

Improved Rendezvous Algorithms for Heterogeneous Cognitive Radio Networks

Zhaoquan Gu*, Haosen Pu*, Qiang-Sheng Hua[†], and Francis C.M. Lau[‡]

*Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

[†]School of Computer Science & Technology, Huazhong University of Science and Technology, China

[‡]Department of Computer Science, The University of Hong Kong, Hong Kong, China

Abstract—Cognitive radio networks (CRNs) have been proposed to solve the spectrum scarcity problem. One of their fundamental procedures is to construct a communication link on a common channel for the users, which is referred as *rendezvous*. In reality, the capability to sense the spectrum may vary from user to user, and such users form what is known as a heterogeneous cognitive radio network (HCRN). The licensed spectrum is divided into n channels, $U = \{1, 2, \dots, n\}$. We denote the capability of user i as $C_i \subseteq U$ and the set of available channels (i.e. the channels not occupied by the paying users) as $V_i \subseteq C_i$. We study the rendezvous problem in HCRN under two circumstances: fully available spectrum ($V_i = C_i$) and partially available spectrum ($V_i \neq C_i$). For any two users a, b , we propose the Traversing Pointer (TP) algorithm that guarantees rendezvous in $O(\max\{|C_a|, |C_b|\} \log \log n)$ time slots for the fully available spectrum scenario. This result is only $O(\log \log n)$ larger than our constructive lower bound. Moreover, it removes an $O(\min\{|C_a|, |C_b|\})$ factor as compared to the state-of-the-art result ($O(|C_a||C_b|)$ in [26]). For the partially available spectrum scenario, we propose the Moving Traversing Pointers (MTP) algorithm to guarantee rendezvous in $O((\max\{|V_a|, |V_b|\})^2 \log \log n)$ time slots, which works more efficiently than the previous best result ($O(|C_a||C_b|)$ in [25]) in various circumstances. We also conduct extensive simulations and the results corroborate our analysis.

Index Terms—Heterogeneous Cognitive Radio Network, Rendezvous, Fully available spectrum, Partially available spectrum

I. INTRODUCTION

The wireless spectrum is becoming scarce due to the rapid growth of wireless devices. There are two types of wireless spectrum, licensed spectrum which is owned by paying users, such as the TV frequency bands [8]; and unlicensed spectrum which is free to use for wireless devices, such as the Industrial Scientific and Medical (ISM) band [9]. The unlicensed spectrum is overcrowding with the increasing demands for wireless services, while the utilization of licensed spectrum remains consistently at a low level [27]. Cognitive radio networks (CRNs) were proposed to improve the situation by allowing unlicensed users to exploit and access the unused parts of the licensed spectrum. Unless otherwise specified, ‘users’ hereafter refers to the unlicensed users.

In constructing a CRN, the users have to establish a link on a common frequency band (channel) for communication, which is referred as *rendezvous* [19]. The process of constructing a communication link includes such detailed operations as beaconing and handshaking, which we leave out in this paper

but focus on the rendezvous problem: how to choose the same channel at the same time? Technically, the licensed spectrum is assumed to be divided into n non-overlapping channels, $U = \{1, 2, \dots, n\}$; correspondingly, the time is divided into slots of equal length [10], [11], [13], [18], [22]. The user can access an *available* channel in each time slot where available means the channel is not occupied by nearby paying users, and two users rendezvous if they choose the same channel in the same time slot.

Many extant rendezvous algorithms have been proposed by constructing channel sequences based on the channels’ labels [5], [10], [13], [18], [22], [24], and the users accessing the channels by repeating the sequences are guaranteed to rendezvous in bounded time based on such principles as the Chinese Remainder Theorem. The state-of-the-art results can guarantee rendezvous in $O(n^2)$ time slots even for the worst situations. However, they all assume that the users have the capability to sense and access *all* the licensed channels, which is unrealistic when the number of channels (n) is very large and some wireless devices may only operate on a small fraction of the channels. Therefore, the notion of heterogeneous cognitive radio network (HCRN) came about, in which the users may have different spectrum-sensing capabilities. Several algorithms have been proposed for the rendezvous problem in HCRN [21], [25], [26].

Let $C_i \subseteq U$ be the spectrum sensing capability for user i where C_i is a set of continuous channels in U [21], and $V_i \subseteq C_i$ be the set of sensed available channels. We say the spectrum is *fully available* for user i if $V_i = C_i$; otherwise it is *partially available*. For any two users a and b with capability sets C_a, C_b and available channel sets V_a, V_b , rendezvous is verified in [21] when $|C_a|, |C_b| \leq 8$, but they do not provide a theoretical guarantee for all cases. The Heterogeneous Hopping (HH) algorithm, proposed in [26], guarantees rendezvous in $O(|C_a||C_b|)$ time slots when the spectrums of both users are fully available. Moreover, the Interlocking Channel Hopping (ICH) algorithm, proposed in [25] for users with partially available spectrum, also guarantees rendezvous in $O(|C_a||C_b|)$ time slots. In this paper, we propose improved algorithms for both scenarios. Different from most Chinese Remainder Theorem based constructions, our new method introduces two ‘pointers’ to construct the rendezvous sequence. For the fully available scenario, the *moving pointer* traverses the channels in the capability set while the *fixed pointer* stays at the first

channel. For the partially available scenario, the fixed pointer is modified to move once the moving pointer has traversed the channels. The contributions of this paper are as follows:

- 1) We propose a rendezvous scheme for the special case that $|V_a| = |V_b| = 2, V_a \cap V_b \neq \emptyset$. Based on three disjoint relaxed difference sets (DRDSs), the scheme guarantees rendezvous in $O(\log \log n)$ time slots, which is the foundation of our rendezvous algorithms.
- 2) We propose the Traversing Pointer (TP) algorithm when the users' spectrum is fully available. Two 'pointers' are introduced to traverse the channels and the algorithm guarantees rendezvous in $O(\max\{|C_a|, |C_b|\} \log \log n)$ time slots, which improves the latest result in [26] by an $O(\min\{|C_a|, |C_b|\})$ factor.
- 3) We propose the Moving Traversing Pointers (MTP) algorithm when the users' spectrum is partially available, which guarantees rendezvous in $O(\max\{|V_a|, |V_b|\}^2 \log \log n)$ time slots. When V_i only accounts for a small fraction of C_i , the MTP algorithm works more efficiently than the ICH algorithm [25].
- 4) We conduct extensive simulations to compare our algorithms with extant algorithms and the results show that our algorithms have better performance.

The remainder of the paper is organized as follows. The next section highlights some related work on the rendezvous problem. The system model and problem formulations are given in Section III. We introduce the rendezvous scheme for two available channels in Section IV as a foundation. Section V describes the TP algorithm for the scenario that the users' spectrum is fully available and Section VI presents the MTP algorithm for when the users' spectrum is partially available. We have conducted extensive simulations, and the results are discussed in Section VII. Finally, we conclude the paper in Section VIII.

II. RELATED WORKS

A. Rendezvous for CRN

There are three categories of rendezvous algorithms: centralized algorithms, decentralized algorithms based on Common Control Channel (CCC) and blind rendezvous algorithms.

Centralized algorithms assume the existence of a central controller or a common control channel (CCC) which simplifies the problem as users can coordinate through the central unit [15], [20]. However, the central controller or the CCC could be a bottleneck in large scale networks as well as being easily vulnerable to adversary attacks. There are also several decentralized algorithms establishing local CCCs [14], [16] based on the sensed channels and they can be used for communication between neighbors. Nevertheless, these algorithms incur too much overhead in establishing and maintaining the local CCCs.

Therefore, blind rendezvous algorithms without any CCC were introduced in recent years [1]–[3], [5]–[7], [10], [12], [18], [22]. Channel Hopping (CH) is the main technique behind, where the users hop among the sensed available

channels in different time slots on the basis of a pre-generated CH sequence, and rendezvous can be guaranteed once they access the same channel in the same time slot.

Generated Orthogonal Sequence (GOS) [7] is a pioneering work, which generates an orthogonal sequence of length $N(N+1)$ based on a random permutation of $\{1, 2, \dots, n\}$. Nevertheless, GOS works for the situation that all channels are available. Quorum-based Channel Hopping (QCH) [1]–[3] is proposed for synchronous users (i.e. the users start at the same time) on the basis of quorum systems, while the enhanced Asynchronous QCH [4] is applicable to two asynchronous users, which is only limited to two channels.

Jump-Stay (JS) [18], Channel Rendezvous Sequence (CRSEQ) [22], Disjoint Relaxed Difference Set (DRDS) [10], Conversion Based Hopping (CBH) [13], and the scheme in [5] are several representative blind rendezvous algorithms. JS generates a sequence of length $O(n^3)$ for each user by generating a *jump*-pattern and a *stay*-pattern. Two users are guaranteed to rendezvous in $O(n^3)$ time slots in one of four possible pattern combinations: jump-jump, jump-stay, stay-jump, stay-stay. This result is later improved to $O(n^2)$ as the Enhanced JS in [17]. CRSEQ constructs a sequence of $O(n^2)$ numbers on the basis of the triangle number (i.e. $T_i = \frac{i(i+1)}{2}$ when $i \in [1, n]$) and modular operations. Two users can rendezvous on the same channel quickly by repeating the sequence. DRDS based rendezvous algorithm is a new method guaranteeing rendezvous in $O(n^2)$ time slots, which is implemented by constructing a DRDS and transforming the DRDS into a CH sequence. In [5], the rendezvous sequences are designed based on a special construction for two channels. It counts the number of available channels and finds two primes larger than the number, and then two channels are chosen according to the primes and rendezvous is guaranteed in $O(|V_a||V_b| \log \log n)$ time slots where V_a, V_b are the sets of available channels. CBH algorithm is the only one using no global information such as the number of channels n . Assuming each user has a unique identifier (ID), CBH converts the user's ID into a distinct string and constructs sequences based on the conversion. CBH guarantees rendezvous in $O((\max\{|V_a|, |V_b|\})^2)$ time slots. Most of these algorithms are based on the Chinese Remainder Theorem by picking primes and guaranteeing rendezvous according to laws and derivations in number theory.

B. Rendezvous for Heterogeneous CRN

When the users have different spectrum-sensing capabilities, the rendezvous process becomes different, especially when the sensed channels account for a small fraction of all channels. There are mainly three works [21], [25], [26] that propose guaranteed rendezvous for heterogeneous networks.

In [21], rendezvous can be achieved when $|C_a|, |C_b| \leq 8$, where $C_a, C_b \subseteq U$ represent the sensing capabilities. However, it does not provide a theoretical guarantee for all cases under the heterogeneous circumstance. In [26], a new channel hopping algorithm called Heterogeneous Hopping (HH) is proposed, which is realized with a two-layer design: fixed-short-

TABLE I
MTTR COMPARISON FOR FULLY & PARTIALLY AVAILABLE SCENARIOS

Algorithms	Fully Available Scenario	Partially Available Scenario
HH [26]	$O(C_a C_b)$	—
ICH [25]	$O(C_a C_b)$	$O(C_a C_b)$
TP (this paper)	$O(\max\{ C_a , C_b \} \log \log n)$	—
MTP (this paper)	$O(\max\{ V_a , V_b \}^2 \log \log n)$	$O(\max\{ V_a , V_b \}^2 \log \log n)$

Remarks: 1) “—” means the algorithm is not applicable to the partially available spectrum scenario; 2) $C_a, C_b \subseteq U$ represent the capability sets of user a and b respectively; 3) $V_a \subseteq C_a, V_b \subseteq C_b$ represent the available channel set of users a and b respectively.

cycle and parity-alignment. These two techniques could help guide rendezvous in $O(|C_a||C_b|)$ time slots. However, this result is only applicable to the fully available spectrum. Interlock Channel Hopping (ICH) is proposed in [25], where three types of sequences are constructed for heterogeneous networks: fixed sequence, rotating sequence and insurance sequence. This method could guarantee rendezvous in $O(|C_a||C_b|)$ time slots but it is inefficient when the number of available channels is much smaller than the capability.

III. PRELIMINARIES

A. System Model

Assume the licensed spectrum is divided into n non-overlapping channels as $U = \{1, 2, \dots, n\}$. Each user is equipped with a cognitive radio to sense the licensed spectrum and a channel is *available* for the user if it is not occupied by any nearby paying users. The users may have different spectrum sensing capabilities and suppose that user i can sense a set of continuous channels as $C_i = \{c_x, c_{x+1}, \dots, c_{x+k_i-1}\} \subseteq U$ [25], [26], where c_x is the starting channel and $k_i = |C_i|$, $1 \leq x \leq n - k_i + 1$. The channels in C_i are either occupied by nearby paying users or available for the (unlicensed) user i , and denote $V_i \subseteq C_i$ as the set of all available channels after the spectrum sensing stage. Time is assumed to be divided into slots of equal length $2t$, where t is the time sufficient for establishing a communication link if the users access the same channel at the same time slot ($t = 10ms$ according to the IEEE 802.22 [23]). The intuitive idea of setting each time slot to be $2t$ is to ensure an overlap of t exists for link establishment even when the users do not start their process at aligned time slots [12].

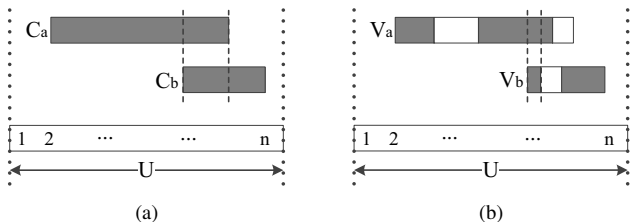


Fig. 1. Example of different spectrum sensing capability sets and available channel sets. Channels in black are available and those in white means they are occupied by some paying users.

For two users a and b with different spectrum sensing capability sets C_a, C_b and available channel sets V_a, V_b , they

can rendezvous on some common channel if $V_a \cap V_b \neq \emptyset$, which implies that their capability sets intersect. In this paper, we consider two scenarios: *fully available spectrum* and *partially available spectrum*. If all channels in the users' sensing capability sets are available after the spectrum sensing stage, we call that fully available (i.e. $V_i = C_i$). But in most circumstances, some channels are likely occupied ($V_i \neq C_i$) and we call that partially available. For example, Fig. 1 shows the different capability sets of users a and b , and it is a fully available scenario when all channels in the capability sets are available, as in Fig. 1(a). Fig. 1(b), where some channels are occupied by the paying users, is a partially available scenario.

In each time slot, user i can access an available channel from V_i and attempt rendezvous with its potential neighbors. We say *rendezvous* happens when the users choose the same channel in the same time slot. *Time to rendezvous (TTR)* denotes the number of time slots taken to rendezvous once all users have begun their attempt. Since the users are dispersed in different places and they may begin the rendezvous process in different time slots, we are interested in designing efficient algorithms for asynchronous users and we use *Maximum Time to Rendezvous (MTTR)* to judge the performance of the rendezvous algorithms.

B. Problem Formulations

We formulate the rendezvous problem for the fully available spectrum in HCRN as follows:

Problem 1: For any spectrum sensing capability $C_i \subseteq U$, design an algorithm to access channels over different time slots $t : f_{C_i}(t) \in C_i$, such that for any two users a and b with $C_a, C_b \subseteq U, C_a \cap C_b \neq \emptyset$, supposing user a starts $\delta \geq 0$ time slots earlier than user b ,

$$\exists T_\delta, \text{ s.t. } f_{C_a}(T_\delta + \delta) = f_{C_b}(T_\delta)$$

The *TTR* value is T_δ and $MTTR = \max_{\forall \delta} T_\delta$. The goal is to design a rendezvous algorithm with bounded *MTTR*.

Although a fully available spectrum rarely happens in practice, it represents the best spectrum condition for designing rendezvous algorithms. For more general situations, we formulate the rendezvous problem for the partially available spectrum scenario as follows:

Problem 2: For any spectrum sensing capability $C_i \subseteq U$ and available channel set $V_i \subseteq C_i$, design an algorithm to access channels over different time slots $t : f_{C_i, V_i}(t) \in V_i$, such that for any two users a and b with $C_a, C_b \subseteq U, V_a \subseteq C_a, V_b \subseteq C_b, V_a \cap V_b \neq \emptyset$, supposing user a starts $\delta \geq 0$ time slots earlier than user b ,

$$\exists T_\delta, \text{ s.t. } f_{C_a, V_a}(T_\delta + \delta) = f_{C_b, V_b}(T_\delta)$$

The *TTR* value is T_δ and $MTTR = \max_{\forall \delta} T_\delta$. The goal is to design rendezvous algorithm with bounded *MTTR*.

For example, $U = \{1, 2, \dots, 100\}$, $C_a = \{2, 3, 4, 5, 6\}$, $C_b = \{5, 6, 7\}$, and suppose that both users a and b adopt a simple algorithm by repeating the channels in their sensing capability set and user a is two time slots earlier than user b . As depicted in Fig. 2, they rendezvous on channel 5 at time slot 9,

and thus $TTR = 9 - 2 = 7$ time slots. In fact, if the users apply the extant algorithms based on all channels in U , the maximum rendezvous time could be $O(n^2) \approx 10000$ time slots, which is unacceptable. This figure is an example of the fully available spectrum scenario. When some channels are occupied, for example $V_a = \{2, 3, 6\}$, $V_b = \{6, 7\}$, they cannot rendezvous on channel 5 and one more time slot is needed, and this is an example of the partially available spectrum scenario.

Time	1	2	3	4	5	6	7	8	9	10	...
User a	2	3	4	5	6	2	3	4	5	6	...
User b			5	6	7	5	6	7	5	6	...

Fig. 2. An example of rendezvous problem in HCRN

IV. RENDEZVOUS SCHEME FOR TWO CHANNELS

In this section, we propose a rendezvous scheme for the special scenario where each user has two available channels $|V_a| = |V_b| = 2$ and there exists at least one common channel ($V_a \cap V_b \neq \emptyset$). Our method is to construct a sequence of length $T_2 = 16(\lceil \log \log n \rceil + 1)$ based on three Disjoint Relaxed Difference Sets (DRDSs). Supposing the available channel set of the user is $V = \{v_1, v_2\} \subseteq U$ where $v_1 < v_2$, the scheme is described in Alg. 1.

Algorithm 1 Rendezvous Scheme for Two Channels

- 1: $l_1 = \lceil \log n \rceil + 1$, $l_2 = \lceil \log l_1 \rceil + 1$;
- 2: Find the smallest number $c \in [1, l_1]$ such that the c -th bit of v_2 is 1 and the c -th bit of v_1 is 0;
- 3: Let $\vec{D} = \{*, c_{l_2}, c_{l_2-1}, \dots, c_1\}$ where $(c_{l_2}, c_{l_2-1}, \dots, c_1)$ is the binary representation of c ;
- 4: Denote the rendezvous sequence $S = \emptyset$;
- 5: **for** $r = 1 : l_2 + 1$ **do**
- 6: If $\vec{D}(r) = *$, add $S_* = (v_1, v_1, v_2, v_1, v_1, v_2, v_2, v_2)$ twice to S ;
- 7: If $\vec{D}(r) = 0$, add $S_0 = (v_1, v_1, v_2, v_1, v_2, v_1, v_2, v_2)$ twice to S ;
- 8: If $\vec{D}(r) = 1$, add $S_1 = (v_1, v_1, v_2, v_1, v_2, v_2, v_2, v_1)$ twice to S ;
- 9: **end for**
- 10: Repeat the rendezvous S until rendezvous;

Alg. 1 finds the smallest number $c \in [1, l_1]$ such that the c -th bit of v_2 is 1 but the c -th bit of v_1 is 0, where $l_1 = \lceil \log n \rceil + 1$ (since $v_1 < v_2$, c must exist). It is obvious that c can be represented by $l_2 = \lceil \log \log n \rceil + 1$ binary bits. We construct vector \vec{D} by adding a special symbol $*$ to the binary representation as in Line 3, and we generate the rendezvous sequence in $l_2 + 1$ rounds. In each round, different sequences S_*, S_0, S_1 are added twice to S and the intuitive idea of designing these sequences comes from the DRDS [10].

Definition 4.1: A set $S = \{a_1, a_2, \dots, a_k\} \subseteq Z_n$ (the set of all nonnegative integers less than n) is called a Relaxed

Difference Set (RDS) if for every $d \neq 0 \pmod n$ there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j \equiv d \pmod n$, where $a_i, a_j \in D$.

Definition 4.2: A set $D = \{S_1, S_2, \dots, S_n\}$ is called a Disjoint Relaxed Difference Set (DRDS) under Z_n if $\forall S_i \in D$, S_i is an RDS under Z_n and $\forall S_i, S_j \in D, i \neq j, S_i \cap S_j = \emptyset$.

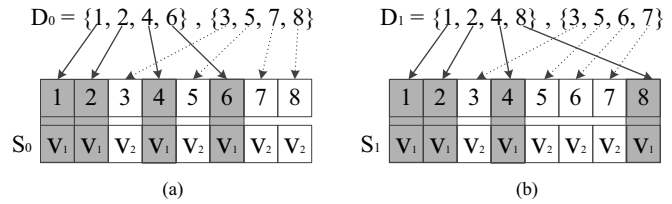


Fig. 3. Constructing sequences S_0, S_1 on the basis of D_0, D_1

Let $D_* = \{\{1, 2, 4, 5\}, \{3, 6, 7, 8\}\}$, $D_0 = \{\{1, 2, 4, 6\}, \{3, 5, 7, 8\}\}$ and $D_1 = \{\{1, 2, 4, 8\}, \{3, 5, 6, 7\}\}$; it is easy to check that they are three DRDS under Z_8 . S_* , S_0 and S_1 are then constructed on the basis of D_*, D_0, D_1 respectively. The construction is shown in Fig. 3 (S_0, S_1 as examples). In each round, S_*, S_0 or S_1 is added twice to the rendezvous sequence because the users can start the algorithm asynchronously.

Lemma 4.1: Every 8 continuous time slots in each round correspond to a DRDS.

Proof: Consider the round containing two S_0 where S_0 is constructed based on the DRDS D_0 . Every 8 continuous time slots $[i, i + 7]$ where $1 \leq i \leq 9$ can be seen as rotating S_0 by $i - 1$ time slots. From the definition of RDS, the rotation of the RDS is also an RDS and thus the rotation of S_0 also corresponds to a DRDS. For example, when $i = 3$, the 8 continuous time slots are $\{v_2, v_1, v_2, v_1, v_2, v_2, v_1, v_1\}$ and that corresponds to the DRDS $\{\{2, 4, 7, 8\}, \{1, 3, 5, 6\}\}$. We can get the same result for the other two sequences S_1, S_* , and thus the lemma holds. ■

Consider two users a and b with available channel sets $V_a = \{a_1, a_2\}$ and $V_b = \{b_1, b_2\}$, and suppose the chosen numbers in Line 2 are c_a, c_b respectively. We show the correctness of Alg. 1 based on c_a, c_b .

Lemma 4.2: Alg. 1 guarantees rendezvous in 16 time slots if $c_a = c_b$.

Proof: When $c_a = c_b$, we claim that $a_1 \neq b_2$ and $a_2 \neq b_1$. If $a_1 = b_2$, we have $b_1 < b_2 = a_1 < a_2$. From Line 2, the c_b -th bit of b_2 is 1 and the c_a -th bit of a_1 is 0, but $c_a = c_b$ and it is a contradiction. Thus $a_1 \neq b_2$. Similarly, $a_2 \neq b_1$. Since the users have at least one common channel, $a_1 = b_1$ or $a_2 = b_2$ and we show that both pairs $(a_1, b_1), (a_2, b_2)$ appear in the sequences when two users have begun their process.

Denote the constructed sequences for the users as S_a and S_b respectively, and they are composed of $l_2 + 1$ rounds. We say the i -th round of user a (denoted as $r(a, i)$) overlaps with the j -th round of user b ($r(b, j)$) if their intersection length is at least 8 (time slots). Without loss of generality, suppose user a is δ time slots earlier than user b . We show the lemma in two cases.

- 1) If $r(b, 1)$ overlaps with $r(a, 1)$ and there are at least 8

overlapping time slots. From Lemma 4.1, the continuous 8 time slots correspond to two DRDSs for users a and b . From the definition of DRDS, we can check that (a_1, b_1) and (a_2, b_2) both exist in the 8 time slots, and thus they rendezvous in the first round of user b .

- 2) If $r(b, 1)$ overlaps with $r(a, i)$ where $1 < i \leq l_2 + 1$ and there are at least 8 overlapping time slots. If (a_1, b_1) does not exist in the intersecting 8 slots, channel b_1 meets a_2 in four time slots and b_2 also has to meet a_1 in four time slots. However, the sequence added in $r(b, 1)$ is different from the sequence in $r(a, i)$ (S_* is added twice in $r(b, 1)$ while S_0 or S_1 is added in $r(a, i)$), and this situation cannot happen. Thus, (a_1, b_1) exists in the first round of user b . Similarly, we can prove that (a_2, b_2) exists. Thus they can rendezvous in 16 time slots.

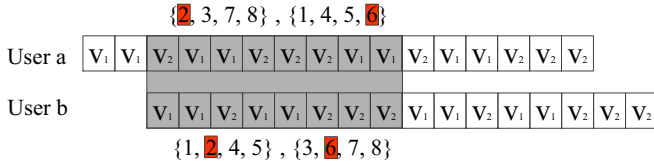


Fig. 4. Example of $r(b, 1)$ overlapping with $r(a, 1)$

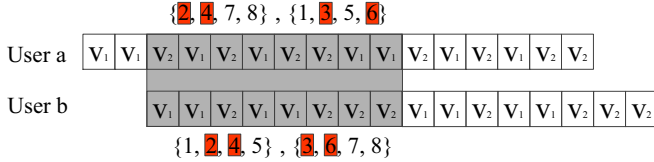


Fig. 5. Example of $r(b, 1)$ overlapping with $r(a, i)$ where $1 < i \leq l_2 + 1$

As depicted in Fig. 4, $r(b, 1)$ overlaps with $r(a, 1)$ and the first 8 overlapping time slots form two DRDSs as $\{\{2, 3, 7, 8\}, \{1, 4, 5, 6\}\}$ and $\{\{1, 2, 4, 5\}, \{3, 6, 7, 8\}\}$. Then we can check that (a_1, b_1) exists in the 2-nd time slot and (a_2, b_2) happens in the 6-th time slot. Similarly, Fig. 5 shows an example that $r(b, 1)$ overlaps with $r(a, i)$ where $1 < i \leq l_2 + 1$, and both pairs (a_1, b_1) and (a_2, b_2) exist in the first overlapping 8 time slots. Therefore, the lemma holds. ■

Lemma 4.3: Alg. 1 guarantees rendezvous in $T_2 = 16(\lceil \log \log n \rceil + 1)$ time slots if $c_a \neq c_b$.

Proof: When $c_a \neq c_b$, there are four possible combinations of rendezvous situations: $a_1 = b_1, a_1 = b_2, a_2 = b_1, a_2 = b_2$. Thus the two users' overlapping sequences must contain the four pairs $(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)$. We show the lemma in two cases.

- 1) If $r(b, 1)$ overlaps with $r(a, 1)$, $(a_1, b_1), (a_2, b_2)$ exists in the overlapping part from Lemma 4.2. Since $c_a \neq c_b$, without loss of generality, suppose $c_a < c_b$ and there exists $1 \leq i \leq l_2$ such that the i -th bit of c_a is 0 but the i -th bit of c_b is 1. When $r(b, i+1)$ overlaps with $r(a, i+1)$, we claim that (a_1, b_2) and (a_2, b_1) exist in the overlapping part. If (a_1, b_2) does not happen, a_1 has to meet b_1 four

times and a_2 has to meet b_2 four times; however, $r(a, i+1)$ and $r(b, i+1)$ use different sequences (S_0 and S_1) and this scenario cannot happen. Thus (a_1, b_2) appears at least once during the intersecting part. Similarly, (a_2, b_1) also exists. Therefore, rendezvous can be guaranteed in $16(i+1) \leq T_2$ time slots.

- 2) If $r(b, 1)$ intersects with $r(a, i)$ where $1 < i \leq l_2 + 1$, the pairs (a_1, b_1) and (a_2, b_2) both exist from Lemma 4.2. Using a similar technique, we can check that (a_1, b_2) and (a_2, b_1) exist in the first round of user b .

Combining the two cases, rendezvous can be guaranteed in T_2 time slots, and the lemma holds. ■

From Lemma 4.2-4.3, we have the theorem:

Theorem 1: Alg. 1 guarantees rendezvous in $T_2 = 16(\lceil \log \log n \rceil + 1)$ time slots for the special situation that each user has two available channels.

Remark 4.1: Another method to achieve rendezvous for two available channels is proposed in [5] which has sequence length $O(\log \log n)$. It however is too complicated and difficult to implement. In contrast, our proposed algorithm is simple and easy to implement in reality. More importantly, the intuition and the method of our construction are entirely different from [5].

V. RENDEZVOUS FOR FULLY AVAILABLE SPECTRUM

For the fully available spectrum scenario, we propose a new method called Traversing Pointer (TP) algorithm based on the rendezvous scheme for two channels. Consider two users a and b with spectrum sensing capability sets $C_a, C_b \subseteq U$, the TP algorithm guarantees rendezvous in $O(\max\{|C_a|, |C_b|\} \log \log n)$ time slots. Moreover, we show a constructive lower bound such that $\max\{|C_a|, |C_b|\}$ time slots are needed for rendezvous.

A. Traversing Pointer Algorithm

Suppose user i has the spectrum sensing capability set as $C_i = \{c_x, c_{x+1}, \dots, c_{x+k_i-1}\} \subseteq U$ where $k_i = |C_i|$, $1 \leq x \leq n - k_i + 1$ and $\forall c_j \in C_i$, channel c_j is available.

The TP algorithm works on the basis of the rendezvous scheme for two available channels. There are two constructed 'pointers' where fp is fixed at the first channel c_x and mp is a moving pointer that traverses the capability set back and forth. We divide the time into rounds and each round contains $L = 2T_2$ time slots (we repeat the constructed sequence from Alg. 1 twice to tackle the asynchronous situation). fp is fixed but mp changes in each round. As illustrated in Fig. 6, mp moves from the last channel c_{x+k_i-1} to the first one c_x in the first $k_i - 1$ rounds, and then from the first one to the last one in the next $k_i - 1$ rounds. The user continues the process until rendezvous.

For users a and b with $C_a = \{c_x, c_{x+1}, \dots, c_{x+k_a-1}\}$, $C_b = \{c_y, c_{y+1}, \dots, c_{y+k_b-1}\}$ where $1 \leq x \leq n - k_a + 1$, $1 \leq y \leq n - k_b + 1$, and $C_a \cap C_b \neq \emptyset$, a situation must happen with $c_x \in C_b$ or $c_y \in C_a$. Therefore, the constructed two pointers can help guarantee rendezvous when one user's moving pointer coincides with the other's fixed pointer.

Algorithm 2 Traversing Pointer Algorithm

```

1:  $t := 1, r := 1, L := 2T_2$ ;
2:  $fp := c_x, mp := c_{x+k_i-1}$ ;
3: while not rendezvous do
4:    $r := \lfloor t/L \rfloor + 1, p := (t-1)\%L + 1$ ;
5:    $r' := (r-1)\%(2(k_i-1))$ ;
6:   if  $0 \leq r' < k_i - 1$  then
7:      $mp := c_{x+k_i-1-r'}$ ;
8:   else
9:      $mp := c_{x+r'\%(k_i-1)}$ ;
10:  end if
11:  Invoke Alg. 1 with available channels  $\{fp, mp\}$  and
    repeat the output twice to construct the rendezvous
    sequence  $RS_r = \{s_1, s_2, \dots, s_L\}$ ;
12:  Access the  $p$ -th channel of the sequence  $s_p \in RS_r$ ;
13:   $t := t + 1$ ;
14: end while

```

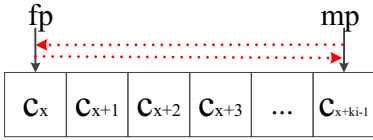


Fig. 6. There are two pointers constructed in Alg. 2. fp is fixed at the first channel in all rounds, while mp traverses the channels back and forth and round by round.

Theorem 2: Alg. 2 guarantees rendezvous for the fully available spectrum scenario in $O(\max\{|C_a|, |C_b|\} \log \log n)$ time slots.

Proof: Since the channels in the capability sets C_a and C_b are continuous and $C_a \cap C_b \neq \emptyset$, the first channel of C_a is in C_b (i.e. $c_x \in C_b$) or the first channel of C_b is in C_a (i.e. $c_y \in C_a$). Without loss of generality, suppose $c_x \in C_b$. Denote the consecutive L time slots constructed in Line 11 as a round, and the chosen available channels in the r -th round of two users are $\{fp_{a,r}, mp_{a,r}\}, \{fp_{b,r}, mp_{b,r}\}$ respectively. We say the i -th round of user a (denoted as $r_{a,i}$) overlaps with the j -th round of user b ($r_{b,j}$) if their intersection part contains at least $L/2$ time slots. From Theorem 1, if $r_{a,i}$ overlaps with $r_{b,j}$ and $\{fp_{a,i}, mp_{a,i}\} \cap \{fp_{b,j}, mp_{b,j}\} \neq \emptyset$, two users can achieve rendezvous in $L = 32(\lceil \log \log n \rceil + 1)$ time slots.

If user a starts earlier than user b , suppose the i -th round of user a overlaps with the first round of user b , after $r = y + k_b - 1 - x$ rounds, $r_{a,i+r}$ overlaps with $r_{b,1+r}$ where user b 's moving pointer chooses channel $mp_{b,1+r} = c_{y+k_b-(1+r)} = c_x = fp_{a,i+r}$, then rendezvous is guaranteed in $(r+1)L \leq |C_b|L$ time slots.

If user b starts earlier than user a , and suppose the i -th round of user b overlaps with the first round of user a , then there are two situations according to the moving direction of user b 's moving pointer (mp). It is easy to check that user b 's moving pointer chooses channel c_x within $2k_b$ rounds no matter which direction it is heading. Thus rendezvous is guaranteed in $2|C_b|L$ time slots.

Similarly, when $c_y \in C_a$, rendezvous can be guaranteed in $2|C_a|L$ time slots. Therefore, Alg. 2 guarantees rendezvous in $2 \max\{|C_a|, |C_b|\}L = O(\max\{|C_a|, |C_b|\} \log \log n)$ time slots when the spectrum is fully available. ■

B. A Constructive Lower Bound

In order to show the efficiency of the TP algorithm, we show a constructive lower bound:

Theorem 3: $\max\{|C_a|, |C_b|\}$ time slots are needed to guarantee rendezvous for the fully available spectrum condition.

Proof: Suppose user a can sense only 1 channel (i.e. $|C_a| = 1$) which belongs to C_b . In order to discover the channel for rendezvous, user b has to traverse all channels in C_b at least once and thus (at least) $\max\{|C_a|, |C_b|\}$ time slots are needed, which concludes the theorem. ■

It is clear that the lower bound still holds even two users are synchronous, and the TP algorithm is nearly optimal with only an additional $O(\log \log n)$ factor. Compared to the state-of-the-art result $O(|C_a||C_b|)$ in [26], the TP algorithm removes an $O(\min\{|C_a|, |C_b|\})$ factor and it works more efficiently.

VI. RENDEZVOUS FOR PARTIALLY AVAILABLE SPECTRUM

In practical situations, the sensed available channels may be only a fraction of the spectrum sensing capability set. For two users a and b with capability sets $C_a, C_b \subseteq U$ and available channel sets $V_a \subseteq C_a, V_b \subseteq C_b$, the TP algorithm may not guarantee rendezvous. For example, suppose the first channel of user a (c_x) belongs to user b 's capability set ($c_x \in C_b$), c_x is available for user a ($c_x \in V_a$), but it is not available for user b ($c_x \notin V_b$). The fixed pointer of user a stays at channel c_x all the time but user b cannot access c_x , and thus rendezvous may not happen. In order to overcome this, we propose a modified algorithm called Moving Traversing Pointers (MTP) in this section. The intuitive idea is to move the 'fixed pointer' after the 'moving pointer' has already traversed the channels.

A. Algorithm Description

Suppose user i has the spectrum sensing capability set as $C_i = \{c_x, c_{x+1}, \dots, c_{x+k_i-1}\} \subseteq U$ where $k_i = |C_i|$ and $1 \leq x \leq n - k_i + 1$, and the available channel set $V_i \subseteq C_i$. Order the available channels by increasing order and denote $V_i = \{c_{i,1}, c_{i,2}, \dots, c_{i,m_i}\}$ where $m_i = |V_i|$ and $\forall 1 \leq j_1 < j_2 \leq m_i, c_{i,j_1} < c_{i,j_2}$ holds. The MTP algorithm is presented in Alg. 3.

Alg. 3 is different from the TP algorithm where the 'fixed pointer' does not always stay at the same channel. Assume time is divided into loops of length $P = 2(m_i - 1)L$ time slots and each loop contains $2(m_i - 1)$ rounds of length $L = 2T_2 = 32(\lceil \log \log n \rceil + 1)$. The pointer fp stays at a fixed available channel in each loop and it moves to the next available one every P time slots as in Line 7. Similar to the TP algorithm, the 'moving pointer' stays at a fixed channel in each round and traverses the available channels back and forth and round by round. As illustrated in Fig. 7(a), fp is fixed at channel $c_{i,1}$ for the first P time slots and mp traverses from the last available channel c_{i,m_i} to the first one $c_{i,1}$, and then back to

Algorithm 3 Moving Traversing Pointers Algorithm

```

1:  $t := 1, r := 1, m_i = \lfloor V_i \rfloor$ ;
2:  $L := 2T_2, P := 2(m_i - 1)L$ ;
3:  $fp := c_{i,1}, mp := c_{i,m_i}$ ;
4: while Not rendezvous do
5:    $l := \lfloor t/P \rfloor + 1, p_1 = (t - 1) \% P + 1$ ;
6:    $r := \lfloor p_1/L \rfloor + 1, p_2 := (p_1 - 1) \% L + 1$ ;
7:    $l' := (l - 1) \% m_i + 1, fp := c_{i,l'}$ ;
8:    $r' := (r - 1) \% (2(m_i - 1)) + 1$ ;
9:   if  $0 < r' < m_i$  then
10:     $mp := c_{i,m_i+1-r'}$ ;
11:   else
12:     $mp := c_{i,r' \% (m_i - 1)}$ ;
13:   end if
14:   Invoke Alg. 1 with available channels  $\{fp, mp\}$  and
     repeat the output twice to construct the rendezvous
     sequence  $RS_{l,r} = \{s_1, s_2, \dots, s_L\}$ ;
15:   Access the  $p_2$ -th channel as  $s_{p_2} \in RS_{l,r}$ ;
16:    $t := t + 1$ ;
17: end while

```

the last one every L time slots. In the next loop of P time slots, fp moves to channel $c_{i,2}$ as Fig. 7(b) and mp repeats the traversal. This process continues until rendezvous.

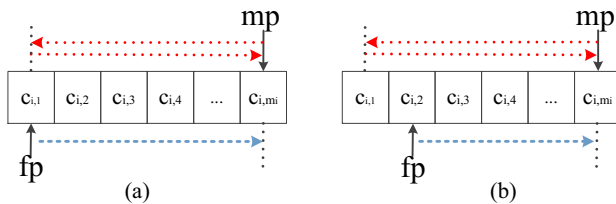


Fig. 7. There are two pointers constructed in Alg. 3. mp traverses the channels back and forth and round by round, while fp moves to the next available channel every $2(m_i - 1)$ rounds.

B. Correctness and Efficiency

Consider users a and b with capability sets $C_a, C_b \subseteq U$ and available channel sets $V_a \subseteq C_a, V_b \subseteq C_b$ where $C_a \cap C_b \neq \emptyset$. Denote $V_a = \{c_{a,1}, c_{a,2}, \dots, c_{a,m_a}\}$, $V_b = \{c_{b,1}, c_{b,2}, \dots, c_{b,m_b}\}$ where $m_a = |V_a|, m_b = |V_b|$. We show the correctness and the efficiency as follows:

Theorem 4: Alg. 3 guarantees rendezvous for the partially available spectrum scenario in $O((\max\{|V_a|, |V_b|\})^2 \log \log n)$ time slots.

Proof: Since $V_a \cap V_b \neq \emptyset$, there exist $1 \leq x \leq m_a, 1 \leq y \leq m_b$ such that $c_{a,x} = c_{b,y}$ is a common channel. Denote the consecutive L time slots constructed in Line 14 as a round and every $2(m_i - 1)$ rounds as a loop ($i = a$ or b). Denote the r -th round of l -th loop for users a and b as $r_a(l, r)$ and $r_b(l, r)$, and the chosen available channels in the round as $\{fp_a(l, r), mp_a(l, r)\}$ and $\{fp_b(l, r), mp_b(l, r)\}$ respectively. Similar to the analysis of Theorem 2, we say round $r_a(l_a, r_a)$ overlaps with round $r_b(l_b, r_b)$ if their intersection part contains at least $L/2$ time slots. From Theorem 1, if

$r_a(l_a, r_a)$ overlaps with $r_b(l_b, r_b)$ and the chosen channels satisfy $\{fp_a(l, r), mp_a(l, r)\} \cap \{fp_b(l, r), mp_b(l, r)\} \neq \emptyset$, rendezvous can be achieved in the intersection part.

Without loss of generality, assuming $m_a = |V_a| \leq |V_b| = m_b$ and we show the theorem in two cases.

If user a starts the algorithm earlier than user b , suppose $r_a(l_a, r_a)$ overlaps with the first round of user b ($r_b(1, 1)$). After $(y-1) \cdot 2(m_b - 1)$ rounds, user b 's fixed pointer (fp) stays at channel $c_{b,y}$ for the next $2(m_b - 1)$ rounds. Since $2(m_b - 1) \geq 2(m_a - 1)$, user a 's moving pointer (mp) has enough time (rounds) to traverse all available channels including $c_{a,x} = c_{b,y}$, and therefore the chosen channels overlap in $2(m_a - 1)$ rounds and rendezvous is guaranteed in $[2(m_b - 1) \cdot (y - 1) + 2(m_a - 1)] \cdot L \leq 2(m_b - 1)m_b L$ time slots.

If user b starts the algorithm earlier than user a , rendezvous is also guaranteed in $2(m_b - 1)m_b L$ time slots. Due to the page limits, we omit the details.

Similarly, if $m_a \geq m_b$, rendezvous is guaranteed in $2(m_a - 1)m_a L$ time slots. Therefore, Alg. 3 guarantees rendezvous in $2(\max\{m_a, m_b\})^2 \cdot 32(\lceil \log \log n \rceil + 1) = O((\max\{|V_a|, |V_b|\})^2 \log \log n)$ time slots. ■

Compared to the state-of-the-art result $O(|C_a||C_b|)$ in [25], the MTP algorithm works much faster when the number of available channels $|V_a|, |V_b|$ only accounts for a small fraction of $|C_a|, |C_b|$. When the spectrum is fully available, the MTP algorithm is only an $O(\log \log n)$ factor worse than that in [25], which is also acceptable.

VII. SIMULATION

In this section, we evaluate our proposed algorithms under various circumstances and compare the results with the state-of-the-art rendezvous algorithms. We choose the Heterogeneous Hopping (HH) algorithm [26] to compare with the Traversing Pointer (TP) algorithm for the fully available spectrum scenario, and the Interlock Channel Hopping (ICH) algorithm [25] to compare with the Moving Traversing Pointers (MTP) algorithm for the partially available spectrum scenario. For users a and b with capability sets $C_a, C_b \subseteq U = \{1, 2, \dots, n\}$ and available channel sets $V_a \subseteq C_a, V_b \subseteq C_b$, we simulate these algorithms under different circumstances and use the maximum time to rendezvous ($MTTR$) as the measurement. The presented results in the section are based on 1000 separate runs.

For the fully available scenario, let $C_a = \{1, 2, \dots, n/2\}$ and $C_b = \{n/2, n/2 + 1, \dots, n\}$ and thus there is only one common channel in their capabilities. When n increases from 50 to 500, Fig. 8 shows that the TP algorithm increases much slower than the HH algorithm. From Theorem 2, the TP algorithm guarantees rendezvous in $O(\max\{|C_a|, |C_b|\} \log \log n)$ time slots, which grows linearly when $|C_a|, |C_b|$ is linearly increasing, which corroborate our analysis. The worst situation for the HH algorithm is $O(|C_a||C_b|)$ which is agreed to also by the simulation result.

We evaluate the performance of the HH and the TP algorithms for the more general situation; let $C_a = \{1, 2, \dots, n/2\}$ and $C_b = \{0.1n, 0.1n + 1, \dots, 0.6n\}$ and thus the ratio

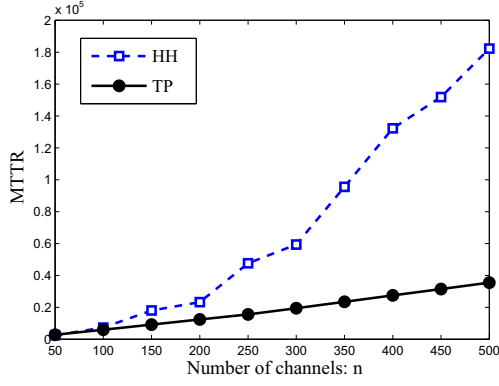


Fig. 8. Comparison between the HH algorithm and the TP algorithm when n increases from 50 to 500 and $|C_a \cap C_b| = 1$

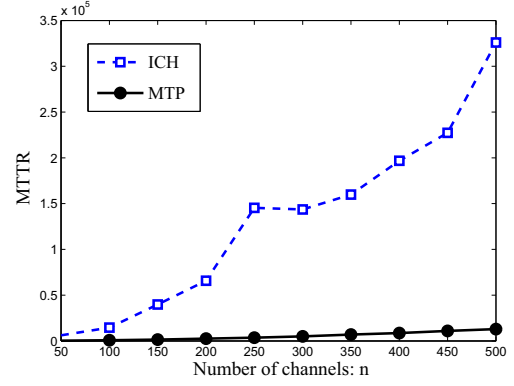


Fig. 11. Comparison between the ICH algorithm and the MTP algorithm when n increases from 50 to 500

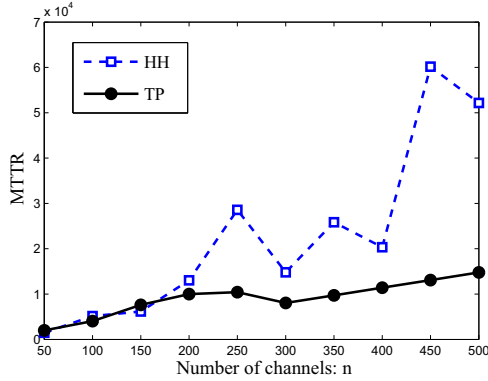


Fig. 9. Comparison between the HH algorithm and the TP algorithm when n increases from 50 to 500 and $|C_a \cap C_b| = 0.4n$

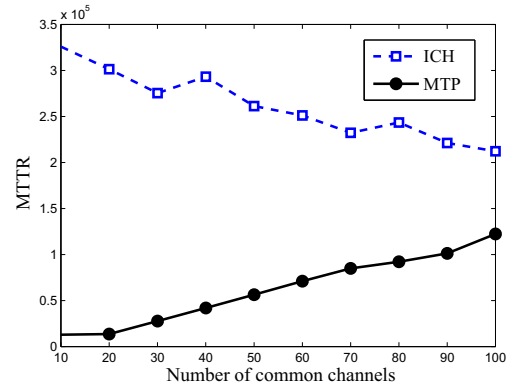


Fig. 12. Comparison between the ICH algorithm and the MTP algorithm when $n = 500$ and $|V_a \cap V_b|$ increases from 1 to 100

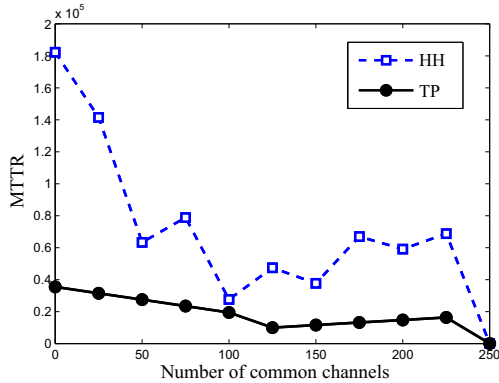


Fig. 10. Comparison between the HH algorithm and the TP algorithm when $n = 500$ and $|C_a \cap C_b|$ increases from 1 to 250

of common channels is high (80% for each set). When n increases from 50 to 500, the TP algorithm also outperforms the HH algorithm, as illustrated in Fig. 9. Similarly, the $MTTR$ value of the TP algorithm increases linearly as n increases, while the value of the HH algorithm is much larger.

To show the impact of the number of common channels,

we fix $n = 500$ and $C_a = \{1, 2, \dots, 250\}$, denote $C_b = \{x, x + 1, \dots, x + n/2\}$, and when $|C_a \cap C_b|$ increases from 1 to 250 (correspondingly, x decreases from 250 to 1), the $MTTR$ values of both HH and TP algorithms decrease as depicted in Fig. 10. The figure shows that the TP algorithm has better performance compared to the HH algorithm. When $|C_a \cap C_b| = 250$, only a constant number of time slots are needed (3 and 4 respectively) and that suits the analysis of Alg. 1 since they can achieve rendezvous in the first round by choosing the same starting channel.

For the partially available spectrum scenario, we first generate the capability sets as $C_a = \{1, 2, \dots, 0.6n\}$ and $C_b = \{0.4n, 0.4n + 1, \dots, n\}$ and define θ_g as the ratio of available channels in their intersection part ($C_g = C_a \cap C_b = \{0.4n, 0.4n + 1, \dots, 0.6n\}$) and θ_a, θ_b as the ratio of available channels in $C_a \setminus C_g, C_b \setminus C_g$ respectively. We fix $\theta_g = \theta_a = \theta_b = 0.1$; when n increases from 50 to 500, Fig. 11 shows that the MTP algorithm is much better than the ICH algorithm. This is because the MTP algorithm guarantees rendezvous in $O((\max\{|V_a|, |V_b|\})^2 \log \log n)$ time slots while the ICH algorithm guarantees rendezvous in $O(|C_a||C_b|)$ time slots. When the ratios $\theta_g, \theta_a, \theta_b$ are small, $|V_a|, |V_b|$ are much smaller

and the MTP algorithm is significantly better. In order to show the impact of the number of common channels, we set $n = 500$ and let $C_a, C_b, \theta_a, \theta_b$ be the same as Fig. 11; when the number of common channels increases from 10 to 100 (θ_g increases from 10% to 100%), Fig. 12 shows that the *MTTR* value of the MTP algorithm increases while the *MTTR* value of the ICH algorithm decreases. This is because the MTP algorithm is impacted by the number of available channels and thus the *MTTR* value increases. Differently, the ICH algorithm has more chance to rendezvous if there are more common channels. However, MTP also outperforms the ICH algorithm under the circumstance.

Through the simulation results, the TP algorithm proposed for the fully available spectrum scenario reduces the *MTTR* value significantly as compared to the HH algorithm; the MTP algorithm proposed for the partially available spectrum scenario also outperforms the ICH algorithm under most circumstances, especially when the number of available channels accounts for only a small fraction.

VIII. CONCLUSION

The rendezvous problem has attracted the attention of both the academia and the industry due to its importance in the construction of Cognitive Radio Networks (CRNs). In this paper, we study the rendezvous problem for Heterogeneous Cognitive Radio Networks (HCRNs) where the users may have different spectrum sensing capabilities. Different from most extant Chinese Remainder Theorem based constructions, we propose a new method to design rendezvous sequence which improves the state-of-the-art results. For the fully available spectrum scenario, we introduce two ‘pointers’, where the *fixed pointer* stays at the first channel of the capability set, while the *moving pointer* traverses all channels. Based on a special rendezvous scheme for two available channels, this new method guarantees rendezvous in $O(\max\{|C_a|, |C_b|\} \log \log n)$ time slots, where $C_a, C_b \subseteq U = \{1, 2, \dots, n\}$ are two capability sets. This result is significantly better than the state-of-the-art result ($O(|C_a||C_b|)$ [26]). For the partially available spectrum scenario, we propose the Moving Traversing Pointers (MTP) algorithm by modifying the fixed pointer, which guarantees rendezvous in $O((\max\{|V_a|, |V_b|\})^2 \log \log n)$ time slots ($V_a \subseteq C_a, V_b \subseteq C_b$ are the sets of available channels). The MTP algorithm works more efficiently than the ICH algorithm [25] when the number of available channels is small.

In the future, we hope to focus on deriving an explicit lower bound for the partially available spectrum scenario and designing algorithms that work efficiently under all circumstances.

IX. ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their very helpful comments which helped improve the presentation of this paper. This work was supported in part by the National Basic Research Program of China Grants 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grants 61103186, 61033001, 61361136003, and Hong Kong RGC-GRF Grant 714311.

REFERENCES

- [1] M. J. Abdel-Rahman, and M. Krunz. Game-theoretic Quorum-based Frequency Hopping for Anti-jamming Rendezvous in DSA Networks. In *DySPAN*, 2014.
- [2] K. Bian, J.-M. Park, and R. Chen. A Quorum-Based Framework for Establishing Control Channels in Dynamic Spectrum Access Networks. In *Mobicom*, 2009.
- [3] K. Bian, and J.-M. Park. Asynchronous Channel Hopping for Establishing Rendezvous in Cognitive Radio Networks. In *IEEE INFOCOM*, 2011.
- [4] K. Bian, J.-M. Park. Maximizing Rendezvous Diversity in Rendezvous Protocols for Decentralized Cognitive Radio Networks. *IEEE Transactions on Mobile Computing*, 12(7):1294-1307, 2013.
- [5] S. Chen, A. Russell, A. Samanta, and R. Sundaram. Deterministic Blind Rendezvous in Cognitive Radio Networks. In *ICDCS*, 2014.
- [6] I. Chuang, H.-Y. Wu, K.-R. Lee, and Y.-H. Kuo. Alternate Hop-and-Wait Channel Rendezvous Method for Cognitive Radio Networks. In *INFOCOM*, 2013.
- [7] L. DaSilva, and I. Guerreiro. Sequence-Based Rendezvous for Dynamic Spectrum Access. In *DySPAN*, 2008.
- [8] A. B. Flores, R. E. Guerra, and E. W. Kightly. IEEE 802.11af: A Standard for TV White Space Spectrum Sharing. *IEEE Communications Magazine*, 2013.
- [9] A.G. Fragkiadakis, E.Z. Tragos, and I.G. Askoxylakis. A survey on Security Threats and Detection Techniques in Cognitive Radio Networks. *Communications Survey & Tutorials*, 15(1): 428-455, 2013.
- [10] Z. Gu, Q.-S. Hua, Y. Wang, and F. C.M. Lau. Nearly Optimal Asynchronous Blind Rendezvous Algorithm for Cognitive Radio Networks. In *SECON*, 2013.
- [11] Z. Gu, Q.-S. Hua, and W. Dai. Local Sequence Based Rendezvous Algorithms for Cognitive Radio Networks. In *SECON*, 2014.
- [12] Z. Gu, Q.-S. Hua, Y. Wang, and F. C.M. Lau. Oblivious Rendezvous in Cognitive Radio Networks. In *SIROCCO*, 2014.
- [13] Z. Gu, Q.-S. Hua, and W. Dai. Fully Distributed Algorithms for Blind Rendezvous in Cognitive Radio Networks. In *MOBIHOC*, 2014.
- [14] J. Jia, Q. Zhang, and X. Shen. HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management. *IEEE Journal on Selected Areas in Communications*, 26(1):106-117, 2008.
- [15] Y. Kondareddy, P. Agrawal, and K. Sivalingam. Cognitive Radio Network Setup without a Common Control Channel. In *MILCOM*, 2008.
- [16] L. Lazos, S. Liu, and M. Krunz. Spectrum Opportunity-Based Control Channel Assignment in Cognitive Radio Networks. In *SECON*, 2009.
- [17] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung. Enhanced Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks. *IEEE Communications Letters*, 17(9):1742-1745, 2013.
- [18] 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*, 23(10):1867-1881, 2012.
- [19] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung. Taxonomy and Challenges of Rendezvous Algorithms in Cognitive Radio Networks. In *ICNC*, 2012.
- [20] J. Perez-Romero, O. Salient, R. Agusti, and L. Giupponi. A Novel On-Demand Cognitive Pilot Channel enabling Dynamic Spectrum Allocation. In *DySPAN*, 2007.
- [21] S. Romaszko. A Rendezvous Protocol with the Heterogeneous Spectrum Availability Analysis for Cognitive Radio Ad Hoc Networks. *Journal of Electrical and Computer Engineering*, 2013.
- [22] J. Shin, D. Yang, and C. Kim. A Channel Rendezvous Scheme for Cognitive Radio Networks. *IEEE Communications Letters*, 14(10):954-956, 2010.
- [23] C.R. Stevenson, G. Chouinard, Z. Lei, W. Hu, S.J. Shellhammer, and W. Caldwell. IEEE 802.22: The First Cognitive Radio Wireless Regional Area Network Standard. *IEEE Communications Magazine*, 47(1): 130-138, 2009.
- [24] N.C. Theis, R.W. Thomas, and L.A. DaSilva. Rendezvous for Cognitive Radios. *IEEE Transactions on Mobile Computing*, 10(2):216-227, 2011.
- [25] C.-C. Wu, and S.-H. Wu. On Bridging The Gap Between Homogeneous and Heterogeneous Rendezvous Schemes for Cognitive Radios. In *MobiHoc*, 2013.
- [26] S.-H. Wu, C.-C. Wu, W.-K. Hon, and K. G. Shin. Rendezvous for Heterogeneous Spectrum-Agile Devices. In *INFOCOM*, 2014.
- [27] Q. Zhao, and B. M. Sadler. A Survey of Dynamic Spectrum Access (signalprocessing, networking and regulatory policy). *IEEE Signal Processing Magazine*, 2007.