Fully Distributed Algorithms for Blind Rendezvous in Cognitive Radio Networks

Zhaoquan Gu
Institute for Interdisciplinary
Information Sciences
Tsinghua University
Beijing, 100084, P.R. China
demin456@gmail.com

Qiang-Sheng Hua
Institute for Interdisciplinary
Information Sciences
Tsinghua University
Beijing, 100084, P.R. China
qshua@mail.tsinghua.edu.cn

Weiguo Dai
Institute for Interdisciplinary
Information Sciences
Tsinghua University
Beijing, 100084, P.R. China
daiweiguo@yeah.net

ABSTRACT

Rendezvous process is the cornerstone to construct Cognitive Radio Networks (CRNs), through which a secondary user can establish a link for communication with its neighbor on a common channel. Although many blind rendezvous algorithms have been proposed which do not rely on a central controller or a common control channel, all of these works still rely on the global parameters such as the number of licensed channels N and the number of users. This paper aims to design fully distributed blind rendezvous algorithms only based on each user's local information. We first give the Synchronous Check & Hop (SCH) algorithm for two synchronous users where they start the rendezvous process at the same time. The SCH algorithm guarantees rendezvous in $O(\min\{k_a, k_b\}N)$ time slots where k_a, k_b are the corresponding number of sensed channels of these two users. Our main contribution is a fully distributed algorithm called Conversion Based Hopping (CBH), where each user only uses its identifier (ID) and its number of sensed channels. CBH guarantees rendezvous between two asynchronous users in $O((\max\{k_a, k_b\})^2)$ time slots. To our knowledge, this is the first result with rendezvous time independent of the global parameter N. We also derive a lower bound of rendezvous time between two users as $\Omega((k_a - k_g)(k_b - k_g))$ where k_g is the number of their common channels. All of our results also apply to a more general blind rendezvous problem which we call *Oblivious Blind Rendezvous* where each user is free to assign their local labels to the sensed channels. Extensive simulation results compared with the state-of-the-art rendezvous algorithms corroborate our theoretical analyses.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

MobiHoc'14, August 11–14, 2014, Philadelphia, PA, USA.
Copyright 2014 ACM 978-1-4503-2620-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2632951.2632981
.

Keywords

cognitive radio networks; blind rendezvous; distributed algorithm

1. INTRODUCTION

1.1 Blind Rendezvous in Cognitive Radio Networks

Cognitive Radio Network (CRN) is a promising paradigm to solve the spectrum scarcity problem [1], which consists of primary users (PUs) who own the licensed spectrum and secondary users (SUs) who can opportunistically exploit and access the portion of unused licensed spectrum without causing interference to PUs. Unless otherwise specified, 'users' mentioned hereafter in the paper refers to SUs.

Many interesting problems in CRN have been studied, such as neighbor discovery [7, 25], data gathering [5], routing [12], and broadcasting [23]. *Rendezvous* is a fundamental process of these problems where the users attempt to establish a link for communication on a common frequency band (channel) [18]. Some previous works simplify this process by adopting a central controller or a Common Control Channel (CCC), but they incur a bottleneck when an increasing number of users rely on it. Moreover, such centralization is vulnerable to adversary attacks and is not flexible in practical situations. Therefore, many *blind rendezvous* algorithms have been proposed with no centralization [6, 11, 17, 21].

More specifically, all these blind rendezvous algorithms assume the licensed spectrum is divided into N non-overlapping channels with fixed labels $\{1, 2, \ldots, N\}$, and each user can sense the channel not occupied by any nearby PUs as an *available channel.* Time is divided into slots of equal length and each user can access an available channel in each time slot. Rendezvous is achieved only when the users access the same channel in the same time slot. All the extant blind rendezvous algorithms assume they know the global parameter N and the labels of these N channels, some works [6] also assume each user knows the number of users in the network.

Nevertheless, the above assumptions may not be that practical when designing blind rendezvous algorithms. In the first place, all users may not see the same labels for the licensed channels. For example, the 'TV white space' that could be sensed by the users has operating frequencies ranging from $470 - 790$ MHz in Europe [9, 19], but it's located in the VHF (i.e. very high frequency) ($54 - 216$ MHz) and UHF (i.e. ultra high frequency) ($470 - 698$ MHz) bands in the United States [10]. Obviously, the labeling of this space could be different and the same frequency band (channel) may be assigned different labels under different administrations. Secondly, no general standard exists dividing the total licensed spectrum into N channels (for example, IEEE 802.11af only operates frequencies ranging $470 - 710$ MHz [10]), it's impractical for the users to know the N value. Thirdly, all users are physically dispersed in the network and they may join or leave freely, they cannot know the number of users beforehand as no central controller is adopted.

In this paper, we assume the scenario where there is a set of channels within some licensed spectrum that can be sensed by the users, but there is no common labeling that is seen by all users. This is a more general assumption and each user can assign their own labels to its sensed available channels. Without such a common labeling, blind rendezvous becomes much harder since the same frequency band (channel) may have different labels for two users. We call this kind of blind rendezvous as ***oblivious blind rendezvous***. All previous blind rendezvous algorithms are *non-oblivious* where the users see the same labels for the channels and it's a special case of oblivious blind rendezvous. As listed in Table 1, all previous algorithms depend on the global parameters N, M (the largest user ID) and we aim to design ***fully distributed algorithms*** for oblivious blind rendezvous where the users don't know the number of licensed channels (N) and the number of users in the CRN, i.e. only each user's local information can be used. In this paper, this kind of local information is limited to each user's identifier (ID) and the number of available channels each user has.

Oblivious Blind Rendezvous is promising in designing large scale network systems where global information is hard to obtain for the users. However, we face the following challenges in designing fully distributed algorithms for oblivious blind rendezvous. First of all, because each user may have different labels for the channels, traditional methods based on the channels' labels cannot be applied at all. Second, each user doesn't know the number of total licensed channels and the number of users, it's impossible to design global information based algorithms [11]. Third, each user can join the network at any time, and thus algorithms should guarantee the rendezvous for asynchronous users where they may start the rendezvous process at any time. Fourth, the user cannot obtain other users' (even neighbors') information until rendezvous is achieved for communication, thus ***symmetric algorithms*** are preferable, which means all users should execute the same algorithm. In this paper, our proposed algorithms address all these issues.

1.2 Contributions and Paper Organization

In this paper, we design fully distributed algorithms for oblivious blind rendezvous problem between two users and it can be extended smoothly to multiuser multihop scenarios as in [11]. In the first place, we derive a lower bound of oblivious blind rendezvous between two users as $\Omega((k_a - k_g) \cdot (k_b - k_g))$, where k_a, k_b are the number of two users' available channels respectively, and k_g is the number of channels they have in common. Following the lower bound, we introduce a deterministic distributed algorithm called Synchronous Check & Hop (SCH) for two synchronous users, which generates different hopping sequences for the users based on the distinct identifiers. Our main contribution is a fully distributed algorithm called Conversion Based Hopping (CBH) algorithm, which builds on the idea of SCH and guarantees oblivious blind rendezvous between two asynchronous users only based on the user's identifier (ID) and its number of available channels. SCH guarantees rendezvous in $O(\min\{k_a, k_b\}N)$ time slots, and the $MTTR$ value (i.e. Maximum Time to Rendezvous, defined in Section 3) for CBH is bounded by $O((\max\{k_a, k_b\})^2)$ time slots, which is comparable to the lower bound if $k_a = \Theta(k_b)$ and $k_g = o(\max\{k_a, k_b\})$. More importantly, our CBH algorithm is the first result with bounded rendezvous time independent of the global parameter N. Finally, we compare our algorithms with the state-of-the-art rendezvous algorithms through extensive simulations to validate our theoretical analyses. Note that although our algorithms are designed for oblivious blind rendezvous, our results can be applied to traditional non-oblivious blind rendezvous (Table 1) since the latter is a special case of the former problem.

Table 1: MTTR Comparison

Algorithms	Non-Oblivious Blind Rendezvous	Oblivious Blind Rendezvous
Jump-Stay [17]	$3NP^2 + 3P = O(N^3)$	–
Enhanced JS [16]	$4P^2 = O(N^2)$	–
CRSEQ [21]	$P(3P-1) = O(N^2)$	–
DRDS [11]	$3P^2 + 2P = O(N^2)$	–
AHW [6]	$O(N^2 \log M)$	–
MMC [24]	$ETTR = O(N^2)$	$ETTR = O(N^2)$
SCH (this paper)	$O(\min\{k_a, k_b\}N)$	$O(\min\{k_a, k_b\}N)$
CBH (this paper)	$O((\max\{k_a, k_b\})^2)$	$O((\max\{k_a, k_b\})^2)$

Remarks: 1) "–" means the algorithm is not applicable to oblivious blind rendezvous; 2) ETTR means expected time to rendezvous (note: MMC cannot guarantee rendezvous in bounded time); 3) P is the smallest prime number larger than N, $P = O(N)$. 4) k_a and k_b denote the numbers of two users' available channels; 5) SCH in this paper is only suitable for synchronous users.

The remainder of the paper is organized as follows. The next section highlights some related work on blind rendezvous. The system model and problem definitions are provided in Section 3. We derive the lower bound of oblivious blind rendezvous between two users in Section 4. The deterministic distributed algorithm for two synchronous users is described in Section 5, and the asynchronous algorithm is proposed in Section 6. Extensive simulations are presented in Section 7 and we conclude the paper in Section 8.

2. RELATED WORK

2.1 Non-Oblivious Rendezvous Algorithms

Most previous works aiming at non-oblivious rendezvous algorithms can be classified into three categories: centralized algorithms, decentralized algorithms based on Common Control Channel (CCC) and blind rendezvous algorithms.

Centralized algorithms assume a central controller or a CCC exists during the process to simplify the problem [14, 20]. However, the central controller or the CCC could be a bottleneck in practical situations and it's vulnerable to any adversary attacks. Some decentralized algorithms are proposed to establish local CCCs to communicate with the neighbors [13, 15]. Nevertheless, these algorithms incur too much overhead in establishing and maintaining such local CCCs.

Therefore, blind rendezvous algorithms without any CCC have been attracting the attention of many researchers. The main technique used is Channel Hopping (CH), where each user hops among the sensed available channels in different time slot on the basis of certain pre-generated CH sequence, and rendezvous can be achieved once the users hop on the same channel in the same time slot. Several state-of-the-art results are listed in Table 1.

Generated Orthogonal Sequence (GOS) [8] is a pioneering work, which generates an orthogonal sequence of length $N(N+1)$ based on the random permutations of $\{1, 2, \cdots, N\}$. However, GOS is limited to the scenario where all channels are not occupied by PUs and all users can access them. Quorum-based Channel Hopping (QCH) [2,3] works for synchronous users making using of quorum systems, while the enhanced Asynchronous QCH [4] suits for two asynchronous users, but only applicable to two channels.

Jump-Stay (JS) [17], Channel Rendezvous Sequence (CRSEQ) [21], and Disjoint Relaxed Difference Set (DRDS) [11] are three efficient blind rendezvous algorithms. JS generates a sequence of length $O(N^3)$ for each user with *jump*-pattern and *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 Enhanced JS in [16]. CRSEQ constructs a sequence of $O(N^2)$ numbers on the basis of triangle number (i.e. $T_i = \frac{i(i+1)}{2}$ when $i \in [1, N]$) and modular operations, such that the users repeating the sequence could meet on the same channel quickly. DRDS-based rendezvous algorithm is a new method guaranteeing rendezvous in $O(N^2)$ time slots, by designing a DRDS and transforming the set into a CH sequence. Moreover, a lower bound $\Omega(N^2)$ is also derived in [11] for such blind rendezvous algorithms. Alternate Hop-and-Wait (AHW) [6] generates different sequences for different users with distinct identifiers, such that the users can achieve rendezvous in $O(N^2 \log M)$ time slots where M is the maximum ID value.

2.2 Oblivious Blind Rendezvous Algorithms

Though various algorithms have been proposed for non-oblivious blind rendezvous process in CRN, very few results can be applied to oblivious blind rendezvous problem. Modified Modular Clock (MMC) [24] is the only algorithm which may achieve oblivious blind rendezvous between two users with high probability. MMC constructs a sequence based on $k \leq P \leq 2k$ and some modular operations of a rate value $r < P$, where k is the number of available channels. Nevertheless, MMC cannot guarantee bounded rendezvous. As a step forward, this papers offers fully distributed deterministic algorithms that guarantee oblivious blind rendezvous in bounded time.

3. MODEL AND PROBLEM DEFINITIONS

3.1 System Model

We study a more general blind rendezvous problem in Cognitive Radio Networks (CRNs) called oblivious blind rendezvous with m ($m \geq 2$) users, where each user has a unique identifier (ID) ranging in $[1, M]$ (here M means the maximum value for the users' ID and we assume M is bounded as $M = N^c$, where c could be an arbitrary large constant). The total licensed spectrum is assumed to be divided into N non-overlapping channels as $U = \{u_1, u_2, \cdots, u_N\}$, where u_i represents certain frequency band (e.g. $470 - 478$ MHz in TV white space). Each user is equipped with cognitive radios to sense the licensed spectrum and a frequency band (channel) is *available* to a particular user if it's not occupied by any nearby PUs. Because we consider the scenario no central entity exists, all users don't know the label for each frequency band (i.e. they don't know the set U), the number of total licensed channels N, the number of network users m and the maximum ID value M.

After the spectrum sensing, each user can figure out the available frequency bands (channels) and it labels them locally from 1 to k, where k is the number of available channels. For two different users A and B, rewrite the available channel sets as $C_a = \{c_a(1), c_a(2), \ldots, c_a(k_a)\}$ and $C_b = \{c_b(1), c_b(2), \ldots, c_b(k_b)\}$, where k_a, k_b represent the number of available channels for two users, respectively. Each channel $c_a(i) \in C_a$ or $c_b(i) \in C_b$ represents an available frequency band, but $c_a(i)$ and $c_b(i)$ do not necessarily mean the same band (see the example in Fig. 1).

Assume time is divided into slots of equal length $2t$, where t is the time duration for establishing a link for communication if the users access the same channel. According to IEEE 802.22 [22], $t = 10$ ms and thus each time slot has a fixed duration of 20 ms. The idea of setting each time slot to be $2t$ is natural because the network then can be made slot-aligned since $2t$ is sufficient to ensure an overlap of t for link establishment even if the start time of different users is not slot-aligned.

In each time slot, each user can access an available channel and attempt rendezvous with its potential neighbors. *Time to Rendezvous* (TTR) denotes the number of time slots cost to achieve rendezvous once all users have begun the process. Since the users are dispersed physically and they may begin the rendezvous process in different time slots, the proposed algorithms should be applicable to both synchronous (i.e. all users start the process at the same time) and asynchronous scenarios. We use *Maximum Time to Rendezvous* ($MTTR$) to evaluate the efficiency of rendezvous algorithms.

3.2 Problem Definition

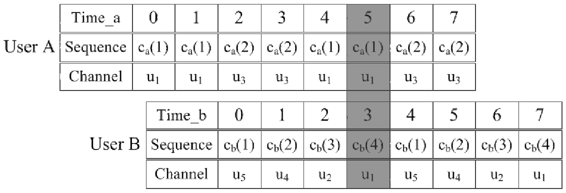
The *Oblivious Blind Rendezvous Problem* (OBRP) is defined as: Consider a multihop CRN with $m (m \geq 2)$ users where each user has a distinct ID $I \in [1, M]$. Denote the available channel set for user i as $C_i = \{c_i(1), c_i(2), \cdots, c_i(k_i)\}$ where $k_i = |C_i|$. Let $G = \bigcap_i C_i$ and $G \neq \emptyset$. Design a distributed algorithm for the users such that all users are guaranteed to rendezvous on the same channel, regardless of different time when the users begin the process.

The difference between OBRP and non-oblivious blind rendezvous problem is the users don't necessarily see the same labels for the frequency bands. All previous results are designed for non-oblivious blind rendezvous which is a special case of OBRP. Our goal is to design a fully distributed algorithm for OBRP where each user only knows its limited local information: the unique ID and the available frequency bands (channels) (in fact, the available channels set implies only the number of available channels can be used since there is no common labels seen by all users). In this paper, we focus on the rendezvous between two users (OBRP-2) and these algorithms can be extended to multiuser multihop networks as in [11, 17].

OBRP-2: Given an available channel set $C \subseteq U$ and the ID $I \in [1, M]$, design the algorithm to access channels over different time slots $t : f_{C,I}(t) \in C$ such that for any two users A and B with $C_a, C_b \subseteq U, C_a \bigcap C_b \neq \emptyset$ and ID $I_a, I_b \in [1, M], I_a \neq I_b$ respectively,

$$\forall \delta, \exists T_\delta,\ s.t.\ f_{C_a,I_a}(T_\delta + \delta) = f_{C_b,I_b}(T_\delta). \qquad (1)$$

The TTR value is T_δ when user B starts the rendezvous process δ time slots later than user A. The $MTTR$ value of the algorithm is $MTTR_f = \max_{\forall\delta} T_\delta$ and our goal is to design such algorithms with bounded $MTTR$ to guarantee rendezvous between two users.



User A	Time_a	0	1	2	3	4	5	6	7
	Sequence	$c_a(1)$	$c_a(1)$	$c_a(2)$	$c_a(2)$	$c_a(1)$	$c_a(1)$	$c_a(2)$	$c_a(2)$
	Channel	u_1	u_1	u_3	u_3	u_1	u_1	u_3	u_3

User B	Time_b	0	1	2	3	4	5	6	7
	Sequence	$c_b(1)$	$c_b(2)$	$c_b(3)$	$c_b(4)$	$c_b(1)$	$c_b(2)$	$c_b(3)$	$c_b(4)$
	Channel	u_5	u_4	u_2	u_1	u_5	u_4	u_2	u_1

Figure 1: An example of OBRP-2

Fig. 1 is an example for OBRP-2. Assume there are 5 licensed channels as: $U = \{u_1, u_2, u_3, u_4, u_5\}$, of which u_1, u_3 are available channels sensed by user A with ID $I_a = 1$ as $C_a = \{c_a(1), c_a(2)\}$:

$$c_a(1) = u_1, c_a(2) = u_3;$$

while u_1, u_2, u_4, u_5 are sensed available by user B with ID $I_b = 2$ as $C_b = \{c_b(1), c_b(2), c_b(3), c_b(4)\}$:

$$c_b(1) = u_5, c_b(2) = u_4, c_b(3) = u_2, c_b(4) = u_1$$

Consider a simple algorithm: user A repeats accessing the channels by sequence $\{c_a(1), c_a(1), c_a(2), c_a(2)\}$ while user B accesses channels by sequence $\{c_b(1), c_b(2), c_b(3), c_b(4)\}$. When user B starts the process $\delta = 2$ time slots later, rendezvous can be achieved on channel u_1 with $TTR = 4$ when $c_a(1) = c_b(4) = u_1$, as illustrated in Fig. 1.

However, it's easy to check that the above simple algorithm cannot guarantee rendezvous for all scenarios such as $\delta = 0$. Thus we aim to design deterministic distributed algorithm with bounded $MTTR$ value for both synchronous and asynchronous users.

4. LOWER BOUND FOR OBRP-2

Since the users have distinct IDs, different sequences may be generated by different users to access channels, which implies good algorithms could be made with small $MTTR$ value. However, we demonstrate that the $MTTR$ value for any deterministic algorithms could be $\Omega((k_a - k_g) \cdot (k_b - k_g))$ for the worst case situation, where k_a, k_b are the number of available channels for two asynchronous users and k_g is the number of common channels they share.

For any deterministic algorithm $\mathcal{F}$ (i.e. no randomization is involved) to OBRP-2, let a_t, b_t be the channels to access in time slot t when the users run the algorithm with inputs $(I_a, C_a), (I_b, C_b)$, respectively. It's obvious that a_t, b_t depend on the inputs and the previous results as:

$$\begin{aligned} a_t &= \mathcal{F}(a_0, a_1, \cdots, a_{t-1}, I_a, C_a) \\ b_t &= \mathcal{F}(b_0, b_1, \cdots, b_{t-1}, I_b, C_b) \end{aligned}$$

We derive the lower bound as follows:

Theorem 1. *For any deterministic algorithm $\mathcal{F}$ solving the oblivious blind rendezvous problem between two users with $(I_a, C_a), (I_b, C_b), C_a \bigcap C_b \neq \emptyset, I_a \neq I_b$, there exist mappings $f_a : C_a \mapsto U_a \subseteq U$ and $f_b : C_b \mapsto U_b \subseteq U$ such that the MTTR value is $\Omega((k_a - k_g) \cdot (k_b - k_g))$ where $k_a = |C_a|, k_b = |C_b|, k_g = |C_a \bigcap C_b|$.*

Proof. Consider the scenario when two users (A and B) start $\mathcal{F}$ at the same time, without loss of generality, let $U_g = \{u_1, u_2, \cdots, u_g\}$ be the common channels they share.

We introduce the Adversary Assignment Graph (AAG) as shown in Fig. 2. There are two rows of nodes in the graph. The number of the upper row is k_a while the other row's number of nodes is k_b, where each row represents the available channels of each user. For each time slot t, connect (a_t, b_t) in the graph if they are not connected before t, where a_t corresponds a node in the upper row and b_t is in the lower row. As shown in Fig. 2, $(c_a(1), c_b(1))$, $(c_a(1), c_b(2))$, $(c_a(2), c_b(3))$, $(c_a(3), c_b(2)), \cdots$ are connected and at most one edge is added to the graph in each time slot.

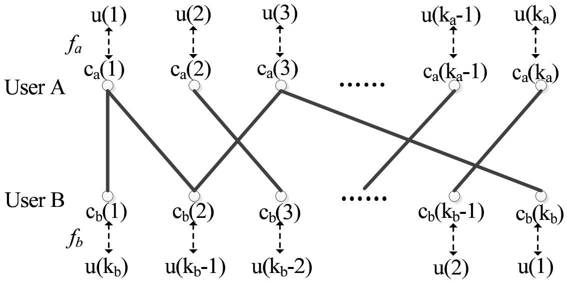


Figure 2: Adversary Assignment Graph

Assume there exists an adversary who can assign any channel $c_a(i) \in C_a$ or $c_b(j) \in C_b$ to any frequency band (channel) $u' \in U$ at any time slot t. As shown in Fig. 2, the adversary maps every channel in C_a as $f_a : c_a(i) \mapsto u(i)$ and C_b as $f_b : c_b(i) \mapsto u(k_b + 1 - i)$. Rendezvous is not achieved if there exists such an assignment that $\forall u' \in U_g$, u' in the upper row is not connected to u' in the lower row. Thus the lower bound of the $MTTR$ value is the smallest t such that for every adversary assignment, there exists $u' \in U_g$ such that (u', u') is connected in the graph.

We demonstrate that rendezvous can't be guaranteed in $t < (k_a - k_g)(k_b - k_g)$ time slots. Denote $A_t = \{a_0, a_1, \cdots, a_t\}$, $B_t = \{b_0, b_1, \cdots, b_t\}$ and construct the AAG described above. Let $\delta_a(i)$ be the degree of node $c_a(i)$ and sort these nodes of the upper row in ascending order as $c_a(1'), c_a(2'), \cdots, c_a(k'_a)$ where $\delta_a(1') \leq \delta_a(2') \leq \cdots \leq \delta_a(k'_a)$. It can be verified that $\delta_a(i') \leq (k_b - k_g), \forall 1 \leq i \leq k_g$ from the Pigeonhole Principle. Assign these k_g nodes to the universal channels in U_g as:

$$f_a : c_a(i') \mapsto u_i \in U_g, \forall 1 \leq i \leq k_g.$$

When i increases from 1 to k_g, find a node $c_b(\hat{i})$ from the lower row corresponding to node $c_a(i')$ such that $(c_a(i'), c_b(\hat{i}))$ is not connected in the graph. (Since $\delta_a(i') \leq k_b - k_g$, there are at least k_g nodes not connected to node $c_a(i')$, and at most $i - 1 < k_g$ nodes of the lower row are assigned, thus such node exists.) Then assign this node as:

$$f_b : c_b(\hat{i}) \mapsto u_i \in U_g, \forall 1 \leq i \leq k_g.$$

Finally, assign all other nodes to $U \setminus U_g$ as:

$$f_a : c_a(i') \mapsto u' \in U'_a, \forall k_g < i \leq k_a$$

$$f_a : c_b(\hat{i}) \mapsto u' \in U'_b, \forall k_g < i \leq k_b$$

where $c_b(\hat{i})$ represents the nodes haven't been assigned and $U'_a, U'_b \subseteq U \setminus U_g, U'_a \bigcap U'_b = \emptyset$. Thus such adversary assignment exists, which insinuates rendezvous in not achieved. Hence, we conclude that the $MTTR$ value for any deterministic algorithm to OBRP-2 is $\Omega((k_a - k_g) \cdot (k_b - k_g))$. □

5. SYNCHRONOUS OBLIVIOUS BLIND RENDEZVOUS

Oblivious Blind Rendezvous Problem is important and challenging, but most extant works can't guarantee rendezvous even for two synchronous users. In this section, we introduce a deterministic distributed algorithm called Synchronous Check & Hop (SCH) for two synchronous users with bounded $MTTR$ value, which works as a foundation for the fully distributed rendezvous in the next section.

5.1 Algorithm Description

Before we present the SCH algorithm, we introduce a trivial ID conversion algorithm (Alg. 1) with input ID I and the base value b. The output consists of $l+1$ bits where each bit ranges in $[0, b)$. Alg. 1 converts the user's ID to a new number under base b. For example, input $(8, 2)$ corresponds to output $(1, 0, 0, 0)$ which is the common binary representation.

Algorithm 1 ID Conversion (I, b)

```
Input: $I, b$;
Output: $d = \{d_0, d_1, \cdots, d_l\}$;
$l := \lfloor log_b I \rfloor$, $i := l$;
while $i \geq 0$ do
    $d_i := I \mod b$;
    $I := \lfloor I/b \rfloor$;
    $i := i - 1$;
end while
```

In Alg. 2, supposing each user has a unique ID $I \in [1, M]$, available channel set C, and an upper bound estimation of the number of total licensed channels $\widetilde{N} = O(N)$. It consists of two stages: *Synchronous Check Stage* and *Hop Stage*. Synchronous Check Stage generates $CT = p\widetilde{P}$ elements from Lines 9-10, where p is the smallest prime number $p \geq \max\{k, 3\}$, $k = |C|$ and $\widetilde{P}$ is the smallest prime number no less than the estimation $\widetilde{N}$. From Line 10, this stage repeats the sequence $\overrightarrow{z} = \{1, 2, \cdots, p\}$ for $\widetilde{P}$ times, which is then mapped as $\overrightarrow{z'} = \{1, 2, \cdots, k, 1, 2, \cdots, p-k\}$ in Line 16 since only k channels are available. Fig. 3 shows the process of the construction.

Hop Stage generates $HT = p^2(l+2)$ elements from Lines 12-14, where $l = \lfloor log_{p-1} I \rfloor$. It consists of $l+2$ frames and the length of each frame is $FL = p^2$. In the i-th frame, the construction of first p elements can be thought of the user hops in a circle of p nodes with label $\{1, 2, \cdots, p\}$ from $1 \to 1 + D(i) \to (2D(i)) \mod p + 1 \to (3(D(i)) \mod p + 1 \to \cdots$ from Lines 13-14. We call $D(i)$ the *hopping step* as the difference between two consecutive elements. Then the next p elements are constructed the same way by increasing the first element to 2 and holding the same hopping step $D(i)$. Thus the Hop Stage can be constructed iteratively. For example, when $D(i) = 0$,

$$\overrightarrow{z} = \{\underbrace{1, 1, \cdots, 1}_{p}, \underbrace{2, 2, \cdots, 2}_{p}, \cdots, \underbrace{p, p, \cdots, p}_{p}\}$$

and when $D(i) = 1$,

$$\overrightarrow{z} = \{\underbrace{1, 2, \cdots, p}_{p}, \underbrace{2, 3, \cdots, p, 1}_{p}, \cdots, \underbrace{p, 1, 2, \cdots, p-1}_{p}\}$$

Algorithm 2 Synchronous Check & Hop Algorithm

```
1: Input: $I, C$, an estimation $\widetilde{N}$;
2: $k := |C|$;
3: Find the smallest prime numbers $p \geq \max\{k, 3\}$, $\widetilde{P} \geq \widetilde{N}$;
4: $l := \lfloor log_{p-1} I \rfloor$;
5: Invoke ID Conversion $(I, p-1)$ and the output is $d$;
6: $D := \{d_0 + 1, d_1 + 1, \cdots, d_l + 1, 0\}$;
7: $CT := p\widetilde{P}$, $HT = p^2(l+2)$, $FL = p^2$, $t := 0$;
8: while Not rendezvous and $t < CT + HT$ do
9:     if $t < CT$ then
10:        $z = t \mod p + 1$;
11:    else
12:        $x = \lfloor (t - CT)/FL \rfloor$, $y = (t - CT) \mod FL$;
13:        $y_1 = \lfloor y/p \rfloor$, $y_2 = y \mod p$;
14:        $z = (y_1 + y_2 \cdot D(x)) \mod p + 1$;
15:    end if
16:    $z' = (z - 1) \mod k + 1$, access channel $c(z') \in C$;
17:    $t = t + 1$;
18: end while
```

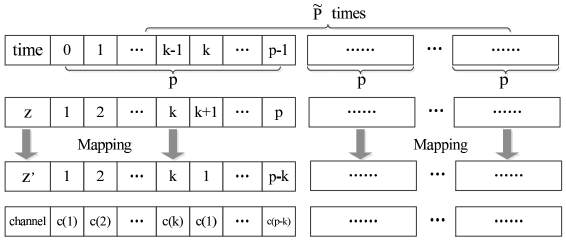


Figure 3: Construction of Synchronous Check Stage

5.2 Correctness and Time Complexity

The intuitive idea of constructing the Synchronous Check Stage follows from Lemma 5.1:

LEMMA 5.1. *Consider two vectors* $X = \{x_1, x_2, \cdots, x_m\}$, $Y = \{y_1, y_2, \cdots, y_n\}$, *if* $gcd(m, n) = 1$, *let* $\hat{X} = [\underbrace{XX\cdots X}_{n}]$ *and* $\hat{y} = [\underbrace{YY\cdots Y}_{m}]$, $\forall i \in [1, m], j \in [1, n]$, *there exists* k *such that* $\hat{X}(k) = x_i$ *and* $\hat{Y}(k) = y_j$.

PROOF. From the construction of $\hat{X}$ and $\hat{Y}$, when $k_x = i + m\theta_x, \theta_x \in [0, n)$ and $k_y = j + n\theta_y, \theta_y \in [0, m)$, $\hat{X}(k_x) = x_i$ and $\hat{Y}(k_y) = y_j$. Let $k_x = k_y$, we get:

$$i + m\theta_x = j + n\theta_y \tag{2}$$

Take modular operation on both sides to derive:

$$\begin{cases} i = j + n\theta_y \mod m \\ i + m\theta_x = j \mod n \end{cases}$$

Since $gcd(m,n) = 1$, there exist m^{-1}, n^{-1} such that $m \cdot m^{-1} = 1 \mod n$ and $n \cdot n^{-1} = 1 \mod m$, so $\theta_x = (j - i) \cdot m^{-1} \mod n$ and $\theta_y = (i - j) \cdot n^{-1} \mod m$. Thus such $k = k_x = k_y$ exists that $\hat{X}(k) = x_i$ and $\hat{Y}(k) = y_j$. □

From this lemma, for any two users with (I_a, C_a) and (I_b, C_b), if the corresponding prime numbers in Line 3 of Alg. 2 satisfy $p_a \neq p_b$, which implies $gcd(p_a, p_b) = 1$, then rendezvous is guaranteed in the Synchronous Check Stage.

Corollary 1. *For two synchronous users with (I_a, C_a) and (I_b, C_b) running Alg. 2, if $p_a \neq p_b$, they can achieve rendezvous in $T = \min\{CT_a, CT_b\} = \min\{p_a, p_b\}\widetilde{P}$ time slots, where $CT_a = p_a\widetilde{P}, CT_b = p_b\widetilde{P}$ in Line 7.*

When $p_a = p_b$, they may not rendezvous on a common channel in the Synchronous Check Stage. Thus we need Hop Stage to guarantee rendezvous under this particular situation.

Lemma 5.2. *For two synchronous users with (I_a, C_a) and (I_b, C_b) running Alg. 2, if $p_a = p_b = p$, rendezvous is guaranteed in $T = CT_a + \min\{HT_a, HT_b\} = p\widetilde{P} + (min\{l_a, l_b\} + 2) \cdot p^2$ time slots, where $l_a = \lfloor \log_{p-1} I_a \rfloor, l_b = \lfloor \log_{p-1} I_b \rfloor$.*

Proof. Denote $HT_a = p_a^2(l_a + 2), HT_b = p_b^2(l_b + 2)$ in Line 7 of Alg. 2. We show that two users with $p_a = p_b$ and $I_a \neq I_b$ can achieve rendezvous in the Hop Stage. Denote the output of ID Conversion of (I_a, p) and (I_b, p) in Alg. 1 as $d_a = \{d_{a,0}, d_{a,1}, \cdots, d_{a,l_a}\}$ and $d_b = \{d_{b,0}, d_{b,1}, \cdots, d_{b,l_b}\}$. Denote D_a, D_b in Line 6 when they run Alg. 2.

Claim 5.3. *There exists $\lambda \leq \min\{l_a, l_b\} + 1$ such that $D_a(\lambda) \neq D_b(\lambda)$.*

From the construction of D_a, D_b as Line 6, $D_a(l_a + 1) = D_b(l_b + 1) = 0$ and $\forall i \in [0, l_a], \forall j \in [0, l_b], 0 < D_a(i) < p, 0 < D_b(j) < p$. If $l_a \neq l_b$, without loss of generality, suppose $l_a < l_b$, let $\lambda = l_a + 1$, $D_a(\lambda) = 0$ but $D_b(\lambda) = d_{b,l_a} + 1 \geq 1$, thus the claim is proved. When $l_a = l_b$, we can check that there exists $0 \leq \lambda \leq l_a$ such that $d_{a,\lambda} \neq d_{b,\lambda}$ in Alg. 1 since $I_a \neq I_b$, thus $D_a(\lambda) = d_{a,\lambda} + 1 \neq d_{b,\lambda} + 1 = D_b(\lambda)$.

Suppose $C_a \bigcap C_b \neq \emptyset$, for any channel $u' \in C_a \bigcap C_b$, there exists $1 \leq i \leq k_a, 1 \leq j \leq k_b$ such that $c_a(i) = u'$ and $c_b(j) = u'$. Since two users begin the algorithm at the same time with $p_a = p_b = p$, assume they don't rendezvous in the first $T = CT + \lambda \cdot p^2$ time slots, consider the p^2 elements in the λ-th frame of the Hop Stage. For $T < t < T + p^2$, let $y_1 = \lfloor (t - T)/p \rfloor$, $y_2 = (t - T) \mod p$, the goal is to find t such that:

$$\begin{cases} (y_1 + y_2 \cdot D_a(\lambda)) \mod p + 1 = i \\ (y_1 + y_2 \cdot D_b(\lambda)) \mod p + 1 = j \end{cases} \tag{3}$$

Combining the two equations, we derive:

$$y_2 \cdot [D_a(\lambda) - D_b(\lambda)] = i - j \mod p$$

Since $D_a(\lambda) \neq D_b(\lambda)$,the modular reverse $[D_a(\lambda) - D_b(\lambda)]^{-1}$ exists which suits $[D_a(\lambda) - D_b(\lambda)] \cdot [D_a(\lambda) - D_b(\lambda)]^{-1} = 1 \mod p$, thus $y_2 = (i - j) \cdot [D_a(\lambda) - D_b(\lambda)]^{-1} \mod p$. Plug this into Equation (3) to get y_1, thus $t = T + y_1 p + y_2$. Then rendezvous is guaranteed in $CT + (\lambda + 1) \cdot p^2 = p\widetilde{P} + (\min\{l_a, l_b\} + 2) \cdot p^2$ time slots. □

Combining Corollary 1 and Lemma 5.2 to conclude:

Theorem 2. *For two synchronous users with (I_a, C_a), (I_b, C_b), if $C_a \bigcap C_b \neq \emptyset$, rendezvous can be guaranteed in $T = O(\min\{k_a, k_b\} \cdot N)$ time slots if I_a, I_b are polynomial functions of p_a, p_b, respectively.*

Remark 5.1. *The smallest prime number $\widetilde{P} \geq \widetilde{N}$ is proved to be $\widetilde{N} \leq \widetilde{P} < 2\widetilde{N}$ from the* Bertrand-Chebyshev Theorem.

6. ASYNCHRONOUS OBLIVIOUS BLIND RENDEZVOUS

In this section, we propose a fully distributed rendezvous algorithm that designs rendezvous sequences only based on the user's identifier (ID) and the number of available channels. It takes advantage of SCH's idea about generating different sequences based on the results of ID conversion. We show the correctness of our proposed algorithm with derived bounded $MTTR$ value for two asynchronous users.

6.1 Conversion Based Hopping Algorithm

Algorithm 3 Conversion Based Hopping Algorithm

```
1: Input: $I, C$;
2: $k := |C|$;
3: Find the smallest prime numbers $p \geq \max\{k, 3\}$;
4: $l := \lfloor log_{p-1} I \rfloor$;
5: Invoke ID Conversion $(I, p-1)$ and the output is $d$;
6: if $(l + 2) \mod 2 = 0$ then
7:     $l_p := l + 2$; $D := \{0, d_0 + 1, d_1 + 1, \cdots, d_l + 1\}$
8: else
9:     $l_p := l + 3$, $D := \{0, 1, d_0 + 1, d_1 + 1, \cdots, d_l + 1\}$
10: end if
11: $T := 2l_p \cdot p^2$, $FL := 2l_p \cdot p$, $SL = 2p$;
12: while Not rendezvous do
13:     $t' := t \mod T$;
14:     $x := \lfloor t'/FL \rfloor$, $x' = t' \mod FL$;
15:     $y_1 := \lfloor x'/SL \rfloor$, $y_2 = x' \mod SL$;
16:     $z := x + D(y_1) \cdot y_2 \mod p + 1$;
17:     $z' := (z - 1) \mod k + 1$, access channel $c(z') \in C$;
18:     $t = t + 1$;
19: end while
```

The Conversion Based Hopping (CBH) Algorithm is described in Alg. 3. With local input (I, C), Alg. 3 finds the smallest prime number $p \geq \max\{k, 3\}$ where $k = |C|$ and invokes ID Conversion $(I, p - 1)$ (Alg. 1) to get the results d. This preprocessing is almost the same as Alg. 2. Then it constructs the array D containing l_p elements as Lines 6-10, where l_p is defined to be an even number, which is different from Alg. 2. Following the preprocessing, Alg. 3 generates a sequence of length $T = 2l_p \cdot p^2$ as Lines 13-16. This sequence consists of p frames of equal length $FL = 2l_p \cdot p$, where each frame contains l_p segments of length $SL = 2p$. In Line 17, the sequence is mapped from $[1, p]$ to $[1, k]$ and the corresponding channel is accessed by the user.

The construction of the sequence is illustrated in Fig. 4. It consists of p frames as $\{F_0, F_1, \cdots, F_{p-1}\}$ and each frame has l_p segments as: $\{S_0, S_1, \cdots, S_{l_p-1}\}$. The way to generate S_j of F_i is to construct $2p$ elements, starting with i and the hopping step is $D(j)$, then the k-th element is $(i+kD(j)) \mod p + 1$. The reason each segment contains $2p$ elements is to eliminate the asynchronous situation through doubling

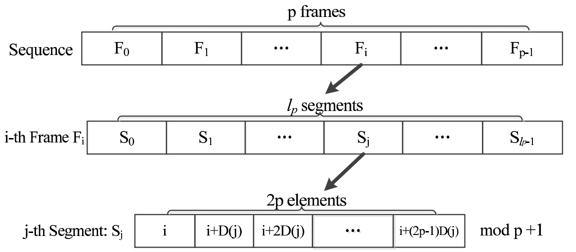


Figure 4: Construction of the sequence in Alg. 3

the length, which is similar with the method transforming time slots into slot-aligned scenarios in Section 3.

There are two intuitive ideas in designing the CBH algorithm. The first one comes from Lemma 5.1 when the corresponding prime numbers of two users in Line 3 satisfy $p_a \neq p_b$, each user repeating its own channels can guarantee rendezvous. When $p_a = p_b$, distinct IDs have different representations through ID conversion, thus accessing the channels with these hopping steps may assure rendezvous. Our proposed CBH algorithm combines these two principles and we'd like to show the correctness in the next part.

6.2 Correctness and MTTR Bound

Assume two asynchronous users (A and B) run Alg. 3 with inputs (I_a, C_a) and (I_b, C_b) where $C_a \bigcap C_b \neq \emptyset, I_a \neq I_b$. Without loss of generality, suppose user B is $\delta \geq 0$ time slots later. Denote $(k_a, p_a, l_a, l_{p_a}, D_a, T_a, FL_a, SL_a, t_a)$ and $(k_b, p_b, l_b, l_{p_b}, D_b, T_b, FL_b, SL_b, t_b)$ be the corresponding variables in Alg. 3. Since $C_a \bigcap C_b \neq \emptyset$, there exists a frequency band (channel) $u' \in C_a \bigcap C_b$ and there exist $1 \leq i \leq k_a$, $1 \leq j \leq k_b$ such that $c_a(i) = u'$ and $c_b(j) = u'$. We show the correctness and derive the $MTTR$ value from the following three cases:

1) $p_a = p_b = p$ and $l_{p_a} = l_{p_b} = l_p$;

2) $p_a = p_b = p$ but $l_{p_a} \neq l_{p_b}$;

3) $p_a \neq p_b$;

Lemma 6.1. *If $p_a = p_b = p$ and $l_{p_a} = l_{p_b} = l_p$, rendezvous between users A and B can be guaranteed in $T = 2l_p \cdot p^2$ time slots.*

Proof. If $0 \leq \delta \mod 2p < p$, there exists $x^* \geq 0, 0 \leq y_1^* < l_p, 0 \leq y_2^* < p$ such that:

$$\delta = x^* \cdot (2pl_p) + y_1^* \cdot (2p) + y_2^* \qquad (4)$$

Suppose users A and B can achieve rendezvous on channel u' at time t_a, t_b respectively, and there exists $x(a), x(b) > 0$, $0 \leq y_1(a), y_1(b) < l_p$, $0 \leq y_2(a) < 2p, 0 \leq y_2(b) < p$ such that:

$$\begin{aligned} t_a &= x(a) \cdot (2pl_p) + y_1(a) \cdot (2p) + y_2(a) && (5) \\ t_b &= x(b) \cdot (2pl_p) + y_1(b) \cdot (2p) + y_2(b) && (6) \end{aligned}$$

From Lines 13-16 of Alg. 3, the corresponding z values could be generated to be i, j, thus:

$$\begin{aligned} x(a) + D_a(y_1(a)) \cdot y_2(a) \mod p + 1 = i && (7) \\ x(b) + D_b(y_1(b)) \cdot y_2(b) \mod p + 1 = j && (8) \end{aligned}$$

Since user B is δ time slots later, i.e. $t_a = t_b + \delta$ and then plug Equations (4-6) to get:

$$\begin{aligned} [x(a) - x(b) - x^*] \cdot (2pl) + [y_1(a) - y_1(b) - y_1^*] \cdot (2p) \\ +[y_2(a) - y_2(b) - y_2^*] = 0 \end{aligned} \qquad (9)$$

Since $y_2(b) \in [0, p)$, $y_2(a) - y_2(b) - y_2^* = 0$. Combining this with Equations (7-8) to derive:

$$\begin{aligned} [D_a(y_1(a)) - D_b(y_1(b))] \cdot y_2(b) + D_a(y_1(a)) \cdot y_2^* = \\ i - x(a) - (j - x(b)) \mod p \end{aligned} \qquad (10)$$

If we can find $y_1(a), y_1(b)$ satisfying $D(y_1(a)) - D(y_1(b)) \neq 0$ and $y_1(a) - y_1(b) - y_1^* \mod l_p = 0$, Equation (10) can be solved under the constraint Equation (9). Find $y_1(a), y_1(b)$ as follows:

$$\begin{cases} y_1(a) = y_1(b) = k & \text{If } y_1^* = 0 \\ y_1(a) = y_1^*, y_1(b) = 0 & \text{If } 0 < y_1^* \leq l_p - 1 \end{cases} \qquad (11)$$

If $y_1^* = 0$, there exist $1 \leq k \leq l_p - 1$ such that $D_a(k) \neq D_b(k)$ from ID conversion. If $0 < y_1^* \leq l_p - 1$, $D_a(y_1(a)) - D_b(y_1(b)) = D_a(y_1^*) > 0$. Thus such $y_1(a), y_1(b)$ exist and $y_1(a) - y_1(b) - y_1^* = 0$.

Since $D_a(y_1(a)) - D_b(y_1(b)) \neq 0$, $y_2(b)$ can be computed from Equation (10) (pluging $x(a) - x(b) = x^*$ from the constraint Equation (9)). Then $x(b) = j - 1 - D_b(y_1(b)) \cdot y_2(b) \mod p$ and thus $x(b) \in [0, p)$. So $TTR = t_b = x(b) \cdot (2pl_p) + y_1(b) \cdot (2p) + y_2(b)$ and it's bounded by $2l_p \cdot p^2$.

For example, users A and B have inputs $I_a = 5, |C_a| = 4$, $I_b = 20, |C_b| = 5$ and $c_a(2) = c_b(4)$ is their only common channel. Thus $p_a = p_b = 5$, $l_a = l_b = 4$ and $D_a = \{0, 1, 2, 2\}$, $D_b = \{0, 2, 2, 0\}$. Let $\delta = 2014$ and it can be rewritten as: $\delta = 50 \cdot 40 + 1 \cdot 10 + 4$, thus $x^* = 50, y_1^* = 1, y_2^* = 4$ from Equation (4). Since $y_1^* = 1$, from Equation (11), $y_1(a) = y_1^* = 1, y_1(b) = 0$ and $x(a) - x(b) = x^* = 50$. From Equation (10), $y_2(b) = 4$ and $x(b) = 3$. Thus $t_b = 3 * 40 + 4 = 124$ and $t_a = t_b + \delta = 2138$. We can check that user A accesses channel $c_a(2)$ and B accesses channel $c_b(4)$ at the same time.

If $p \leq \delta \mod 2p < 2p$, the TTR value is also bounded by $2l_p \cdot p^2$ time slots using the same technique above. Thus the lemma holds. □

Lemma 6.2. *If $p_a = p_b = p$ but $l_{p_a} \neq l_{p_b}$, rendezvous between users A and B can be guaranteed in $T = 2\min\{l_{p_a}, l_{p_b}\} \cdot p^2$ time slots.*

Proof. If $0 \leq \delta \mod 2p < p$, there exists $x^* \geq 0, 0 \leq y_1^* < l_{p_a}, 0 \leq y_2^* < p$ such that:

$$\delta = x^* \cdot (2pl_{p_a}) + y_1^* \cdot (2p) + y_2^*$$

Suppose two users can rendezvous on channel u' at time t_a, t_b respectively, and:

$$\begin{aligned} t_a = x(a) \cdot (2pl_{p_a}) + y_1(a) \cdot (2p) + y_2(a) \\ t_b = x(b) \cdot (2pl_{p_b}) + y_2(b) \cdot (2p) + y_2(b) \end{aligned}$$

where $x(a), x(b) > 0, 0 \leq y_1(a) < l_{p_a}, 0 \leq y_1(b) < l_{p_b}, 0 \leq y_2(a) < 2p, 0 \leq y_2(b) < p$. Combining these with $t_a = t_b + \delta$ to derive:

$$\begin{aligned} [l_{p_a}x(a) - l_{p_b}x(b) - l_{p_a}x^* + y_1(a) - y_1(b) - y_1^*] \cdot 2p \\ +y_2(a) - y_2(b) - y_2^* = 0 \end{aligned}$$

Similarly, we have:

$$\begin{cases} l_{p_a} \cdot x(a) - l_{p_b} \cdot x(b) - l_{p_a} \cdot x^* + y_1(a) - y_1(b) - y_1^* = 0 \\ y_2(a) - y_2(b) - y_2^* = 0 \end{cases} \tag{12}$$

We also formulate Equations (7-8). If $l_{p_a} > l_{p_b}$, let $y_1(a) = 0$, $y_1(b) = k \neq 0$, $x(a) = (i-1) \mod p$ is derived from Equation (7). Plugging this into Equation (12), $l_{p_b} x(b) = [l_{p_a} x(a) - l_{p_a} x^* - y_1(b) - y_1^*] \mod p$. Since l_{p_b} is an even number, $x(b) \in [0, p)$ can be easily computed. Then from Equation (8), $y_2(b) \in [0, p)$ can be figured out. Thus the TTR value is $t_b = x(b) \cdot (2pl_{p_b}) + y_1(b) \cdot (2p) + y_2(b) \leq 2p^2 l_{p_b}$. If $l_{p_a} < l_{p_b}$, we can bound the $TTR = t_a - \delta = (x(a) - x^*) \cdot (2pl_{p_a}) + (y_1(a) - y_1^*) \cdot (2p) + (y_2(a) - y_2^*) \leq 2p^2 l_{p_a}$. Thus $MTTR \leq 2\min\{l_{p_a}, l_{p_b}\}p^2$.

If $p \leq \delta \mod 2p < 2p$, the TTR value is also bounded by $T = 2\min\{l_{p_a}, l_{p_b}\} \cdot p^2$ time slots using the same technique above. Thus the lemma holds. □

Lemma 6.3. *If $p_a \neq p_b$, rendezvous between users A and B can be guaranteed in $T = 2l_p \cdot p^2$ time slots, where $p = \max\{p_a, p_b\}$ and l_p is the corresponding value from $\{l_{p_a}, l_{p_b}\}$.*

Proof. This lemma can be concluded similarly. Suppose $p_a < p_b$, we can derive the following equations:

$$x(a) + D_a(y_1(a)) \cdot y_2(a) \mod p_a + 1 = i$$
$$x(b) + D_b(y_1(b)) \cdot y_2(b) \mod p_b + 1 = j$$

Let $y_1(b) = 0$, then $x(b) = (j-1) \mod p_b$ and $y_2(b) \in [0, 2p_b)$. Suppose $x(a) = i' \mod p_a$ and $y_1(a) \neq 0$, then $y_2(a)$ exists. Since $t_a = t_b + \delta$ and

$$t_a = x(a) \cdot (2p_a l_{p_a}) + y_1(a) \cdot (2p_a) + y_2(a)$$
$$t_b = x(b) \cdot (2p_b l_{p_b}) + y_2(b) \cdot (2p_b) + y_2(b)$$

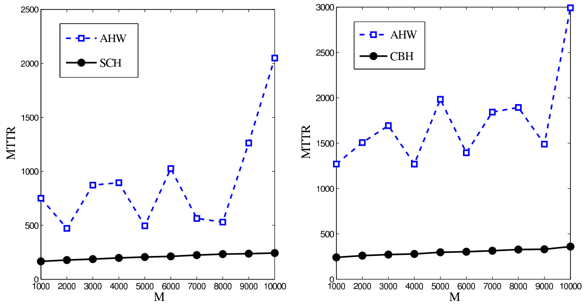
We can find such $x(a) = i' + v(a)p_a$ satisfying $\delta_b(v_b) + \delta - \delta_a(v_a) \in [2p_a - 2p_b, T_a)$ ($T_a = 2p_a^2 l_{p_a}$ is define above), where $\delta_b(v_b) = (2p_b l_{p_b}) \cdot (j - 1 + v(b)p_b)$ and $\delta_a(v_a) = (2p_a l_{p_a}) \cdot x(a)$. Obviously, $\delta_b(0) \mod T_a \geq 2p_a - 2p_b$, let $v(b) = 0$, $v(a) = \lfloor (\delta_b(0) + \delta)/T_a \rfloor$ and $i' = \lfloor (\delta_b(0) + \delta - v(a)T_a)/FL_a \rfloor$ ($FL = 2p_a l_{p_a}$), then $y_1(a), y_2(a), y_2(b)$ can be figured out. Thus $TTR = t_b = x(b) \cdot (2p_b l_{p_b}) + y_1(b) \cdot (2p_b) + y_2(b) \leq 2p_b^2 l_{p_b}$. If $p_a > p_b$, $TTR \leq 2p_a^2 l_{p_a}$ can be concluded similarly. Thus the lemma holds. □

Combine Lemmas 6.1-6.3 to conclude:

Theorem 3. *Two users running Alg. 3 can achieve rendezvous in $MTTR = 2l_p \cdot p^2$ time slots where $p = \max\{p_a, p_b\}$ and l_p is the corresponding value from $\{l_{p_a}, l_{p_b}\}$.*

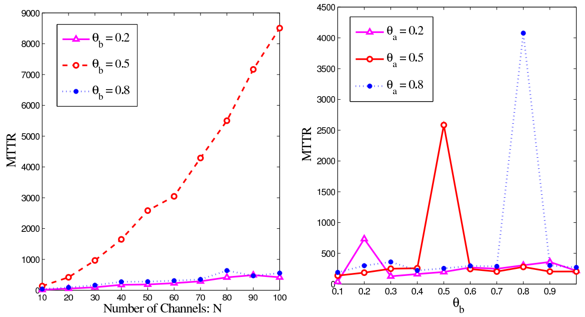
This theorem reveals that CBH can guarantee oblivious blind rendezvous between two users in short time and it is comparable to the lower bound in Section 4 for most cases. More precisely, $MTTR = 2l_p \cdot p^2 = O(k^2)$ time slots if l_p is a constant, which implies the corresponding ID is a polynomial function of p, where $k = \max\{k_a, k_b\}$. For example, if $k_a > k_b$ (thus $p_a \geq p_b$), and I_a is bounded by $I_a \leq p_a^c$ where c can be an arbitrary large constant, $MTTR = 2l_{p_a} \cdot p_a^2 = O(k_a^2)$. If $k_b = \Theta(k_a)$ and $k_g = o(k_a)$, the $MTTR$ value is comparable with the lower bound in Section 4.

Remark 6.1. *All these algorithms for OBRP-2 can be applied to OBRP. The intuitive idea is: once every two users achieve rendezvous on some common frequency band, they can exchange their local information and synchronize their labels for the available channels. Therefore, they generate the same sequence afterwards until rendezvous is achieved among all users.*



(a) $MTTR$ vs. M (b) $MTTR$ vs. M

Figure 5: $MTTR$ comparison when M increases from 1000 to 10000, $N = 10$, $k_a = k_b = 6$: (a) Synchronous scenario; (b) Asynchronous scenario;



(a) $MTTR$ vs. N (b) $MTTR$ vs. θ_b

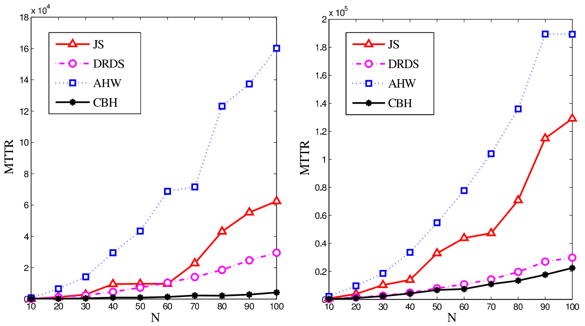
Figure 6: $MTTR$ values for SCH algorithm: (a) $\theta_a = 0.5$, $N \in [10, 100]$; (b) $N = 50$, $\theta_b \in [0.1, 1]$;

7. SIMULATION

In this section, we evaluate the performance of our proposed algorithms under various circumstances and compare the results with several state-of-the-art rendezvous algorithms. Three representative algorithms for non-oblivious blind rendezvous are chosen: Jump-Stay (JS) [17], DRDS [11], and AHW [6]. For oblivious blind rendezvous situation, we choose MMC [24] though it cannot guarantee rendezvous in bounded time.

For two users A and B, denote the corresponding identifiers (IDs) and available channel sets are (I_a, C_a) and (I_b, C_b). Denote $k_a = |C_a|, k_b = |C_b|, k_g = |C_a \bigcap C_b|$, define $\theta_a = \frac{k_a}{N}, \theta_b = \frac{k_b}{N}, \theta_g = \frac{k_g}{N}$. In each simulation, I_a, I_b are generated randomly in $[1, M]$ and the starting time of each user is random if they are asynchronous. Based on different circumstances, the available channel sets are also generated randomly. We describe the detailed parameters for the corresponding figures and these results are based on 5000 separate simulation runs.

Since AHW and our proposed algorithms are related to the users' ID, we firstly evaluate the impact of the IDs' maximum value M. Fix $N = 10, k_a = k_b = 6$, as shown in Fig.



(a) $MTTR$ vs. N (b) $MTTR$ vs. N

Figure 7: $MTTR$ comparison when N increases from 10 to 100, $M = 100$: (a) $\theta_a = \theta_b = 0.2$, $k_g = 1$; (b) $\theta_a = \theta_b = 0.5$, $k_g = 1$;

5(a), both AHW and SCH increase when M increases from 1000 to 10000 for synchronous situation, but SCH is stable and not impacted by the increasing M value largely. For asynchronous scenario, the results in Fig. 5(b) also show that our proposed CBH algorithm performs much better than AHW. In the following scenarios, we set $M = 100$.

In order to evaluate the performance of SCH for two synchronous users, fix $\theta_a = 0.5$ and evaluate the performance when N increases from 10 to 100 for three situations: $\theta_b = 0.2, 0.5, 0.8$, Fig. 6(a) shows that the $MTTR$ value is much larger when $\theta_b = 0.5$. The reason is rendezvous can be guaranteed in the Synchronous Check Stage when $\theta_b = 0.2, 0.8$, but have to achieve rendezvous in the Hop Stage when $\theta_b = \theta_a$. Fix $N = 50$ and $\theta_a = 0.2, 0.5, 0.8$ respectively, when θ_b increases from 0.1 to 1, Fig. 6(b) also shows that the $MTTR$ value is much larger than normal when $\theta_b = \theta_a$.

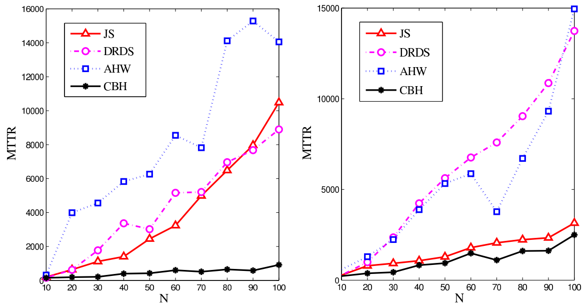
Although our proposed CBH algorithm is designed for oblivious blind rendezvous, it's also applicable to non-oblivious blind rendezvous and we evaluate the performance in Figs. 7-9.

To begin with, we evaluate the performance compared with JS, DRDS, and AHW for some extreme situations. As shown in Fig. 7(a), fix $\theta_a = \theta_b = 0.2$ and $k_g = 1$ (which means only one common channel exists), when N increases from 10 to 100, it reveals that CBH works best among these algorithms. In Fig. 7(b), CBH also outperforms the others when $\theta_a = \theta_b = 0.5$, $k_g = 1$. JS works badly because it can only guarantee rendezvous in $O(N^3)$ time slots for the worst case, while AHW is influenced by both N and M values.

In Fig. 8(a), $\theta_a = \theta_b = 0.2$, $\theta_g = 0.1$, the results show that CBH works best. In Fig. 8(b), $\theta_a = \theta_b = 0.8$, CBH also outperforms the others, but JS is also comparable to CBH. The reason is that CBH designs rendezvous algorithms based on the number of available channels and the distinct ID, when two users have less available channels, rendezvous can be achieved more quickly. JS algorithm works well when the number of available channels is large [11].

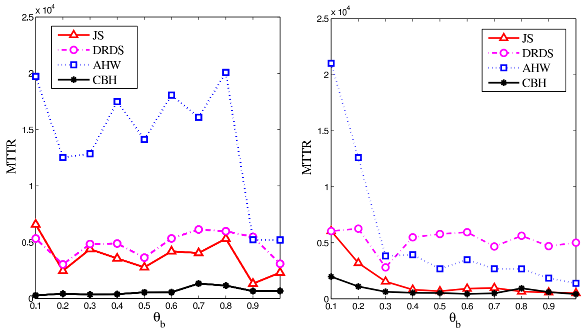
We also evaluate the situations when $k_a \neq k_b$. As shown in Fig. 9, fix $N = 50$ and $\theta_a = 0.2, 0.5$ respectively, when θ_b increases from 0.1 to 1, the results show that CBH has the smallest $MTTR$ value among these algorithms.

For oblivious blind rendezvous algorithm, we also compare the performance with MMC, though it cannot guarantee rendezvous in bounded time for some cases. As shown in Fig. 10(a), $\theta_a = \theta_b = 0.5$, CBH is stable when N increases from 10 to 100, while MMC is not stable when $k_a = k_b$. In Fig. 10(b), $\theta_a = 0.2, \theta_b = 0.8$, MMC is slighter better than CBH because the parameters insure that rendezvous can be achieved for MMC. Even though, CBH also works well and the results are comparable to the MMC algorithm.



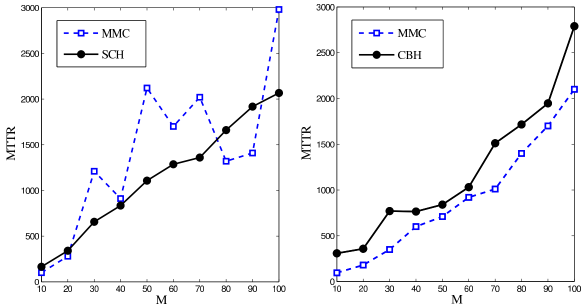
(a) $MTTR$ vs. N (b) $MTTR$ vs. N

Figure 8: $MTTR$ comparison for non-oblivious blind rendezvous when N increases from 10 to 100, $M = 100$: (a) $\theta_a = \theta_b = 0.2, \theta_g = 0.1$; (b) $\theta_a = \theta_b = 0.8$;



(a) $MTTR$ vs. θ_b (b) $MTTR$ vs. θ_b

Figure 9: $MTTR$ comparison for non-oblivious blind rendezvous when θ_b increases from 0.1 to 1, $M = 100, N = 50$: (a) $\theta_a = 0.2$; (b) $\theta_a = 0.5$;



(a) $MTTR$ vs. N (b) $MTTR$ vs. θ_b

Figure 10: $MTTR$ values for oblivious blind rendezvous when N increases from 10 to 100, $M = 100$: (a) $\theta_a = \theta_b = 0.5$; (b) $\theta_a = 0.2, \theta_b = 0.8$;

Accordingly, our proposed SCH algorithm can guarantee two synchronous users in short time when $k_a \neq k_b$ and CBH performs best among the extant non-oblivious blind rendezvous algorithms for most cases. Moreover, CBH is fully distributed based on the users' local information, therefore, it can be used to implement large network system where global information is hard to obtain and maintain.

8. CONCLUSIONS

In this paper, we study the *Oblivious Blind Rendezvous* problem in Cognitive Radio Networks (CRNs), which is more general than the non-oblivious blind rendezvous problems. In the oblivious blind rendezvous problem, all users don't see the same labels for the licensed frequency bands (channels) and thus each user is free to assign its own local label to the sensed channels.

To begin with, we derive a lower bound for oblivious blind rendezvous between two users as $\Omega((k_a - k_g)(k_b - k_g)$, where k_a, k_b are the number of their available channels and k_g is the number of common channels they share. Then, we present a fully distributed algorithm called Conversion Based Hopping (CBH) algorithm, where only the user's identifier (ID) and the number of available channels are used. CBH guarantees rendezvous between two asynchronous users in $O((\max\{k_a, k_b\})^2)$ time slots when the user's ID is a polynomial function of the number of available channels. To our knowledge, this is the first result for blind rendezvous which is independent of the global parameter N. We have also conducted extensive simulations to compare our proposed algorithms with several state-of-the-art rendezvous algorithms, and the results show that our algorithms work much better for most situations

9. ACKNOWLEDGMENTS

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61103186, 61033001, 61361136003.

10. REFERENCES

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. NeXt Generation//Dynamic Spectrum Access//Cognitive Radio Wireless Networks: A Survey. *Computer Networks*, 50(13): 2127-2159, 2006.

[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] Z. Cai, S. Ji, J. He, and A. G. Bourgeois. Optimal Distributed Data Collection for Asynchronous Cognitive Radio Networks. In *ICDCS*, 2012.

[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] Y. Dai, J. Wu, and C. Xin. Virtual Backbone Construction for Cognitive Radio Networks without Common Control Channel. In *INFOCOM*, 2013.

[8] L. DaSilva, and I. Guerreiro. Sequence-Based Rendezvous for Dynamic Spectrum Access. In *DySPAN*, 2008.

[9] ETSI. EN 301 598 White Space Devices (WSD); Wireless Access Systems Operating in the 470 MHz to 790 MHz Frequency Band. 2012.

[10] 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.

[11] 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.

[12] X. Huang, D. Lu, P. Li, and Y. Fang. Coolest Path: Spectrum Mobility Aware Routing Metrics in Cognitive Ad Hoc Networks. In *ICDCS*, 2011.

[13] 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.

[14] Y. Kondareddy, P. Agrawal, and K. Sivalingam. Cognitive Radio Network Setup without a Common Control Channel. In *MILCOM*, 2008.

[15] L. Lazos, S. Liu, and M. Krunz. Spectrum Opportunity-Based Control Channel Assignment in Cognitive Radio Networks. In *SECON*, 2009.

[16] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung. Enhanced Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks. *IEEE Communications Letters*, to appear.

[17] 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.

[18] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung. Taxonomy and Challenges of Rendezvous Algorithms in Cognitive Radio Networks. In *ICNC*, 2012.

[19] Ofcom. Regulatory Requirements for White Space Devices in the UHF TV band. http://www.cept.org/Documents/se-43/6161/, 2013.

[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] J. Shin, D. Yang, and C. Kim. A Channel Rendezvous Scheme for Cognitive Radio Networks. *IEEE Communications Letters*, 14(10):954-956, 2010.

[22] 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.

[23] J. Song, J. Xie, and X. Wang. A Novel unified Analytical Model for Broadcast Protocols in Multihop Cognitive Radio Ad Hoc Networks. *IEEE Transaction on Mobile Computing*, 2013.

[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] D. Zhang, T. He, F. Ye, R. Ganti, and H. Lei. EQS: Neighbor Discovery and Rendezvous Maintenance with Extended Quorum System for Mobile Sensing Applications. In *ICDCS*, 2012.