showed that CRP can effectively counteract the collision and significantly improve the rendezvous performance, especially when the available channels is only a small part of all channels. The improvement is more significant when the number of available channels decreases.
Chapter 7

Conclusions

Cognitive radio network (CRN) is a promising networking paradigm in which unlicensed users with cognitive radios can opportunistically access idle channels in the spectrum of licensed users without causing interference. Rendezvous is a fundamental and essential operation for unlicensed users to meet on a commonly-available channel and establish a communication link on this channel, so that information exchange and data communication can be carried out. In this dissertation, we designed and analyzed rendezvous algorithms and protocol for cognitive radio networks.

7.1 Contributions

We made four major contributions in this dissertation:

1. We investigated the rendezvous problem in CRNs where cognitive users are equipped with multiple radios and different users may have different numbers of radios. Most of the existing works on rendezvous implicitly assume that each cognitive user is equipped with one radio (i.e., one wireless transceiver). We proposed a new rendezvous algorithm, called role-based parallel sequence (RPS) which specifically exploits multiple radios for more efficient rendezvous.

2. We designed a new rendezvous algorithm, called Interleaved Sequences based on Available Channel set (ISAC), that attempts rendezvous on the available channels only for faster rendezvous. ISAC constructs an odd subsequence and an even sub-sequence and interleaves these two subsequences to compose a CH sequence. It provides guaranteed rendezvous and gives significantly smaller MTTR than the existing algorithms.

3. We proposed a new rendezvous algorithm, called Adjustable Multi-Radio Rendezvous (AMRR). It combines the desirable features in previous two works
RPS and ISAC. It exploits multiple radios and generates CH sequences based on available channels for faster rendezvous. AMRR partitions all $R$ radios into two groups: $k$ stay radios and $R - k$ hopping radios. Among all existing multi-radio rendezvous algorithms, AMRR is the only adjustable one which can give its best performance on either MTTR or $E(TTR)$ with different allocations of radios.

4. We consider the collision in CRNs when there are multi-pair users. We evaluated the collision is significant and should not be ignored. We also proposed a new protocol, called Cooperative Rendezvous Protocol (CRP) by which multiple users cooperate with each other to relay the channel information to speed the rendezvous efficiently. CRP can be applied in conjunction with any existing rendezvous algorithm. We proved that CRP can decrease the expected time-to-rendezvous (TTR) efficiently when we apply it to an existing algorithm which is based on available channel set.

7.2 Possibilities for Future Work

As for future works, we will analyze the tight lower-bound of MTTR in using multiple radios, and improve the MTTR of the RPS and AMRR algorithm to achieve or be close to this tight lower-bound. Another future work is rendezvous algorithm based on available channels. Our ISAC has the additional requirement of two roles: sender and receiver. We want to explore an algorithm which can guarantee rendezvous based on available channels and has no additional requirement.

Another important future work is to make use of rendezvous to explore other problems in CRNs. To the best of our knowledge, there is very few research on broadcast algorithms in CRNs. Some existing algorithms implicitly assume that each user has two-hop information or the available channels of their neighbors. They are aimed at reducing the time slots to broadcast successfully. Some also consider the collision. But the cost is large. We know that the neighbor information is difficult
to obtain in CRNs before they build a link between each pair of users. What is more, the availability of channels and the location of users change with time. The cost is considerable to use an algorithm which needs the information of neighbors. The simplest way to broadcast without any information is “blind flooding”, i.e. each user forward the message after it receives from a father node. Flooding is not a good way in traditional networks. It results in serious collision and is extremely costly. There are many redundant rebroadcasts. This disadvantage is fatal in CRNs. A user needs long time to broadcast to all its neighbors even for one time. It should send message in each time slot and try to achieve rendezvous before it send the message to a potential neighbor. There is no negotiation between users. In this way there will be serious traffic problem in all channels in networks. And the neighbor is hard to receive this message accurately. The ideal way is to design a broadcast algorithm which does not need any information about neighbors and we aim at reducing the number of rebroadcasting nodes. It can decrease collision as a matter of fact. It is meaningful to design an algorithm about broadcast sequence to rebroadcast to all one-hop neighbors faster for one time.
Bibliography


Curriculum Vitae

Academic qualification of the thesis author, Ms. YU Lu:

- Received the degree of Bachelor of Computer Science from East China Normal University, June 2011.

August 2015