2017

Guaranteed Rendezvous for Cognitive Radio Networks Based on Cycle Length

Li Gou

Michigan Technological University, lgou@mtu.edu

GUARANTEED RENDEZVOUS FOR COGNITIVE RADIO NETWORKS BASED ON CYCLE LENGTH

By

Li Gou

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Computer Science

MICHIGAN TECHNOLOGICAL UNIVERSITY

2017

This report has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Computer Science.

Department of Computer Science

Report Co-advisor: *Dr. Min Song*

Report Co-advisor: *Dr. Xiaohua Xu*

Committee Member: *Dr. Bo Chen*

Department Chair: *Dr. Min Song*

Contents

List of Figures

List of Tables

Preface

This report is written based on the accepted conference paper [1]. The author of this report Li Gou is the first author of this conference paper, and is in charge of writing the paper, designing the algorithm, doing the analysis, and performing the simulation study. The second author Dr. Xiaohua Xu is in charge of improving the paper writing parts, and providing comments on Li's algorithm. Dr. Chongqing Zhang gave Li Gou some comments on the background setting of the rendezvous problem as well as for the simulations. Dr. Min Song provided many suggestions during the whole process of writing this paper, and improved the writing of the paper.

This report is written based on paper, but performs more systematic literature review, and presents more detailed illustration on the rendezvous problem. This report also gives more explicit solution to the rendezvous problem.

Acknowledgments

I would like to thank all those who have helped me during my master study period as well as those who helped me with LaTeX.

I appreciate Dr. Song and Dr. Xu, who helped me a lot on how to do research and write paper. I experienced hard time to get into the right way to do research, during which Dr. Song and Dr. Xu never forgot to encourage and help me. They guide me to the right way to read papers, formulate problem, design algorithm, perform analysis and improve the algorithm. Dr. Song always teaches me to be rigorous and creative in research, and helps me a lot on how to formulate a good problem, and justifying if a problem and the solution to the problem is good or not, how to analyze the results, as well as how to go back to improve the algorithm. I appreciate the help Dr. Xu gave in guiding me read and analyze newest good papers, as well as improving the algorithm and writings. With your thorough help I can improve the performance and efficiency of the algorithm. Thanks you very much, Dr. Song and Dr. Xu, for your support and help during my master study period.

Thanks for parents and my lovely friends! My parents, thanks so much for your selfless support and encouragement all the time. Thank you, all my nice friends, for accompanying and encouraging me.

List of Abbreviations

CRN	Cognitive Radio Networks
CH	Channel Hopping
DSA	Dynamic Spectrum Access
PU	Primary User
SU	Secondary User
CLR	Cycle Length Based Rendezvous
HH	Heterogeneous Hopping
ETTR	Expected Time to Rendezvous
MTTR	Maximal Time to Rendezvous
ID	Identifier of the radio node

Abstract

Rendezvous is a fundamental process establishing a communication link on common channel between a pair of nodes in the cognitive radio networks. How to reach rendezvous efficiently and effectively is still an open problem. In this work, we propose a guaranteed cycle lengths based rendezvous (CLR) algorithm for cognitive radio networks. When the cycle lengths of the two nodes are coprime, the rendezvous is guaranteed in $T_i * T_j + \delta$ time slots, where T_i and T_j are two prime numbers representing the cycle length of nodes i and j respectively, and $\delta \in [0, T_i)$ is the time skew under asynchronous scenario. When T_i and T_j are not coprime, i.e., $T_i = T_j$, the deadlock checking and node IDs are combined to decide the time point and the way to independently change the cycle length on each node to guarantee rendezvous. In detail, as long as the deadlock situation is detected at $t = T^0_{max} + k * T^1_{max}$, each node can independently oscillate its cycle length between T^0_i and T^1_i based on the k+1-th bit of the node ID, where $k = 0, 1, \cdots, \lceil \log max\{ID_i, ID_j\} \rceil - 1$, and T^0_i, T^1_i are two prime numbers defined for T_i. $T^0_{max} = T_i * T_c + T_c$ (T_c is some constant) and $T^1_{max} = T^0_i * T^1_i + T_c$ are two thresholds used for deadlock checking, which represents the length of the maximum possible rendezvous period between the two nodes. As long as the current checking bits between the two nodes are different, the rendezvous will be reached in the following rendezvous period, guaranteed in $T^0_{max} + \lceil \log max\{ID_i, ID_j\} \rceil * T^1_{max}$ time slots. The theoretical analysis also proves the guarantee of the CLR algorithm

under both the two cases. We use three metrics: success rate of rendezvous, expected time to rendezvous and channel load to conduct simulation studies. The simulation results show that the CLR algorithm always has higher successful rendezvous rate of 100%, and stable and low expected time to rendezvous compared to the HH algorithm. In addition, the channel loads are smoothly distributed on all channels with CLR, while HH algorithm depends on the channels with smaller IDs.

Chapter 1

Introduction

With the increase of spectrum demanding on various services and applications, the spectrum scarcity remains a critical problem for wireless communications. Dynamic Spectrum Access (DSA) was then proposed with cognitive radio networks (CRNs). The DSA technique specifies a 3-tiered spectrum sharing framework [2, 3] that improves the efficiency in usage of licensed spectrum. With DSA, secondary users (SUs) can dynamically detect and access the idle licensed spectrum used by primary users (PUs). However, the dynamics of PUs' activities force SUs to adapt to the variations in channel availability. Thus, finding common available channel among the SUs on demand is the first task before communicating. The process that the SUs establish the communication link on the common channels is called *rendezvous*.

Common control channel based rendezvous schemes [4, 5, 6] use the predetermined common control channel to coordinate the rendezvous among the nodes on demand. In detail, all the nodes wishing to establish a communication link with the other nodes firstly need to communicate on the common control channel to exchange the control information and further find the common channels between them. Although this method simplifies the rendezvous process, the heavy traffic load on the common control channel causes network congestion and requires high overhead cost. To overcome these issues, blind rendezvous algorithms based on channel hopping (CH) were then proposed [7, 8, 9]. The JS rendezvous algorithms proposed in [10, 11] achieve guaranteed rendezvous by constructing CH sequences combining periods of jump-pattern and stay-pattern. But JS assumes that the nodes in the CRN have the same number of available channels even under asymmetric model. The SeR algorithm proposed in [12] was guaranteed in asynchronous environment by combining the parity slots and permutations of the available channels, but it is only applicable to the symmetric model. In [8], the SYNC-ETCH algorithm constructing $2N-1$ different CH sequences, with intersections on different channel over each two of the CH sequences, was proposed for synchronous scenario, and the improved SeR algorithm (ASYNC-ETCH) is used to address the asynchronous rendezvous problem. However, ETCH algorithms were only for symmetric models. The probability based CH algorithm proposed in [13] depends on the preassigned order of the node IDs in the network to further assign roles to each node, which is not able to give the upper bound of the time to rendezvous (TTR). The asynchronous quorum based rendezvous algorithm utilizing the cyclic rotation closure property of the cyclic quorum system is proposed in [14], in which the rendezvous can only be reached on two of the available channels, and its asynchronous Latin square based algorithm can only be applied to symmetric model.

To address the assumption made by the asymmetric model that different radio in the CRN has same number of different available channels, the heterogeneous model based rendezvous algorithms [7, 15, 16] for heterogeneous CRN were proposed. In heterogeneous model, different radio has different spectrum sensing capability. It is clear that the heterogeneous model is more realistic than the asymmetric model, where the available channel set between any two nodes are different in both length and the range of the available channels. The heterogeneous hopping (HH) algorithm in [7] assumed that the labels of the available channels of each node are consecutive, by which the smallest channel ID was chosen as the rotation number and the channel with the smallest ID as the parity channel. The performance of HH depends on the channel IDs significantly. In [17], the rendezvous scheme for two-channel scenario was firstly proposed, which is then applied to fully available spectrum scenario and partially available spectrum scenario by using the TP and MTP respectively. But two problems exist in this paper, firstly, the labels of the globally available channels is a consecutive number sequence. Secondly, as observed in [15], the performance of MTP is not highly efficient even if the spectrum is fully available. The DSCR algorithm in

[15] is based on a set of globally labeled available channels, so the same channel label (or index) of a channel is used between a pair of nodes. In other words, they do not consider the oblivious setting for channel labeling. Here, the definition of oblivious channel labeling is available in [16]. It is clear that the algorithms proposed in [17] is also non-oblivious.

In this project, we propose an oblivious guaranteed rendezvous algorithm called cycle length based rendezvous (CLR) algorithm. The CLR algorithm guarantees rendezvous no matter T_i and T_j are coprime or not under heterogeneous model and asynchronous scenario considering oblivious channel labeling. To guarantee rendezvous when the cycle lengths of two nodes (T_i and T_j) are not coprime, we introduce threshold T_{max} based deadlock checking and node ID based cycle length change mechanisms. The threshold T_{max} is defined as the length of the maximal possible rendezvous period of the two nodes. There are two possible values: i) $T_{max}^0 = T_i * T_c$ for the first deadlock checking, where T_c is the maximum possible cycle length of the nodes in the CRN calculated by the minimum prime number no less than C (C is the maximal channel sensing ability of all the radios in the CRN). ii) $T_{max}^1 = T_i^0 * T_i^1 + T_c$ for the future checking, where T_i^0 and T_i^1 are two prime numbers defined for T_i. If rendezvous between the two nodes is not reached after hopping T_{max}^0 time slots (first deadlock checking), the two nodes will independently change its cycle length by checking the first bit of its binary ID. The future deadlock checking using T_{max}^1 will be combined with the other bits of node IDs to conduct cycle length change. The rendezvous will be reached as long as the current checking bits between the two nodes are different, where $T_i \neq T_j$ is met. The CLR has the following features:

1. When T_i and T_j are not coprime ($T_i = T_j$), it applies threshold based deadlock checking and node ID based cycle length change to guarantee rendezvous in $T^0_{max} + \lceil \log max\{ID_i, ID_j\} \rceil * T^1_{max}$ time slots.

2. The CLR algorithm is proved to be guaranteed under all the possible time skew $\delta \in [0, T_i)$ (asynchronous system) via both theoretical analysis and simulation study.

3. The CLR algorithm is able to run on the heterogeneous model with oblivious channel labeling. The performance of CLR algorithm is totally independent with the channel IDs, and channel loads are smoothly distributed on each channel.

The rest of the report is organized as follows. The system model and problem formulation are provided in Chapter 2. Chapter 3 presents the details of the proposed CLR algorithm and the theoretical analysis. The simulation results are presented in Chapter 4. Chapter 5 reviews the existing work related to the rendezvous problem in CRNs. We conclude our work in Chapter 6.

Chapter 2

System Model and Problem Formulation

In this chapter, we will firstly introduce the rendezvous problem, and then present the system model and problem formulation. Table 2.1 summarizes the notations used.

2.1 Rendezvous Problem

In CRNs, the spectrum availability of each node is dynamic due to PUs' prioritized activities. Thus, the rendezvous process that the two nodes find common available channels between them is the first critical step before the formal communications. The channel hopping technique is the most commonly used method for blind rendezvous between any two nodes in CRNs.

In channel hopping scheme, the time is slotted with the same and fixed length. Each node independently builds a CH sequence on its available channel set by following the corresponding CH sequence generation algorithm. When the node wants to start a communication, it attempts rendezvous with the other node by hopping on its CH sequence. In detail, at each time slot, the node hops on one of the channel in its CH sequence, as long as the two nodes wish to rendezvous hop on the same channel at the same time slot, the rendezvous between them is reached. RTS/CTS techniques can be used to detect if the two nodes are on the same channel.

Fig. 2.1 shows an illustration on how the two nodes reach rendezvous by using channel hopping scheme. Node i hops on the CH sequence $\{1, 2, 3, 5, 8, 9\}$ round by round until the rendezvous is reached with node j who hops on $\{3, 4, 6, 8, 10, 12, 13\}$. The shaded slot shows nodes i and j reach rendezvous at time slot $t = 11$.

t	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
node i	1	2	3	5	8	9	1	2	3	5	8	9	1	2	3	5	8	9
node j	3	4	6	8	10	12	13	3	4	6	8	10	12	13	3	4	6	8

Figure 2.1: Illustration of the rendezvous process with available channels of nodes: $C_i = \{1, 2, 3, 5, 8, 9\}$, $C_j = \{3, 4, 6, 8, 10, 12, 13\}$.

Table 2.1
NOTATION

Variable	Description
C_i	The set of available channels of node i.
c_i	The i-th channel in the C_i of node i.
C	The maximum sensing capability of the radios in the CRN.
T_c	The maximal possible cycle length of nodes in the CRN, calculated by the minimum prime number no less than C.
T_{max}	The threshold for deadlock checking, defined as the length of the maximal rendezvous period.
B_i	The binary ID, binary representation of the ID of node i.
b_l^i	The $l-th$ bit of B_i
T_i	The cycle length of node i.
S_i	The CH sequence of node i.
$S_i^{(t)}$	The channel hopped by node i at time slot t.
$S_i^{[x]}$	The x-th round of the CH sequence of node i.
$S_i^{[x,y]}$	The y-th channel in the x-th round of the CH sequence.
δ	The time skew between CH sequences of nodes i and j.
T	The length of the rendezvous round between the CH sequences of nodes i and j.

2.2 System Model

We consider a CRN of multiple nodes. Each node has a unique ID, denoted as ID_i, for example, the MAC address of the radio can be used as its ID. The binary representation of the ID_i, called binary ID of node i, is defined as $B_i = \{b_i^1, b_i^2, \cdots, b_i^{W_i}\}$, where $b_i^k \in \{0,1\}$ is the k-th bit of B_i and W_i is the width or number of bits of B_i. We assume ID_i is in decimal format, so $W_i = \lceil \log ID_i \rceil$. In addition, the bit sequence of B_i is in the order of least significant to most significant bit. For any two nodes i and j in the CRN, $B_i \neq B_j$, so there is at least one different bit in their binary IDs.

In this report, we only discuss the two-node scenario. Each node is equipped with a single radio. Let $C_i = \{c_0, c_1, \cdots, c_{|C_i|-1}\}$ be the set of channels available to node i, and $|C_i|$ is the total number of available channels. Assume that the channels available to each node cause no interference to any PU. We consider heterogeneous model, thus C_i and C_j can be different in both total number of available channels and range of the available channels, and we have $C_i \bigcap C_j \neq \emptyset$. Assume that all the radios in the CRN have the same *maximal sensing capability*, denoted as C, thus for any node i, $|C_i| \leq C$.

To attempt rendezvous, the radio of each node hops on one channel at each time slot. The rendezvous is reached when the radios of the two users hop on the same channel at the same time slot. The number of time slots required to rendezvous is defined as Time to Rendezvous (TTR). The Maximal Time to Rendezvous (MTTR) is the TTR in the worst case, and the Expected Time to Rendezvous (ETTR) is the average TTR over different cases. The rendezvous *period* is defined as a period where the hopping and alignment of the CH sequences of the two nodes i and j repeat. Let the length of the rendezvous period be T. The length of rendezvous period between nodes i and j in Fig. 2.2(a) is 15, where nodes i and j repeat the hopping and the alignment from $t = 1$ to $t = 15$ at $t = 16$. The *deadlock* in rendezvous is defined as the situation where the two nodes can not rendezvous with each other forever, and in this report we decides deadlock situation by a certain amount of time, the threshold T_{max}, which is the length of the maximal rendezvous period.

2.3 Problem Formulation

We consider an asynchronous system in designing the algorithm. The time skew is defined as the number of time slots the CH sequences of the two nodes misaligned, so any time skew is possible in the asynchronous system. It should be noted that we only consider the slot-aligned CH sequences in this report. Let δ be the time skew between the CH sequences of the two nodes i and j, $\delta \geq 0$. Without loss of generality, we assume node i always starts hopping earlier than node j if $\delta > 0$. That is, if node i starts hopping at time slot t, then node j starts at $t + \delta$, and δ is in the range: $\delta \in [0, T_i)$.

Let the CH sequence of node i be $S_i = \{S_i^{(0)}, S_i^{(1)}, \cdots, S_i^{(T_i-1)}, S_i^{(T_i)}, \cdots, S_i^{(2*T_i-1)}, S_i^{(2*T_i)}, \cdots\}$, where $S_i^{(t)}$ represents the channel hopped at time slot t, and T_i is a prime number representing the cycle length of the CH sequence. The cycle length T_i is defined as the minimum prime number that is no less than $|C_i|$. The CH sequence can also be defined in cycles as: $S_i = \{S_i^{[0]}, S_i^{[1]}, S_i^{[2]}, \ldots\}$, where $S_i^{[x]}$ is the x-th cycle. For each cycle of the CH

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
node i	3	6	8	5	7	3	6	8	5	7	3	6	8	5	7
node j	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1

(a) Synchronous situation, $\delta = 0$.

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
node i	3	6	8	5	7	3	6	8	5	7	3	6	8	5	7	3	6	8	5
node j					3	1	2	3	1	2	3	1	2	3	1	2	3	1	2

(b) Asynchronous situation.

Figure 2.2: Rendezvous processes on synchronous and asynchronous scenarios with $C_i = \{3, 5, 6, 7, 8\}$, $C_j = \{1, 2, 3\}$. a) Time skew $\delta = 0$. b) Time skew $\delta = 4$.

sequence, $S_i^{[x]} = \{S_i^{[x,0]}, S_i^{[x,1]}, \ldots, S_i^{[x,T_i-1]}\}$, where $S_i^{[x,y]}$ denotes the y-th channel at the x-th cycle. Thus the time slot when radio i hopes on channel $S_i^{[x,y]}$ is $t = x * T_i + y$.

As shown in Fig. 2.2(b), the cycle length of nodes i and j are $T_i = 5$, $T_j = 3$, and their CH sequences are $S_i^0 = \{3, 6, 8, 5, 7\}$, $S_j^0 = \{3, 1, 2\}$. The time skew between two CH sequences can be any value between $[0, 5)$.

The rendezvous problem of finding the time slot the rendezvous is reached between nodes i and j can be defined as follows:

$$S_i^{(t)} = S_j^{(t-\delta)} \tag{2.1}$$

where t is the TTR for node i, and $t - \delta$ for node j. Then the maximal time slots required for rendezvous between the two nodes can be defined as $\texttt{MTTR} = max_{\forall\delta} t$.

Chapter 3

Cycle Length based Rendezvous Algorithm

In this chapter, we will present the cycle length based rendezvous (CLR) algorithm, as well as the theoretical analysis on the performance of the CLR algorithm.

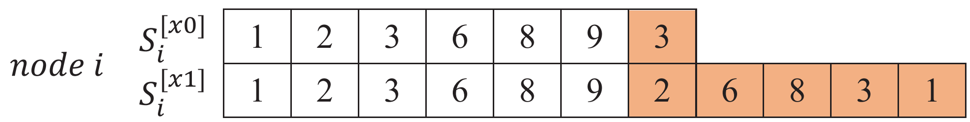
3.1 The CLR Algorithm

The CH sequence of the proposed CLR algorithm is constructed based on cycle length. For each node i, we define two cycle lengths as follows: i) T_i^0, the minimum prime number no less than $|C_i|$ (function $nextPrime(|C_i| - 1)$ in Algorithm 1). ii) T_i^1, the smallest prime number greater than T_i^0 (function $nextPrime(T_i)$ in Algorithm 1). During the rendezvous process, each node will oscillate its cycle length T_i between $\{T_i^0, T_i^1\}$ based on the value of each bit of its binary ID. Based on the two possible cycle lengths, the two CH sequences for each node are constructed in cycle as following:

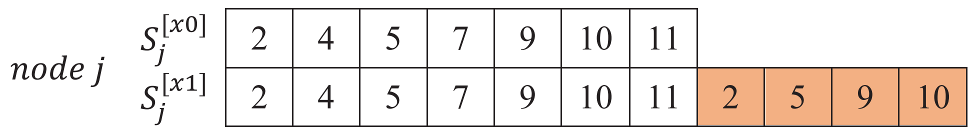
$$\begin{cases} S_i^{[x0]} = \{c_0, c_1, \cdots, c_{|C_i|}, c_{z1}, c_{z2}, \cdots, c_{z(T_i^0 - |C_i|)}\} \\ S_i^{[x1]} = \{c_0, c_1, \cdots, c_{|C_i|}, c_{z1}, c_{z2}, \cdots, c_{z(T_i^1 - |C_i|)}\} \end{cases} \tag{3.1}$$

where $c_{zj}, j \in \{1, 2, \cdots\}$ is an arbitrary channel in C_i. At each cycle x, the node i will choose one of its two CH sequences based on its cycle length. If $T_i = T_i^0$, $S_i^{[x]} = S_i^{[x0]}$; if $T_i = T_i^1$, $S_i^{[x]} = S_i^{[x1]}$. Fig. 3.1 shows an illustration on how to build the CH sequences of each node based on the two cycle lengths. It is clear that, the CH sequence generated by our algorithm is totally independent to the channel IDs and is oblivious.

We consider two cases based on the cycle length of each node when design CLR algorithm: i) T_i and T_j are coprime, *i.e.*, $T_i \neq T_j$, in which the rendezvous is guaranteed at the first rendezvous period within $T_i * T_j + \delta$ time slots. ii) T_i and T_j are not coprime, *i.e.*, $T_i = T_j$, where rendezvous may not be reached within the first rendezvous period. For the second case, a threshold T_{max}, which is the length of the maximum possible rendezvous period between the two nodes wishing to rendezvous, is used to check the



(a) The two CH sequences of node i.



(b) The two CH sequences of node j.

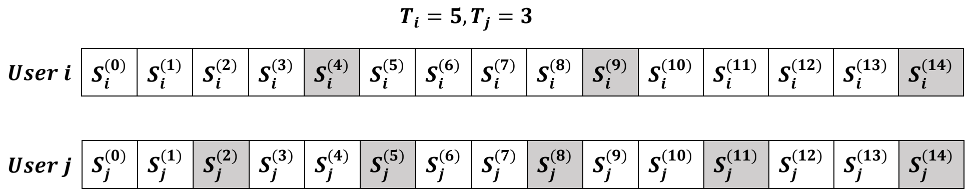
Figure 3.1: Illustration on generating the two cycle lengths and the corresponding CH sequences of each node. (a) For node i, $C_i = \{1, 2, 3, 6, 8, 9\}$, thus $T_i^0 = 7, T_i^1 = 11$. (b) For node j, $C_j = \{2, 4, 5, 7, 9, 10, 11\}$, thus $T_j^0 = 7, T_j^1 = 11$. The channels in shaded slots are randomly selected from C_i or C_j.

deadlock situation. We consider two cases for the definition of the threshold T_{max}: i) $T_{max}^0 = T_i * T_c + T_c$ used at the first deadlock checking, where T_c is the maximal possible cycle length of the nodes in the same CRN, *i.e.*, $T_i <= T_c$, for any node i; ii) $T_{max}^1 = T_i^0 * T_i^1 + T_c$ used for the future deadlock checking. Considering the time skew between the two nodes, T_c is added into both the two thresholds. For the first deadlock checking occurs at $t = T_{max}^0$ (the rendezvous between nodes i and j is not reached after hopping T_{max}^0 time slots), the equality between T_i and T_j will be learned by both the two nodes attempting rendezvous. Then each node will independently update its cycle length between T_i^0 and T_i^1 based on the first bit of their binary IDs B_i. For the future deadlock checking occurs at $t = T_{max}^0 + k * T_{max}^1, k = 1, 2, \cdots, max\{W_i, W_j\} - 1$, each node will independently update its cycle length based on the other bits of their binary IDs until rendezvous is reached.

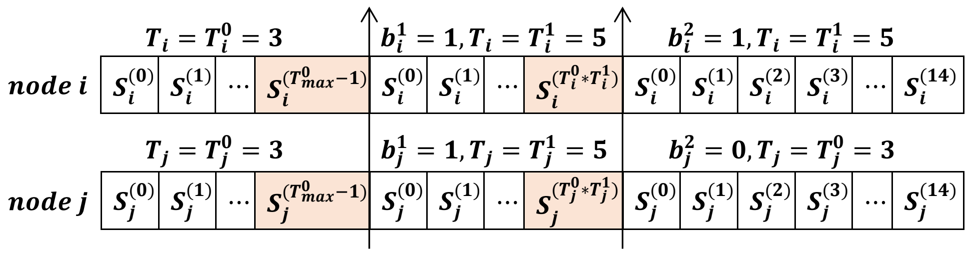
For nodes i and j, based on the definition of the binary ID, $\exists k, k \in$

$\{1, 2, \cdots, max\{W_i, W_j\}\}$, such that, $b_i^k \neq b_j^k$. Let $l \in \{2, \cdots, max\{W_i, W_j\}\}$, the following two situations exist during the deadlock checking on bit b_i^l: i) current checking bit $b_i^l = 0$, node i sets its cycle length to $T_i = T_i^0$; and ii) $b_i^l = 1$, node i sets its cycle length to $T_i = T_i^1$. Then node i hops on the CH sequence generated by Eq. 3.1 with the new cycle length. If the rendezvous is not reached within $T_{max}^1 = T_i^0 * T_i^1 + T_c$ time slots, deadlock situation occurs ($b_i^l = b_j^l, T_i = T_j$), node i will check the next bit $b_i^{(l+1)}$. Otherwise, the rendezvous will be guaranteed during this rendezvous period ($b_i^l \neq b_j^l, T_i \neq T_j$). It should be noted that if the rendezvous is not reached until all the bits in B_j has been checked, then $B_j^{l'} = 0$ when $l' > W_j$, if $W_i > W_j$; vice versa.

Fig. 3.2 shows the rendezvous process described above. When $t < T_i * T_c$, node i hops on its CH sequence with $T_i = T_i^0$ until the rendezvous is reached (Fig. 3.2(a)), otherwise, the deadlock situation will be encountered (first period of Fig. 3.2(b)). When $t \geq T_i * T_c$, the two nodes will hop on its CH sequence with the selected cycle length and independently conduct deadlock checking based on threshold T_{max}. The cycle length will be oscillated between T_i^0 and T_i^1 based on the binary ID as the deadlock is detected. As long as the current checking bits between nodes i and j are different, the cycle lengths between them will also be different, and the rendezvous can be reached in the following rendezvous period, as shown in the third period of Fig. 3.2(b).



(a) T_i and T_j are coprime, $T_i \neq T_j$.



(b) T_i and T_j are coprime, $T_i = T_j$.

Figure 3.2: Illustration of the rendezvous process when $\delta = 0$ and $T_i^0 = 3, T_i^1 = 5; T_j^0 = 3, T_j^1 = 5$. (a) When $T_i \neq T_j$, the rendezvous is guaranteed within $T_i * T_j = 15$ time slots. (b) When $T_i = T_j = 3$, they may not rendezvous within $T_i * T_c$ time slots (deadlock situation). At time slot $t = T_{max}^0$, node i with $b_i^1 = 1$ changes its cycle length to $T_i = T_i^1 = 5$, while node j with $b_j^1 = 1$ changes to $T_j = T_j^1 = 5$. The two nodes may still not rendezvous within this rendezvous period ($T = T_i^0 * T_i^1 = 15$). Continue checking on the second bit at $t = T_{max}^0 + T_{max}^1$: node i with $b_i^2 = 1$ changes its cycle length to $T_i = T_i^1 = 5$, while node j with $b_j^2 = 0$ changes to $T_j = T_j^0 = 3$. The rendezvous between them will be guaranteed during the following 15 time slots.

In Algorithm 1, the function $toBinary(ID_i)$ is used to convert the ID_i to binary format. The function $minPrime(\cdot)$ will return the next prime number greater than the parameter. In line 2, the two cycle lengths are initialized, and the two corresponding CH sequences for node i are generated in line 3. The T_i^0 and $S_i^{[x0]}$ are initially assigned to node i in line 4. In line 5, the two threshold used to do deadlock checking is defined. For the deadlock checking process from line 8 to 24, lines 9-14 are the checking on the first bit, while lines 15-22 are the future checking on the other bits. The rendezvous is attempted on each channel of the CH sequence in lines 25-31.

Algorithm 1 CLR Algorithm

Input: Available channels set $C_i = \{c_0, c_1, \cdots, c_{|C_i|-1}\}$, the threshold T_c, node ID ID_i, time skew δ

Output: Rendezvous channel c_k and the TTR t

$B_i = toBinary(ID_i)$;
$T_i^0 = nextPrime(|C_i| - 1)$; $T_i^1 = nextPrime(T_i^0)$;
Generate $S_i^{[x0]}$ and $S_i^{[x1]}$ according to Eq 3.1.
$T_i = T_i^0$; $S_i^{[0]} = S_i^{[x0]}$;
$T_{max}^0 = T_i * T_c + \delta$; $T_{max}^1 = T_i^0 * T_i^\delta$;
$t = 0, l = 1, t_C = 0$;
while $true$ **do**
    if $(t \geq T_{max}^0)$ **then**
        if $(t = T_{max}^0)$ **then**
            $l = l + 1$;
            if $(b_i^l = 1)$ **then**
                $T_i = T_i^1$; $S_i^{[0]} = S_i^{[x1]}$;
            end if
        end if
        if $(t_C \geq T_{max}^1)$ **then**
            $t_C = 0, l = l + 1$;
            if $(b_i^l = 1)$ **then**
                $T_i = T_i^1$; $S_i^{[0]} = S_i^{[x1]}$;
            else if $(b_i^l = 0)$ **then**
                $T_i = T_i^0$; $S_i^{[0]} = S_i^{[x0]}$;
            end if
        end if
        $t_C = t_C + T_i$;
    end if
    for $t' = 0$ to $T_i - 1$ **do**
        if (Successfully rendezvous on channel $S_i^{[0,t']}$) **then**
            $c_i = S_i^{[0,t']}$;
            return c_i and t
        end if
        $t = t + 1$;
    end for
end while

3.2 Performance Analysis

Here we will analyze the performance of the CLR Algorithm by considering all the possible time skew $\delta \in [0, T_i)$ between the CH sequences of nodes i and j.

Lemma 1 For nodes i and j, with $\delta = 0$ and $T_i \neq T_j$, let the length of the rendezvous period be $T = T_i * T_j$. For any two CH sequences of nodes i and j, if no rendezvous can be reached within T time slots, then there is no guaranteed rendezvous between the two nodes (deadlock situation). Otherwise $\texttt{MTTR} = T$.

Proof: We prove the lemma 1 by contradiction. Assume nodes i and j rendezvous at time slot $T + k$ after the first rendezvous round, $0 \leq k < T$. Within a rendezvous round (T time slots), node i hops for $r = T/T_i$ cycles, and node j for $s = T/T_j$ cycles. Thus, the last hop by nodes i and j at each rendezvous round, $S_i^{(m*r*T_i-1)}$ and $S_j^{(m*s*T_j-1)}$, are always aligned with each other, where m is the number of rendezvous rounds. So at time slot $t = m * r * T_i = m * s * T_j$, both nodes i and j repeat the hopping of the former rendezvous round with same CH sequence and same alignment. Therefore, if no rendezvous is reached in the first cycle, there is no rendezvous in future cycles. This is a contradiction with the assumption.

Theorem 1 The rendezvous between nodes i and j is guaranteed in $\texttt{MTTR} = T_i^0 * T_c + T_c + \lceil \log max\{ID_i, ID_j\} \rceil * (T_i^0 * T_i^1 + T_c) = O(T_i * T_c)$ time slots, considering all the possible time skew $\delta \in [0, T_c)$ between their CH sequences.

Proof: There are four cases:

Case 1 $T_i \neq T_j,\ \delta = 0$. This case is proved by Theorem 6 of [7], and $\texttt{MTTR} = T_i * T_j$. This report mainly focuses on the other three cases.

Case 2 $T_i \neq T_j,\ \delta > 0$, where node i starts δ time slots earlier than node j. Assume that node i starts hopping at time slot $t = 0$, so node j starts hopping at time slot δ. Let $S_i' = \{S_i^{(\delta)}, S_i^{(\delta+1)}, \cdots, S_i^{(T-1)}, S_i^{(0)}, S_i^{(1)}, \cdots, S_i^{(\delta-1)}\}$ be another CH sequence for node i. It is clear that S_i' is a clockwise rotation of the original CH sequence of node $i:\ S_i = \{S_i^{(0)}, S_i^{(1)}, \cdots, S_i^{(\delta-1)}, S_i^{(\delta)}, S_i^{(\delta+1)}, \cdots, S_i^{(T-1)}\}$, by δ time slots. So the rendezvous between S_i and S_j with δ time skew is equivalent with the rendezvous between S_i' and S_j with *zero* time skew. Thus this case is equivalent to the case that both nodes i and j start hopping at time slot $t = 0$ ($\delta = 0$), which is exactly the case 1. So the rendezvous of this case is still guaranteed, and $\texttt{MTTR} = T_i * T_j + \delta$.

Case 3 $T_i = T_j, \delta = 0$. When nodes i and j have the same cycle length, the length of the rendezvous round is $T = T_i = T_j$, and there is a high probability that no rendezvous can be reached in T time slots. Without the threshold, nodes i and j may not rendezvous with each other forever according to Lemma 1. For the deadlock checking process starting at $t = T_i * T_c$, the worst case is when $b_i^{1,2,\cdots,W_i} = 1$ and $B_j^{1,2.\cdots,W_i-1} = 1, b_j^{W_i} = 0$, where $W_j = W_i - 1$ (assuming $W_i > W_j$). In this case, the deadlock checking process will repeat for $W_i - 1$ times, and the rendezvous can only be reached on the last bit. Based on Algorithm 1, both the deadlock checking process on each of the $W_i - 1$ bits and rendezvous process at the last bit requires $T_i^0 * T_i^1$ time slots. Thus, $\texttt{MTTR} = T_i^0 * T_c + \lceil \log max\{ID_i, ID_j\} \rceil * (T_i^0 * T_i^1)$.

Case 4 $T_i = T_j,\ \delta > 0$. Based on cases 2 and 3, it's easy to find that the rendezvous of this case is still guaranteed, and $\texttt{MTTR} = T_i^0 * T_c + T_c + \lceil \log max\{ID_i, ID_j\} \rceil * (T_i^0 * T_i^1 + T_c)$.

Chapter 4

Performance Evaluation

In this section, we will verify the theoretical results in Theorem 1 by comparing with the simulation results. For comparison purpose, the HH algorithm is also simulated. The comparisons are conducted by the following three metrics: i) guarantee of the rendezvous; ii) average/expected time to rendezvous (ETTR); and iii) channel load.

4.1 Simulation Setup

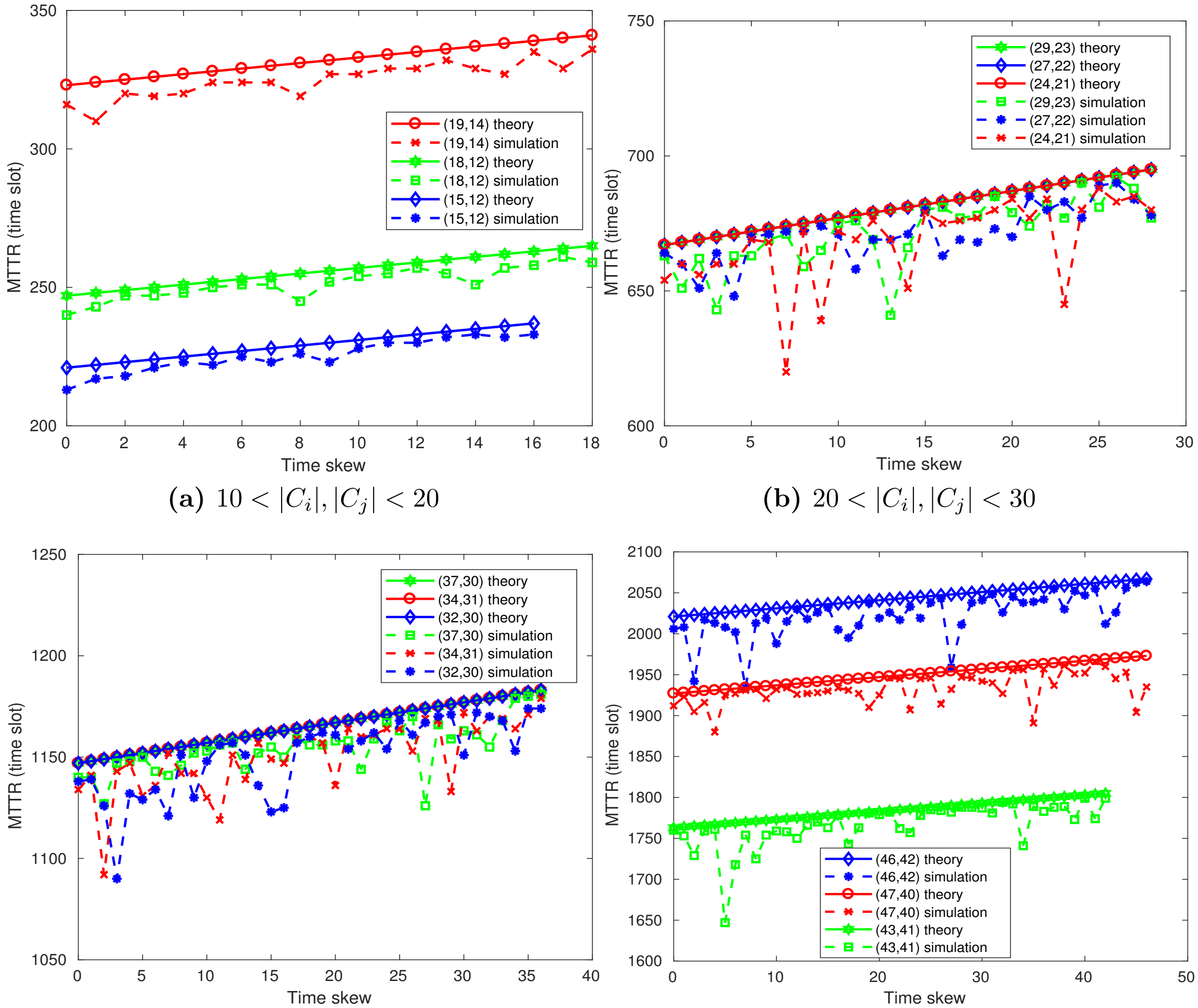
The simulations are implemented in MATLAB R2016a in asynchronous environments. The asynchronous environment is guaranteed by varying the time skew between CH sequences of nodes i and j in range $[0, T_i)$ on all possible values. Here we assume

$T_i >= T_j$. The simulations are conducted in 2-node scenario, and the available channels of the two nodes are randomly selected from $[1, 100]$. The common channels between them are also randomly selected from their available channel set.

We run simulations by varying the number of available channels of the two nodes in the following four periods with the corresponding threshold for CLR: i) $10-20$ period ($C = 20, T_c = 23$); ii) $20-30$ period ($C = 30, T_c = 31$); iii) $30-40$ period ($C = 40, T_c = 41$); and iv) $40-50$ period ($C = 50, T_c = 53$). For each period, we randomly select three pairs of values as the number of available channels for nodes i and j, as shown in Figs. 4.1 and 4.2. For each situation, the time skew between nodes i and j is varied in range $[0, T_i)$ and the number of common channels varied in $[1, |C_j|]$, and each run is repeated for 100 times. So the results in the simulations are got by combing $100 * T_i * |C_j|$ runs. The overlapping ratio is defined as the fraction of common channels between nodes i and j to the total number of available channels of node j, assuming $|C_i| > |C_j|$. During the simulation, we randomly generate a 8-bit binary ID for each node at each run.

4.2 Verification on the Theoretical Results

In this part, we will verify the theoretical results concluded in Theorem 1 by comparing with the simulation results in two situations when T_i and T_j are coprime ($T_i \neq T_j$)



(a) $10 < |C_i|, |C_j| < 20$

(b) $20 < |C_i|, |C_j| < 30$

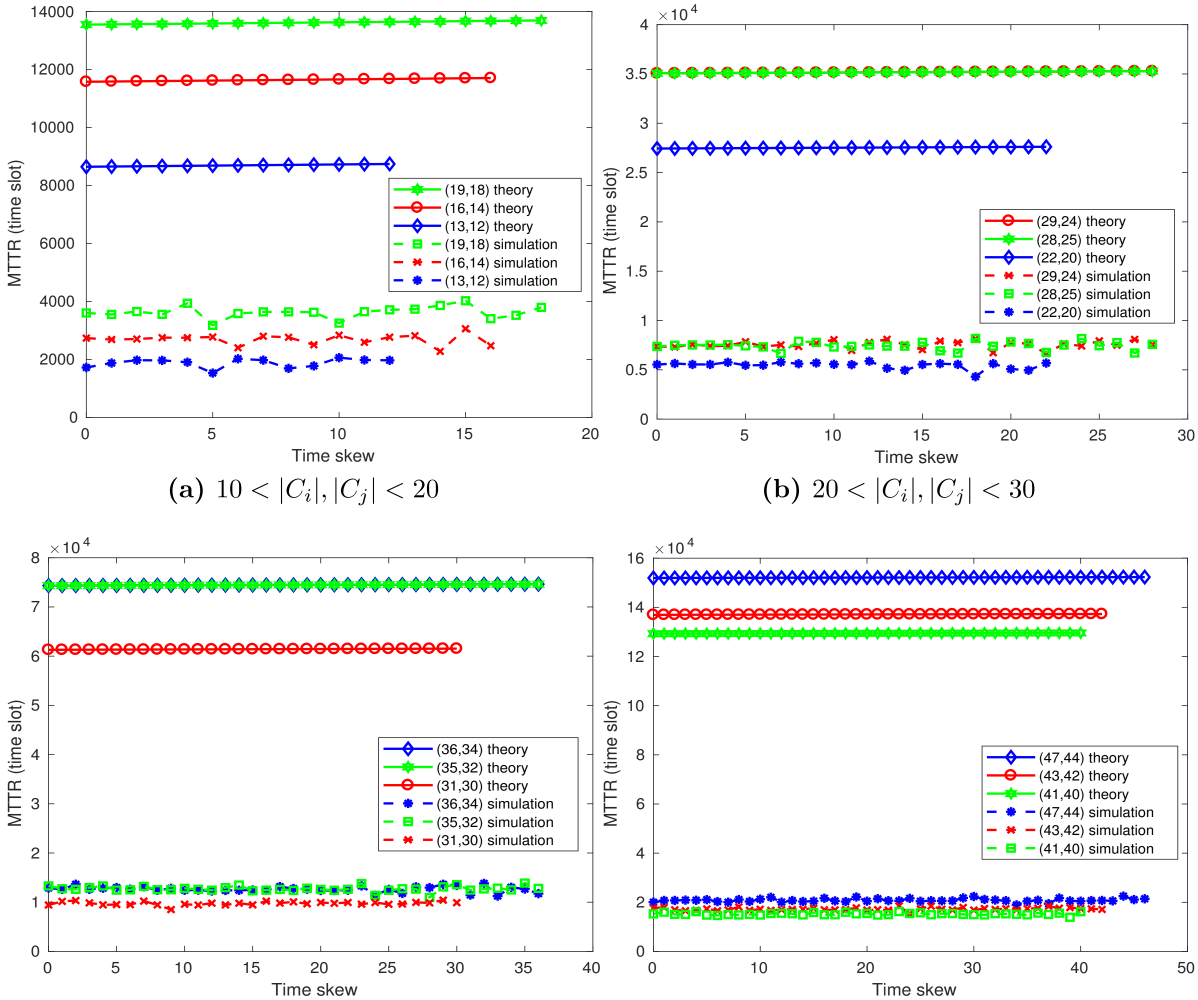
(c) $30 < |C_i|, |C_j| < 40$

(d) $40 < |C_i|, |C_j| < 50$

Figure 4.1: Theoretical and simulated MTTR when T_i and T_j are coprime under varying time skew and different pair number of available channels of nodes i and j.

or not ($T_i = T_j$). We study the MTTR under varying time skew and different pairs of number of available channels of nodes i and j.

At Fig. 4.1(b), the selected three pairs of $|C_i|$ and $|C_j|$ correspond with the same pair of $T_i = 29$ and $T_j = 23$. Thus, their theoretical MTTRs are totally same under all time skew, which is consistent with Theorem 1. For the simulated MTTRs, the small



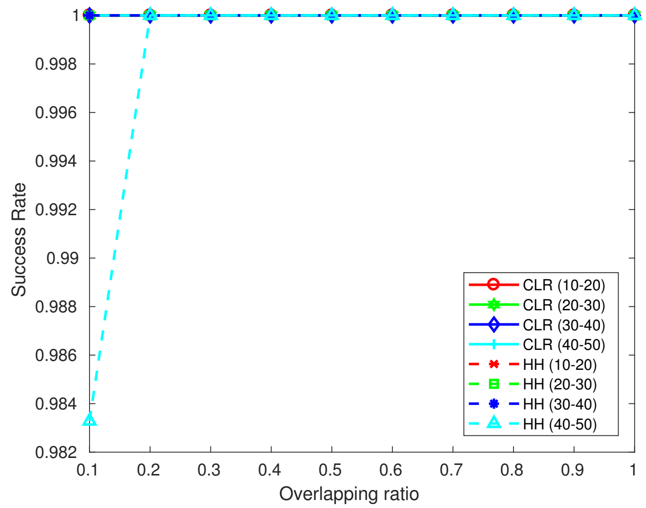
(a) $10 < |C_i|, |C_j| < 20$ **(b)** $20 < |C_i|, |C_j| < 30$

(c) $30 < |C_i|, |C_j| < 40$ **(d)** $40 < |C_i|, |C_j| < 50$

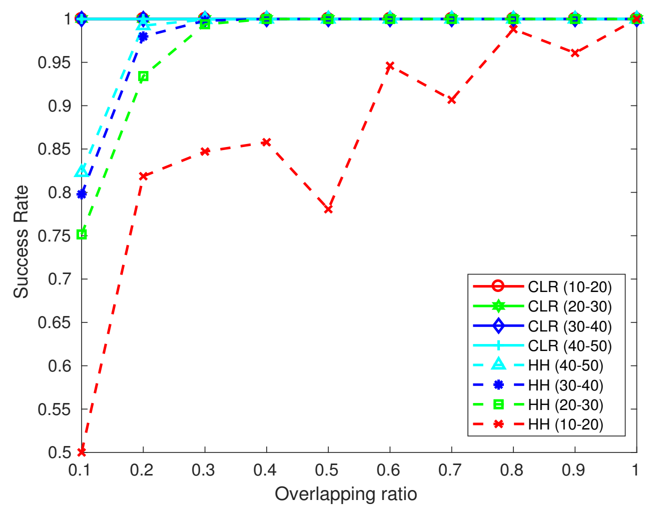
Figure 4.2: Theoretical and simulated MTTR when $T_i = T_j$ under varying time skew and different pair number of available channels of nodes i and j.

differences between them are caused by difference in number of available channels for each selected pair. Figs. 4.1(c), 4.2(b) and (c) also show the similar results. In addition, both Figs. 4.1 and 4.2 show that the simulation results are totally consistent with the theoretical results in Theorem 1: i) MTTR increases by cycle lengths of the two nodes. ii) The MTTR linearly increases by the time skew when $T_i \neq T_j$ (Fig. 4.1), while the MTTR is independent with the time skew when $T_i = T_j$ (Fig. 4.2).

Also, the simulated MTTRs are always smaller than the theoretical MTTR. It should be noted that the simulated MTTR are always much smaller than the theoretical MTTR in magnitude of ten thousand when T_i and T_j are not coprime as shown in Fig. 4.2, thus the performance of CLR in reality is better than in theoretical.



(a) T_i and T_j are coprime



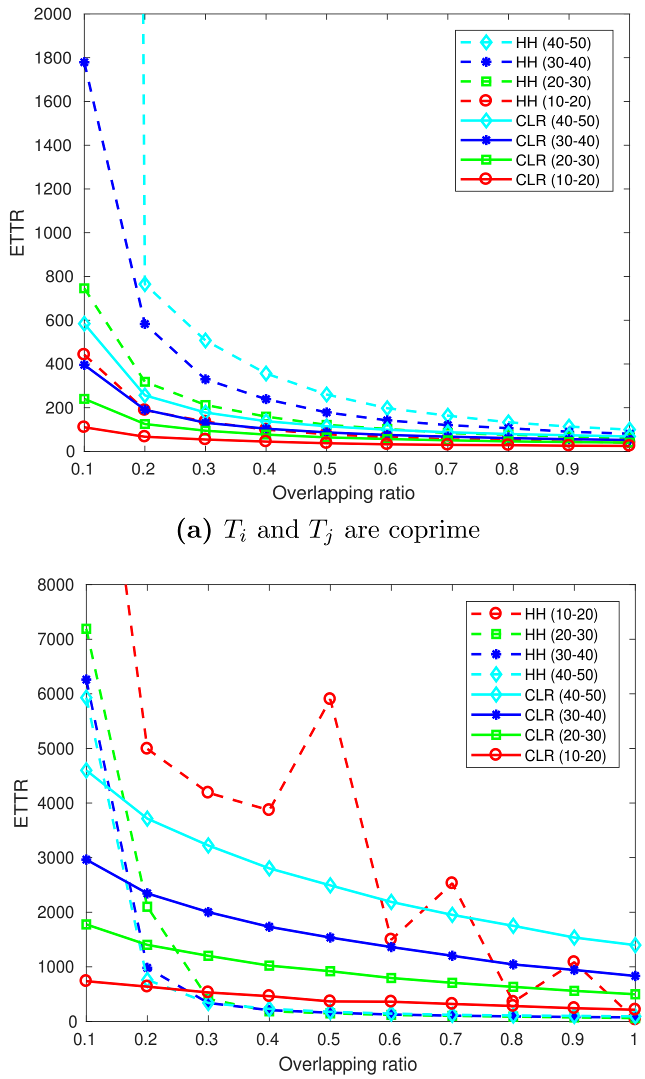
(b) T_i and T_j are not coprime

Figure 4.3: Success rate under varying overlapping ratio.

4.3 The Guarantee of Rendezvous

We study the guarantee of rendezvous under the varying overlapping ratio. By applying deadlock checking bit by bit on the binary ID of each node to independently change the cycle length, the rendezvous of CLR algorithm is guaranteed in both the two situations, as shown in Figs. 4.3(a) and (b). While the rendezvous of HH algorithm is not guaranteed for both the two situations.

For HH, when the number of available channels of the two nodes are large, but the overlapping ratio is small, nodes i and j may rightly miss with each other on the common channels at some time skew due to its interspersed CH seuqences. Thus in Fig. 4.3(a), the rendezvous of HH in $40-50$ period is not guaranteed when the overlapping ratio is less than 0.3. In addition, due to the randomness in the channel IDs, theorem 7 of [7] is not valid, for which the rendezvous can not be guaranteed. Thus at Fig. 4.3(b), the rendezvous of HH is not guaranteed in $20-30$, $30-40$ and $40-50$ periods as the overlapping ratio is less than $0.4, 0.4, 0.3$, respectively. Especially for $10-20$ period, the randomness in the channel IDs becomes more obvious when the number of available channels are small. Thus at Fig. 4.3(b), the HH in $10-20$ period is not guaranteed until the overlapping ratio is 1. Due to the randomness in channel IDs, as the number of available channels of the two nodes decrease, HH can not guarantee rendezvous with higher probability.



(a) T_i and T_j are coprime

(b) T_i and T_j are not coprime

Figure 4.4: ETTR under varying overlapping ratio (CLR: solid lines; HH: dashed lines).

4.4 Expected Time to Rendezvous

We study the ETTR under the varying overlapping ratio. When T_i and T_j are co-prime, the TTR between the two nodes is linear to the cycle length of their CH sequences. Since the cycle length of HH is three times of that of CLR, CLR always outperforms HH significantly under all of the four periods, as shown in Fig. 4.4(a). When T_i and T_j are not coprime, as the probability of failed rendezvous by HH is larger than 0.05 (Fig. 4.3b), the ETTR of HH is much higher than CLR in thousands, as shown in Fig. 4.4b in $30-40$ and $40-50$ periods at the overlapping ratio of 0.1, and the $20-30$ period at $0.1-0.2$. Especially for the $10-20$ period, HH always has much higher ETTR than CLR till full overlapping ratio of 1. The use of threshold to guarantee rendezvous discounts the ETTR of CLR. Thus, when the overlapping ratio is larger than 0.2, the ETTR of CLR is higher than that of the HH as for $20-30$, $30-40$ and $40-50$ periods. But CLR has a more stable performance that always guarantees rendezvous and has ETTR within 5000 time slots. So the trade off between the guaranteed rendezvous and the ETTR should be considered in reality, and we can change the threshold to adjust the ETTR of CLR.

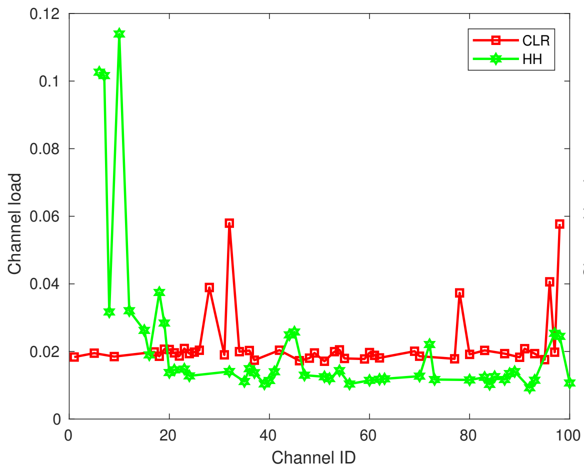
4.5 Channel Load

In CRNs, the channels available to each node are dynamically changed due to PUs' activities. Therefore, the channel load is an important measure to evaluate rendezvous algorithms. $l_c = \{l_{c_1}, l_{c_2}, \ldots, l_{c_{|C_i|}}\}$ is defined as set of probabilities, where l_{c_i} is channel load of channel c_i, the probability of rendezvous occurring on channel c_i considering all the $3 * 100 * T_i * |C_j|$ runs on the algorithm for each period. The smoother the distributions of channel load over all the available channels, the better the algorithm is. So the channel load min-max degree d is defined as follows:

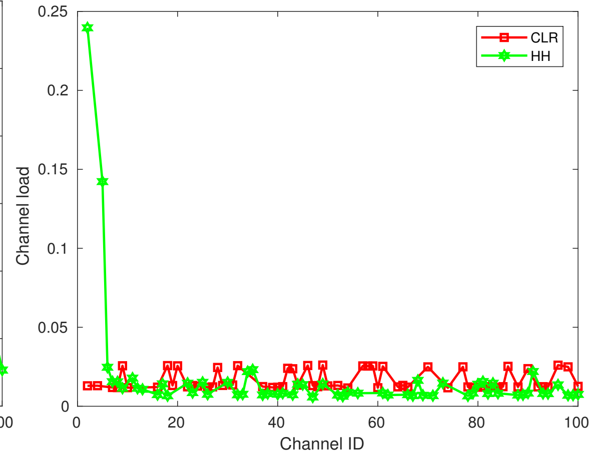
$$d = \frac{max_{\{l_c\}} - min_{\{l_c\}}}{avg_{\{l_c\}}}, \tag{4.1}$$

where $max_{\{l_c\}}$, $min_{\{l_c\}}$ and $avg_{\{l_c\}}$ represent the maximal, minimal and average channel load over all the available channels. The degree d quantifies the distance between the maximal and minimal channel load among all the channels. Thus the smaller d is, the better the algorithm is.

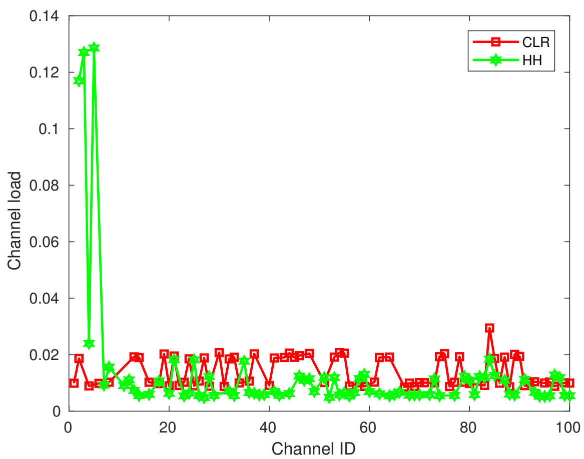
During the simulation, the channels are all randomly assigned with IDs from $[1, 100]$. The CH sequence generated by CLR occupies each channel with almost the same probability without depending on the channel IDs, so the channel load is smoothly distributed on each channel for all situations, as shown in both Figs. 4.5 and 4.6. While the CH sequences of HH depends on channel IDs significantly by assigning the channel with smallest ID to the parity slots. So with HH, the channel with smaller IDs suffers from much higher channel load than other channels, as the peak channel loads on channels with smaller IDs shown in Figs. 4.5 and 4.6. Tables 4.1 and 4.2 demonstrate comparisons of the channel load min-max degree between CLR and HH at all the four periods in the two situations respectively. Tables 4.1 and 4.2 also verify the results shown in Figs. 4.5 and 4.6, where the channel load min-max degrees of CLR are always much smaller than that of HH for all situations.



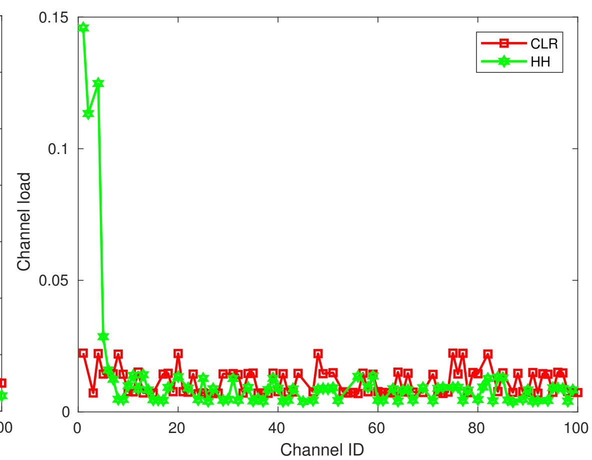
(a) $10 < |C_i|, |C_j| < 20$



(b) $20 < |C_i|, |C_j| < 30$



(c) $30 < |C_i|, |C_j| < 40$

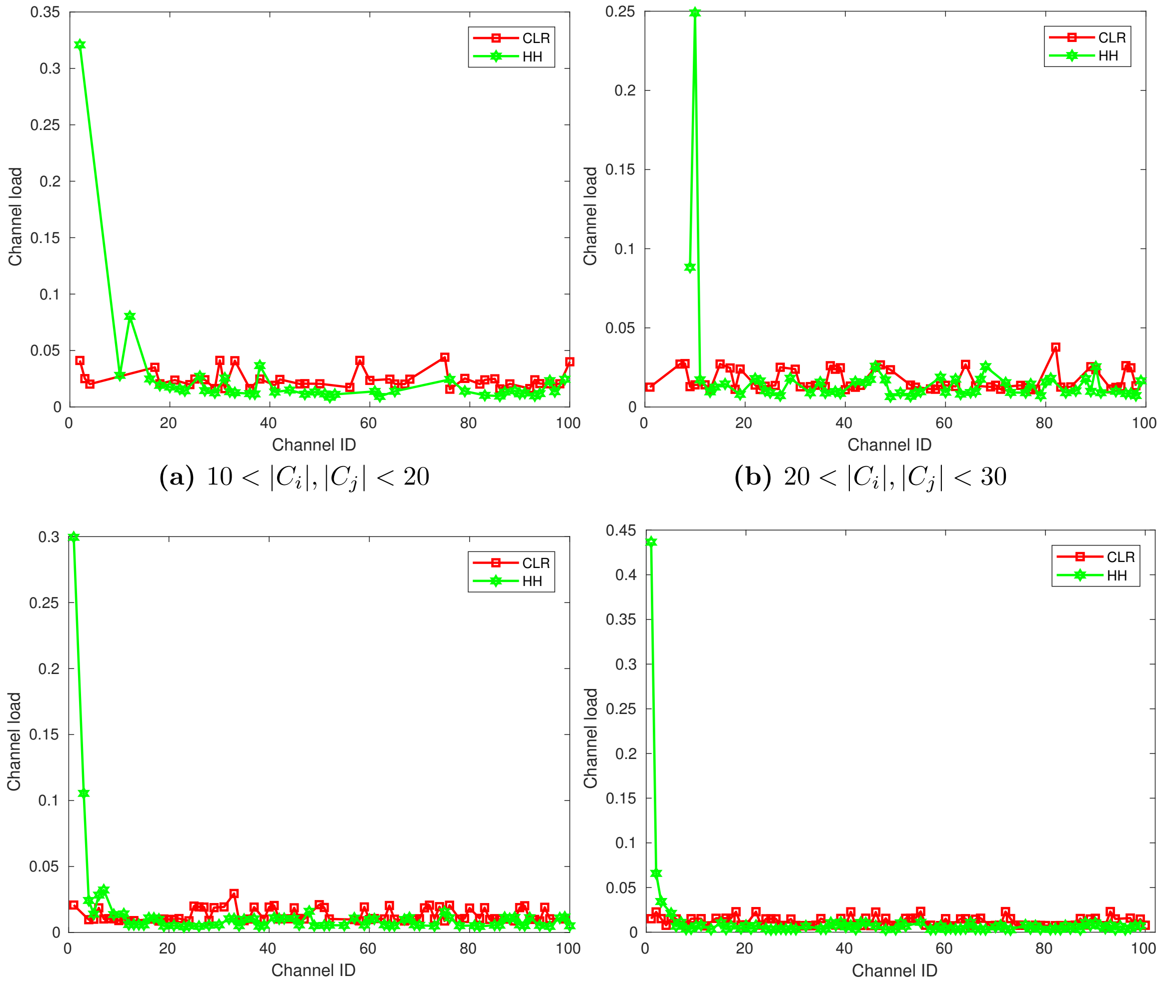


(d) $40 < |C_i|, |C_j| < 50$

Figure 4.5: Channel load distributions when T_i and T_j are coprime.

Table 4.1
Min-max degree when T_i and T_j are coprime.

10-20 period		**20-30 period**		**30-40 period**		**40-50 period**	
CLR	HH	CLR	HH	CLR	HH	CLR	HH
1.84	4.71	0.89	13.79	0.94	9.28	1.35	18.24



(a) $10 < |C_i|, |C_j| < 20$ **(b)** $20 < |C_i|, |C_j| < 30$

(c) $30 < |C_i|, |C_j| < 40$ **(d)** $40 < |C_i|, |C_j| < 50$

Figure 4.6: Channel load distributions when T_i and T_j are not coprime.

Table 4.2

Min-max degree when T_i and T_j are not coprime.

10-20 period		**20-30 period**		**30-40 period**		**40-50 period**	
CLR	HH	CLR	HH	CLR	HH	CLR	HH
1.17	427.3	1.61	433.2	1.59	630.7	1.35	1235.7

Chapter 5

Related Work

The existing work on CH based rendezvous algorithm can be categorized by multiple aspects under single radio or multiple radio scenarios as follows: symmetric/asymmetric, synchronous/asynchronous, homogeneous/heterogeneous and oblivious/non-oblivious.

Symmetric/asymmetric models based CH algorithms. Symmetric model assumes that the nodes wish to rendezvous share the same set of available channel set. Such as, the ETCH algorithm [8], the sequence based rendezvous algorithm combining with the parity slots in [12], the prime modulation technique combining with the parity slot proposed in SSCH algorithm [18], the symmetric model in Jump-stay rendezvous algorithm [10], as well as the three rendezvous algorithm summarized in [19]. In the asymmetric model, the available channels of the two nodes can be different, but it still assumes that the number of available channels between the two nodes are same. Such as, the asymmetric algorithm proposed in [10], and the asymmetric model based ring-walk channel hopping algorithm proposed in [20].

Synchronous/asynchronous models based CH algorithms. The synchronous CH algorithms assume there exist a global clock synchronous mechanism among the nodes with to rendezvous. Such as, the quorum based rendezvous proposed in [14], SYNC-ETCH proposed in [8] and the probability based rendezvous algorithm proposed in [13]. On the other way, the ASYNC-ETCH in [8], the symmetric sequence based rendezvous algorithm [12], and the symmetric Latin-square based asynchronous algorithm in [14] do not make the assumptions about the existence of the synchronous mechanism.

Homogeneous/heterogeneous models based CH algorithms. All the algorithms in [8, 10, 12, 14] consider the homogeneous model, that is, the nodes wish to rendezvous with each other share the same set of available channels. While the heterogeneous model is applicable to the situation where different node sense different set of channels. Such as, the HH algorithm that assumes the consecutive channel labels in [7], the unguaranteed probability based CH algorithm in [13], the MTP algorithm proposed in [17] and the DCSR algorithm in [15].

The oblivious/non-oblivious models based CH algorithms. The oblivious channel labeling is defined as the case where two nodes wish to rendezvous have different labeling policy. While the non-oblivious channel labeling assumes that the two nodes share the same labeling policy on the global available channels, and the two nodes share the same set of globally available channels. Both the DCSR algorithm in [15] and the MTP algorithm with lower efficiency in [17] are achieved under the non-oblivious model.

There are also some work that study the rendezvous problem in multi-radio CRNs. The extended jump stay rendezvous algorithm proposed in [21] studied the multi-radio scenario by assigning two roles to the radios of each node. While the work in [22] studied the rendezvous problem under the multi-radio multi-hop scenario by proposing a channel diverse routing algorithm, and addition and multiplication operators based CH algorithm. For the rendezvous for the multi-hop CRNs, both the link based rendezvous algorithm and the routing algorithm should be studied, as the multi-hop extension of the ring-walk algorithm in [20].

Chapter 6

Conclusion

6.1 Contributions

This report proposes a guaranteed rendezvous algorithm named cycle length based rendezvous (CLR) algorithm. The CLR guarantees rendezvous no matter T_i and T_j are coprime or not, where T_i and T_j are two prime numbers representing the cycle lengths of nodes i and j, respectively. To guarantee rendezvous when T_i and T_j are not coprime, we introduce a new strategy that each node is able to independently change the cycle length between T_i^0 and T_i^1 at each slot the deadlock situation is detected. The threshold T_{max} using for deadlock checking is defined as the length of the maximal possible rendezvous period, and is defined by two possible values T_{max}^0 and T_{max}^1. The deadlock situation detected at $t = T_{max}^0 + k * T_{max}^1, k = 0, 1, \cdots, max\{W_i, W_j\} - 1$

combining with $k+1$-th bit of the binary ID of each node is used to decide the time point and the way to independently change the cycle length, where W_i and W_j are number of bits in the binary ID of node i and j, respectively. As long as the current checking bit of two two nodes are different, the difference of the cycle lengths between them can be guaranteed, thus the rendezvous between the two nodes can be guaranteed.

We have conducted both theoretical and simulation studies to evaluate the performance of the CLR algorithm. The theoretical results show that the CLR is guaranteed under all the possible time skew δ between the two nodes. The simulation results demonstrate that the CLR algorithm outperforms the well-known heterogeneous hopping (HH) algorithms by the following three metrics: i) Guarantee of rendezvous, CLR always provides guaranteed rendezvous, while HH is not able to guarantee rendezvous in both situations when T_i and T_j are coprime or not. ii) ETTR, CLR gives much smaller ETTR than HH when T_i and T_j are coprime. iii) Channel load, the channel loads are smoothly distributed over all the available channels by using CLR, while HH highly depends on the channels with smaller IDs (with peak channel loads on the channels with smaller IDs).

6.2 Future Work

There are two aspects for our future work based on this report. The first one is to study the rendezvous in the multi-radio CRNs, where the bits of binary ID of each radio node will be checked by all the radios in parallel in FIFO order. The other idea is the study of the multi-user scenarios, where the rendezvous among multiple nodes (more than two) will be considered. The algorithm proposed in this report actually works for the multi-user scenario because the oblivious feature of the CLR algorithm. But the collision on channels should be properly studied.

References

[1] L. Gou, X. Xu, C. Zhang, and M. Song, “Guaranteed rendezvous for cognitive radio networks based on cycle length,” in *Proc. of IEEE/CIC ICCC*, 2017.

[2] Y. Xu, Y. Zhang, Q. Wu, L. Shen, and J. Wang, “Dynamic spectrum access for cognitive small cell networks: A robust graphical game approach,” in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, July 2016, pp. 1–6.

[3] X. Wang, J. Chen, A. Dutta, and M. Chiang, “Adaptive video streaming over whitespace: Svc for 3-tiered spectrum sharing,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 28–36.

[4] V. Brik, E. Rozner, S. Banerjee, and P. Bahl, “Dsap: a protocol for coordinated spectrum access,” in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, Nov. 2005, pp. 611–614.

[5] S. Krishnamurthy, M. Thoppian, S. Venkatesan, and R. Prakash, “Control channel based mac-layer configuration, routing and situation awareness for cognitive radio networks,” in *MILCOM 2005 - 2005 IEEE Military Communications Conference*, Oct 2005, pp. 455–460 Vol. 1.

[6] J. Perez-Romero, O. Sallent, R. Agusti, and L. Giupponi, “A novel on-demand cognitive pilot channel enabling dynamic spectrum allocation,” in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, April 2007, pp. 46–54.

[7] S. H. Wu, C. C. Wu, W. K. Hon, and K. G. Shin, “Rendezvous for heterogeneous spectrum-agile devices,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 2247–2255.

[8] Y. Zhang, Q. Li, G. Yu, and B. Wang, “Etch: Efficient channel hopping for communication rendezvous in dynamic spectrum access networks,” in *2011 Proceedings IEEE INFOCOM*, April 2011, pp. 2471–2479.

[9] X. J. Tan, C. Zhou, and J. Chen, “Symmetric channel hopping for blind rendezvous in cognitive radio networks based on union of disjoint difference sets,” *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[10] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, “Jump-stay rendezvous algorithm for cognitive radio networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1867–1881, Oct. 2012.

[11] J. Li, H. Zhao, J. Wei, D. Ma, and L. Zhou, "Sender-jump receiver-wait: a simple blind rendezvous algorithm for distributed cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[12] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Oct 2008, pp. 1–7.

[13] C. Xin, M. Song, L. Ma, and C. C. Shen, "Rop: Near-optimal rendezvous for dynamic spectrum access networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3383–3391, sept. 2013.

[14] K. Bian, J. M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 689–703, April 2011.

[15] B. Yang, M. Zheng, and W. Liang, "A time-efficient rendezvous algorithm with a full rendezvous degree for heterogeneous cognitive radio networks," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[16] Z. Gu, Y. Wang, Q.-S. Hua, and F. C. Lau, *Rendezvous in Distributed Systems.* Springer, 2017.

[17] Z. Gu, H. Pu, Q. S. Hua, and F. C. M. Lau, "Improved rendezvous algorithms for

heterogeneous cognitive radio networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 154–162.

[18] P. Bahl, R. Chandra, and J. Dunagan, "Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '04. ACM, 2004, pp. 216–230.

[19] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 216–227, Feb 2011.

[20] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *Proceedings of the 2010 IEEE/ACM Int'L Conference on Green Computing and Communications & Int'L Conference on Cyber, Physical and Social Computing*, ser. GREENCOM-CPSCOM '10, 2010, pp. 755–760.

[21] L. Yu, H. Liu, Y. W. Leung, X. Chu, and Z. Lin, "Multiple radios for fast rendezvous in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1917–1931, Sept 2015.

[22] T. Y. Lin, K. R. Wu, and G. C. Yin, "Channel-hopping scheme and channel-diverse routing in static multi-radio multi-hop wireless networks," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 71–86, Jan 2015.