Oblivious Rendezvous in Cognitive Radio Networks

Zhaoquan Gu[1], Qiang-Sheng Hua[1],
Yuexuan Wang[2], and Francis Chi Moon Lau[2]

[1] Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing, P.R. China
[2] Department of Computer Science, The University of Hong Kong,
Pokfulam, Hong Kong, P.R. China

Abstract. Rendezvous is a fundamental process in the operation of a Cognitive Radio Network (CRN), through which a secondary user can establish a link to communicate with its neighbors on the same frequency band (channel). The licensed spectrum is divided into N non-overlapping channels, and most previous works assume all users have the same label for the same channel. This implies some degree of centralized coordination which might be impractical in distributed systems such as a CRN. Thus we propose *Oblivious Rendezvous* where the users may have different labels for the same frequency band.

In this paper, we study the oblivious rendezvous problem for M users (ORP-M for short) in a multihop network with diameter D. We first focus on the rendezvous process between two users (ORP-2) and then extend the derived algorithms to ORP-M. Specifically, we give an $\Omega(N^2)$ lower bound for ORP-2, and propose two deterministic distributed algorithms solving ORP-2. The first one is the ID Hopping (IDH) algorithm which generates a fixed length sequence and guarantees rendezvous in $O(N \max\{N, M\})$ time slots; it meets the lower bound when $M = O(N)$. The second one is the Multi-Step Hopping (MSH) algorithm which guarantees rendezvous in $O(N^2 \log_N M)$ time slots by combing ID scaling and hopping with different steps; it meets the lower bound if M can be bounded by a polynomial function of N, which is true of large scale networks. The two algorithms are also applicable to non-oblivious rendezvous and the performance is comparable to the state-of-the-art results. Then we extend the algorithms to ORP-M with bounded rendezvous time by increasing the diameter D by a factor.

1 Introduction

1.1 Rendezvous and Oblivious Rendezvous

Cognitive Radio Network (CRN) is attracting more and more attention in both academia and industry, which was proposed to solve the spectrum scarcity problem [1]. A CRN consists of primary users (PUs) which own the licensed spectrum and secondary users (SUs) which can sense and access the portion of the licensed

M. Halldórsson (Ed.): SIROCCO 2014, LNCS 8576, pp. 165–179, 2014.

spectrum left unused by the PUs. Unless otherwise specified, 'user' in this paper refers to SU.

There have been many interesting works in the CRN community tackling such problems as neighbor discovery [10, 27], broadcasting [16, 25], data gathering [7], and routing [15]. All these works assume one fundamental process in the operation of a CRN, called *rendezvous*, which establishes a link on some frequency band (channel) needed for communication between two or more users. One can imagine that the licensed spectrum is divided into N non-overlapping channels; each user can sense a channel, and if it is not occupied by any PU, it is an *available* channel. For the convenience of our derivations, a CRN over time is time-slotted and each user can access an available channel in each time slot. Practical rendezvous processes consist of many detailed steps, such as beaconing and handshaking. In this paper, we focus on the step of multiple users meeting on the same available channel: we say that rendezvous between users is achieved if they can access the same available channel in the same time slot. We give distributed algorithms for rendezvous. *Time to Rendezvous* (TTR) is used to measure these rendezvous algorithms, which is the time for the users to (achieve) rendezvous on a common channel.

Previous works use either a central controller (such as a base station) or a Common Control Channel (CCC) [18, 22] to simplify the process. However, such centralization could lead to a bottleneck in practical situations when the number of users increases, is vulnerable to adversary attacks, and is not flexible. Therefore, many *blind rendezvous algorithms* have been proposed, where the word 'blind' refers to non-reliance on any central controller or CCC [8, 9, 12, 13, 20, 21, 23, 26]. They construct sequences based on the channels' labels (some also use the users' identifiers) and let users hop on the frequency bands according to the sequences. Obviously, all these blind algorithms assume that the users see the same labels for these licensed frequency bands (channels). These labels represent global knowledge that must be communicated, somehow, to all the participating users. This may imply that there must exist some centralized entity that maintains and disseminates the knowledge.

To do away with the assumption of existing blind rendezous solutions that there is a common set of labels shared by all users, we propose the *oblivious rendezvous problem* where different users may have different labels for the licensed channels. Technically, each user can only assign (local) labels to those sensed available channels and attempt rendezvous based on such local information. Correspondingly, we refer to those other schemes where the users share the same labels for the frequency bands as *non-oblivious rendezvous.*

The oblivious rendezvous problem poses several challenges. First of all, because each user may have different labels for the channels, traditional methods based on a common set of channel labels cannot be applied at all. Second, each user can join the network at any time slot, and thus the algorithm needs to guarantee the rendezvous asynchronously. Third, as the users do not have each other's information until they achieve rendezvous and establish a common link for communication, symmetric algorithms are preferred, which means that all

Table 1. MTTR Comparison for Two Users' Scenario

Algorithms	Non-Oblivious Rendezvous	Oblivious Rendezvous
Jump-Stay [21]	$3NP^2 + 3P = O(N^3)$	–
CRSEQ [23]	$P(3P-1) = O(N^2)$	–
DRDS [12]	$3P^2 + 2P = O(N^2)$	–
Hop-and-Wait [9]	$O(N^2 \log M)$	–
MMC [26]	$ETTR = O(N^2)$	$ETTR = O(N^2)$
IDH (this paper)	$O(N \max\{N, M\})$	$O(N \max\{N, M\})$
MSH (this paper)	$O(N^2 \log_N M)$	$O(N^2 \log_N M)$

Remarks: 1) "–" means the method is not applicable to oblivious rendezvous; 2) ETTR means expected time to rendezvous (note: MMC cannot guarantee bounded time rendezvous); 3) P is the smallest prime number $P > N$, $P = O(N)$.

users should execute the same algorithm. Finally, for scenarios with many users in a large area, two users may not be connected directly, and so multihop communication needs to be considered. In this paper, we present algorithms that address all these issues.

1.2 Related Work

Non-oblivious rendezvous algorithms assume all users share the same labeling for the licensed channels. There are commonly three types of these algorithms: centralized algorithms, decentralized algorithms based on Common Control Channel (CCC), and blind rendezvous algorithms.

Centralized algorithms assume that a central controller or a CCC exists during the rendezvous process, which substantially simplifies the problem [18, 22]. For practical deployment, however, the central controller or the CCC could become a bottleneck and is vulnerable to adversary attacks. There are some decentralized algorithms based on establishing local CCCs through which each user can communicate with their neighbors [17, 19]. However, these algorithms incur too much overhead in establishing and maintaining local CCCs.

Blind rendezvous algorithms without CCC have been attractive to many researchers. Several state-of-the-art results are listed in Table 1; they construct a fixed length sequence for each user to hop through. Generated Orthogonal Sequence (GOS) [11] is a pioneering work which generates an $N(N+1)$-length sequence based on random permutation of $\{1, 2, \ldots, N\}$. However, it assumes that all channels are available to the users. Quorum-based Channel Hopping (QCH) [4, 5] is based on a quorum system for synchronous users. Asynchronous QCH [6] can work even when two users start in different time slots, but it is only applicable to two available channels.

Channel Rendezvous Sequence (CRSEQ) [23], Jump-Stay (JS) [21] and Disjoint Relaxed Difference Set (DRDS) [12] are three representative efficient blind rendezvous algorithms. CRSEQ picks the smallest prime $P > N$ and generates the sequence with P periods, each period containing $3P-1$ numbers based on the triangle number (triangle number means $T_i = \frac{i(i+1)}{2}$, for any $i \in [1, N]$; see [23] for details) and the modular operation. Jump-Stay uses the same idea by picking the prime number P and it generates the sequence with P periods, where each period contains two *jump* frames and one *stay* frame and each frame is P in length. DRDS is a new method we proposed in [12], through constructing a disjoint relaxed difference set and transforming it into a CH sequence of length $3P^2$; two users can achieve rendezvous in $O(N^2)$ time slots.

All these works construct the same sequence for all users, which we call *global sequence*. Correspondingly, there are several works constructing different sequences for the users, which we call *local sequences* [13]. Hop-and-Wait (HW) [9] makes use of each user's ID to construct a sequence of length $3P^2 \log m$, where m is the size of the network. Local sequences based blind rendezvous algorithms have been presented in some recent works [8,13], which favor the scenario where each user's available channels are just a small fraction of all the available channels. However, their worst case rendezvous time could still be $O(N^2 \log \log N)$ and $O(N^2)$ respectively.

Oblivious rendezvous assumes that different users have different labels for the licensed channels, which obviates the need to establish, maintain and communicate a global set of labels. Nearly all previous algorithms cannot be applied to oblivious rendezvous. To our knowledge, Modified Modular Clock (MMC) [26] is the only one that may work and achieve oblivious rendezvous for two users. MMC firstly counts the number of available channels (n) and picks a prime number $n \le P \le 2n$ randomly. Then the user generates a sequence based on P. It is claimed that using MMC, two users can achieve rendezvous within $O(N^2)$ time slots with high probability. However, it cannot guarantee bounded rendezvous. As a step forward, this paper offers deterministic distributed algorithms for bounded oblivious rendezvous.

1.3 Our Contributions

In this paper, we initiate the study of *oblivious rendezvous* in Cognitive Radio Networks. In this problem, each user has a distinct identifier (ID) within the range $[1, M]$ where M is the number of number of secondary users. First, we derive an $\Omega(N^2)$ rendezvous lower bound for any two asynchronous users by introducing the Adversary Assignment Graph, where N is the number of all licensed channels. Then, two deterministic distributed algorithms for the oblivious rendezvous problem for 2 users (ORP-2) are proposed, which subsequently serve as the building block for the cases with more users in a multihop network. The first algorithm is called ID hopping (IDH) which generates a sequence of N frames and each frame consists of $2P$ elements (P is the smallest prime larger than both N and M). We show that each user can repeat accessing the channels by the sequence and rendezvous is guaranteed in $O(N \max\{N, M\})$ time slots. The other one, called Multi-Step Hopping (MSH), is more complicated as it aims at a shorter sequence; it scales the user's ID and then hops among the channels with different steps (scaled values). MSH guarantees rendezvous in $O(N^2 \log_N M)$ time slots, which is much better than IDH, especially when the network size is large. These upper bounds match the presented lower bound if $M = O(N)$ for the IDH algorithm and $M = N^c$ (c can be an arbitrary large constant) for the MSH algorithm. We then extend these algorithms to the multiuser multihop networks with bounded time to rendezvous. Finally, we compare our algorithms with the state-of-the-art rendezvous algorithms through extensive simulations (details are in the full version [14]) which also validate our theoretical analyses.

2 Model and Problem Definitions

2.1 System Model

We consider a multihop Cognitive Radio Network (CRN) with M users (SUs) who coexist with some PUs, and the network diameter is D [1]. Each user has a distinct identifer (ID) $I \in [1, M]$. Suppose the licensed spectrum owned by the PUs is divided into $N (N \geq 1)$ non-overlapping channels where each channel represents certain frequency band (e.g., $470 - 478$ MHz in the TV white space). Each user is equipped with cognitive radios to sense the spectrum for available channels, where a channel is *available* if it is not occupied by any nearby PUs.

Through spectrum sensing, each user can obtain a set of available channels (frequency bands), and all previous blind rendezvous algorithms assume the labels of all these channels are known to all the users. We have already pointed out in the above some possible disadvantages of imposing a common set of labels. We propose the oblivious rendezvous problem where each user labels the sensed channels locally and attempts rendezvous with such local information. More specifically, we rewrite the available channel set for user a as $C_a = \{c_a(1), c_a(2), \ldots, c_a(n_a)\}$ (similarly for user b, as $C_b = \{c_b(1), c_b(2), \ldots, c_b(n_b)\}$) where $n_a = |C_a|, n_b = |C_b|$. Channel $c_a(i) \in C_a$ or $c_b(i) \in C_b$ represents a certain frequency band (channel), where i is a local label in these two users, respectively, but note that $c_a(i)$ and $c_b(i)$ may or may not be the same frequency band (Fig. 1 is an example).

Time is divided into slots of equal length of $2t$, where t is the time duration for establishing a link for communication. According to IEEE 802.22 [24], $t = 10ms$ and thus each time slot has a duration of $20ms$. Then we can consider the system slot-aligned because an overlap of t for link establishment exists even if the start times of different users are not aligned.

In each time slot, the user can access an available channel and attempt rendezvous with its potential neighbors. We use *Time to Rendezvous* (TTR) to denote the number of time slots it takes for users to achieve rendezvous once all users have begun the process. Since all users are physically dispersed and the wake-up time of each user may be different, the rendezvous algorithm should be designed to be applicable to both synchronous and asynchronous users. In this paper, we use *Maximum Time to Rendezvous* (MTTR) as a measure for the worst possible situation for the algorithms and we say rendezvous can be guaranteed if MTTR is bounded.

[1] The minimum number of hops between any two users is no larger than D.

	Time_a	0	1	2	3	4	5	6	7	9
user a	Sequence	1	2	1	2	1	2	1	2	1
	Channel	$c_a(1)$	$c_a(2)$	$c_a(1)$	$c_a(2)$	$c_a(1)$	$c_a(2)$	$c_a(1)$	$c_a(2)$	$c_a(1)$
user b	Sequence	-	1	2	3	4	1	2	3	4
	Channel	-	$c_b(1)$	$c_b(2)$	$c_b(3)$	$c_b(4)$	$c_b(1)$	$c_b(2)$	$c_b(3)$	$c_b(4)$

(a) $\delta = 1$

	Time_a	0	1	2	3	4	5	6	7	9
user a	Sequence	1	2	1	2	1	2	1	2	1
	Channel	$c_a(1)$	$c_a(2)$	$c_a(1)$	$c_a(2)$	$c_a(1)$	$c_a(2)$	$c_a(1)$	$c_a(2)$	$c_a(1)$
user b	Sequence	-	-	1	2	3	4	1	2	3
	Channel	-	-	$c_b(1)$	$c_b(2)$	$c_b(3)$	$c_b(4)$	$c_b(1)$	$c_b(2)$	$c_b(3)$

(b) $\delta = 2$

Fig. 1. An example of ORP-2 for different δ values

2.2 Problem Definition

We define the *Oblivious Rendezvous Problem (ORP)* as follows:

ORP-M: Given a multihop CRN with M users, denote the available channel set for user i as C_i and its ID as I_i. Let $G = \cap_i C_i$, and $G \neq \emptyset$. Design a strategy for the users such that they are guaranteed to hop onto the same channel in the same time slot, no matter when they begin their attempts.

In order to tackle the above problem, we first focus on designing deterministic distributed algorithms for two users' (out of the M users) rendezvous (ORP-2), and then extend these algorithms to the multiuser multihop scenario (ORP-M) (cf. Section 5).

ORP-2: Given available channel set C and ID set I, design an algorithm over time slots $t : f(t) \in [1, |C|]$ such that for any two users a and b with C_a, C_b, $C_a \cap C_b \neq \emptyset$, $I_a, I_b \in [1, M], I_a \neq I_b$, and $\forall \delta \geq 0$:

$$\exists T \ \text{ s.t. } \ c_a(f_a(T + \delta)) = c_b(f_b(T)) \in C_a \cap C_b.$$

where $f_a(T)$ (or $f_b(T)$) represents the the output when user a (or b) runs the algorithm.

The TTR value is T and user b starts the process δ time slots later than user a. The $MTTR$ value of algorithm f is $MTTR_f = max_{\forall\delta} T$. The goal is to find an algorithm f with bounded $MTTR$ and which guarantees rendezvous.

Remark 1. If user b starts the rendezvous process earlier than user a, set $\delta < 0$ in the description of ORP-2 and $TTR = T + \delta$.

Fig. 1 shows an example of ORP-2. Suppose user a has two available channels, $C_a = \{c_a(1), c_a(2)\}$ and user b has four, $C_b = \{c_b(1), c_b(2), c_b(3), c_b(4)\}$. However, only one common channel exists between them, which is $c_a(1) = c_b(4)$. Consider a simple algorithm: each user accesses the channels by repeating the sequence $\{1, 2, \ldots, n\}$ where n is the number of available channels. Thus user a repeats accessing the channels $\{c_a(1), c_a(2), c_a(1), c_a(2), \ldots\}$ until rendezvous, and similarly for user b. For the asynchronous scenario, supposing that user b starts the attempt $\delta = 1$ time slot later, rendezvous is achieved as depicted in Fig. 1(a) at time slot 4 since $c_a(1) = c_b(4)$. However, it is easy to see that the above simple algorithm cannot guarantee rendezvous for all scenarios such as when $\delta = 2$, as in Fig. 1(b). Our goal is to design deterministic distributed algorithms with bounded $MTTR$ value for all δ values.

3 Lower Bound for ORP-2

Theorem 1. *For any deterministic distributed algorithm solving ORP-2, there exist $C_a, C_b, C_a \cap C_b \neq \emptyset$ such that the MTTR value is $\Omega(N^2)$.*

Proof. For any deterministic distributed algorithm $\mathcal{F}$ on the basis of $C, I : f \mapsto [1, n]$ $(n = |C|)$, suppose users a and b have different IDs $I_a \neq I_b$ and let $|C_a| = |C_b| = \lceil N/2 \rceil$, $|C_a \cap C_b| = 1$. Equivalently, denote the only common channel between the users as c^* and there exists $1 \leq i, j \leq \lceil N/2 \rceil$ such that $c_a(i) = c_b(j) = c^*$.

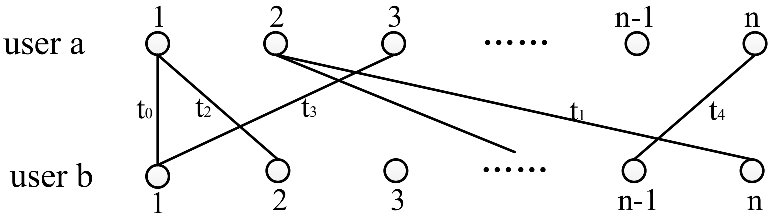

Fig. 2. Adversary Assignment Graph

We introduce the Adversary Assignment Graph (AAG), as in Fig. 2. There are two rows of nodes in the graph and the number of nodes in each row is $n = \lceil N/2 \rceil$. The upper row represents user a's local labels of the available channels with indices $\{1, 2, \ldots, n\}$ and the bottom row represents user b's labels. Let a_t, b_t be the outputs of the algorithm in time slot t, respectively, thus:

$$a_t = f(a_1, a_2, \ldots, a_{t-1}, n, I_a)$$
$$b_t = f(b_1, b_2, \ldots, b_{t-1}, n, I_b)$$

Without loss of generality, suppose user b begins δ slots later; accordingly, we connect node $a_{t+\delta}$ in the upper row with b_t in the other row with an edge having the label t (if the two nodes are already connected, then we just update the label on the edge). For example, $(1, 1)$ is connected in t_0 as depicted in Fig. 2 and $(2, n), (1, 2), (3, 1), (n, n-1)$ are also connected.

Supposing there exists an adversary who can assign licensed channels from the set $U = \{u_1, u_2, \ldots, u_N\}$ to C_a and C_b, rendezvous will not be achieved if the common channel c^* in the upper row is not connected to c^* in the lower row. Since the inputs to the algorithm $\mathcal{F}$ are fixed (for example, the inputs for user a are I_a and $|C_a|$), the lower bound of $MTTR$ is the smallest T such that (c^*, c^*) is connected in every adversary assignment.

Let δ_a be the smallest degree of the upper nodes. If $\delta_a < n$, the adversary can find a node i in the upper row and j in the lower row such that (i, j) is not connected, and then assigns c^* to them, which implies that rendezvous is not achieved. (Then it is easy to assign the other non-intersecting channels to other nodes.) We can verify that $\delta_a < n$ exists if $T < n^2$ and thus the lower bound of $MTTR$ is $n^2 = \Omega(N^2)$. Thus such C_a and C_b can be constructed by the adversary, which implies $MTTR = \Omega(N^2)$. □

4 Algorithms for ORP-2

In this section, we propose two deterministic distributed algorithms for ORP-2, which can meet the lower bound under certain conditions. The first one is based on the channel hopping method where the hopping step is based directly on the ID. The second method scales the user's ID and hops among the channels using different values.

4.1 ID Hopping Rendezvous

Alg. 1 generates a sequence of length $T = 2N\hat{P}$, which is composed of N frames and each frame contains $2\hat{P}$ elements, where $\hat{P}$ is the smallest prime number larger than both N and M. For the i-th frame ($0 \leq i < N$), the $2\hat{P}$ elements are constructed as follows (Lines 5–6): set $i+1$ to the 0-th element and $(i + j \cdot I)$ mod $\hat{P} + 1$ to the j-th element. This procedure can be thought of as picking numbers from a cycle with labels $\{0, 1, \cdots, \hat{P} - 1\}$, where the first one (the 0-th element) is $i+1$ and the second one is I steps later under the modular operation. We refer to this number as the *hopping step* and I is the hopping step in Alg. 1. Since only n available channels exist, elements in $[n+1, \hat{P}]$ are mapped to $[1, n]$ to accelerate the process, as in Line 7.

Algorithm 1. ID Hopping Algorithm

1: Find the smallest prime $\hat{P}$ such that $\hat{P} > \max\{N, M\}$;
2: $T := 2N\hat{P}$, $t := 0$, $n = |C|$;
3: **while** Not rendezvous **do**
4: $\quad t' := t \bmod T$;
5: $\quad x := \lfloor \frac{t'}{2\hat{P}} \rfloor$, $y := t' \mod 2\hat{P}$;
6: $\quad z = (x + yI) \mod \hat{P} + 1$;
7: $\quad z' = (z - 1) \mod n + 1$, access channel $c(z')$ in C;
8: $\quad t := t + 1$;
9: **end while**

For users a and b, the available channel sets are C_a, C_b and their IDs are I_a, I_b respectively. Denote the sequences generated in Alg. 1 (before mapping) as $S_a = \{a_0, a_1, \ldots, a_{T-1}\}$ and $S_b = \{b_0, b_1, \ldots, b_{T-1}\}$ where $T = 2N\hat{P}$. Without loss of generality, suppose user b is $\delta \geq 0$ time slots later than user a:

Lemma 1. *Consider sequences S_a, S_b: $\forall \delta \geq 0$ and $\forall i, j \in [1, \hat{P}]$; there exists $t < T$ such that:*

$$a_{(\delta+t) \ mod \ T} = i \ and \ b_t = j.$$

Proof. The users repeat the generated sequence every T time slots, and thus we only need to consider $0 \le \delta < T$. Let $x_1 = \lfloor \frac{\delta}{2\hat{P}} \rfloor$, $y_1 = \delta \bmod 2\hat{P}$. Two situations are analyzed on the basis of y_1:

Case 1: $0 \le y_1 < \hat{P}$. Consider $t = x_2 \cdot 2\hat{P} + y_2$, $0 \le x_2 < N, 0 \le y_2 < \hat{P}$. Let $x_2 + y_2 I_b + 1 \equiv j \bmod \hat{P}$, and thus:

$$y_2 = (j - x_2 - 1) I_b^{-1} \bmod \hat{P}. \tag{1}$$

Here $I_b^{-1}(I_b I_b^{-1} \equiv 1 \bmod \hat{P})$ exists because I_b and $\hat{P}$ are co-primes. We enumerate x_2 from 0 to $N-1$; y_2 can be computed from Eq. (1) and we denote the value as y_2^h when $x_2 = h$. Then these N values comprise the set $Y = \{y_2^0, y_2^1, \ldots, y_2^{N-1}\}$, and denote the set of corresponding time slots as $T_B = \{t_0, t_1, \ldots, t_{N-1}\}$ where $t_h = h \cdot 2\hat{P} + y_2^h$.

It is clear that $\forall t_h \in T_B$, $0 \le h < N$, $t_h < T$ and $b_{t_h} = j$. Let $T_A = \{t'_0, t'_1, \ldots, t'_{N-1}\}$ where $t'_h = (t_h + \delta) \bmod T$. Then we show that there exists $g \in [0, N)$ such that $a_{t'_g} = i$. Considering any two time slots $t'_g, t'_h \in T_A$ where user a accesses different channels:

$$a_{t'_g} = (x_1 + g) + (y_1 + y_2^g) I_a \bmod \hat{P} + 1$$
$$a_{t'_h} = (x_1 + h) + (y_1 + y_2^h) I_a \bmod \hat{P} + 1$$

Plugging in the expression of y_2^g, y_2^h as in Eq. (1), we can derive:

$$a_{t'_g} - a_{t'_h} \equiv (g - h)(I_a I_B^{-1} - 1) \ne 0 \bmod \hat{P}.$$

Here $I_a \ne I_b$, $I_a, I_b < \hat{P}$ implies $I_a I_b^{-1} \ne 1$. So $a_{t'_g} \ne a_{t'_h}$. As $|T_A| = |T_B| = N$, there are N different values for the N time slots in T_B, and thus there exists t'_g such that $a_{t'_g} = i$, which concludes the lemma.

Case 2: $\hat{P} \le y_1 < 2\hat{P}$. Consider $t = x_2 \cdot 2\hat{P} + y_2$ where $0 \le x_2 < N$ and $\hat{P} \le b_2 < 2\hat{P}$. Using the same technique as in Case 1, we can find $t < T$ such that $a_{(\delta+t) \bmod T} = i$ and $b_t = j$. Thus the lemma holds. □

Theorem 2. *Alg. 1 guarantees rendezvous between two asynchronous users of ORP-2 in* $MTTR = 2N\hat{P}$ *time slots, where* $\hat{P} \le 2\max\{N, M\}$.

Proof. Since $C_a \cap C_b \ne \emptyset$, and supposing channel $c^* \in C_a \cap C_b$, there exists $i \in [1, n_a]$ and $j \in [1, n_b]$ such that $a_i = c^*$ and $b_j = c^*$, where $n_a = |C_a|, n_b = |C_b|$. Without loss of generality, and supposing user b is δ time slots later than user a, from Lemma 1, there exists $t < T$ such that they both access channel c^*, and thus rendezvous can be guaranteed in $T = 2N\hat{P}$ time slots no matter when they start the process. □

Remark 2. P is shown to be $\hat{P} \le 2\max\{M, N\}^2$, and thus $MTTR = O(N\max(N, M))$. If $M = O(N)$ in Alg. 1, $MTTR = O(N^2)$, which meets the lower bound.

[2] *Bertrand-Chebyshev Theorem*: $\forall k > 1$, at least one prime p exists such that $k < p < 2k$.

4.2 Multi-Step Channel Hopping Rendezvous

Alg. 1 works well when $M = O(N)$. However, when the number of users increases, this algorithm becomes inefficient (for example, when $M = N^3$). The reason is that the user's ID is used as the hopping step and it enlarges the TTR when M is large. Therefore, we propose a new algorithm which is more efficient for large scale networks, by combining two techniques: ID scaling and hopping with different steps.

Algorithm 2. ID Scale Function

Input: I;
Output: $d = \{d(1), d(2), \ldots, d(l)\}$;
$l := \lfloor \log_N M \rfloor + 1$, $i := 1$, $cur(0) := I$;
while $i \leq l$ **do**
  $d(i) := cur(i-1) \bmod N + 1$;
  $cur(i) := \lfloor cur(i-1)/N \rfloor$
  $i := i + 1$;
end while

As shown in Alg. 2, the ID is scaled to $\lfloor \log_N M \rfloor + 1$ bits and each bit ranges from 1 to N[3]. For example, for $N = 8, M = 100, I = 30$, the scaled values are $d = \{7, 4, 1\}$. The scale function plays a key role in the rendezvous algorithm design and the scaled values are used as the hopping steps in Alg. 3.

Algorithm 3. Multi-Step Channel Hopping Algorithm

Find the smallest prime P such that $P > N$;
$T := 2NP$, $t := 0$, $n = |C|$, $l := \lfloor \log_N M \rfloor + 1$;
Invoke Alg. 2 on the user's ID and get the output $d = \{d(1), d(2), \ldots, d(l)\}$;
while Not rendezvous **do**
  if $t < T$ **then**
    $z := \lfloor t/2P \rfloor + 1$;
  else
    $t' := (t - T) \bmod (2lT)$;
    $x := \lfloor t'/2T \rfloor + 1$, $y := t' \bmod 2T$;
    $y_1 := y \bmod (2P)$, $y_2 := (\lfloor y/(2P) \rfloor \bmod N + 1$;
    $z := (y_2 + y_1 \cdot d(x) - 1) \bmod P + 1$;
  end if
  $z' := (z - 1) \bmod n + 1$, access channel $c(z')$ in C;
  $t := t + 1$;
end while

Alg. 3 can be thought of as generating two types of sequences. The first one is a Scale Sequence (SS) which is composed of 0 and repetitions of l scaled values

[3] Here, 'bit' does not mean 0 or 1, but represents a value in $[1, N]$.

(since two users can start the rendezvous process asynchronously, bit 0 is added as a special flag to represent the start of the user):

$$SS = \{0, \underbrace{d(1), d(2) \ldots, d(l)}_{l}, \underbrace{d(1), d(2), \ldots, d(l)}_{l}, \ldots\ldots\}$$

The other one is a Channel Hopping Sequence which is composed of different frames based on SS, as shown in Fig. 3. There are $N+1$ different types of frames, $F(0), F(1), \ldots, F(N)$, and each type of frame is composed of N segments. For example, $F(i)$ has N segments and each segment contains $2P$ elements. The 0-th element of the j-th segment is j and the k-th element is $(j + ki - 1) \bmod P + 1$ (the construction of each segment of $F(i)$ can be seen as accessing channel in $[1, P]$ by hopping i steps). For example, $F(0)$ and $F(1)$ are constructed as follows:

$$F(0) = \underbrace{1, 1, \ldots, 1}_{2P}, \underbrace{2, 2, \ldots, 2}_{2P}, \ldots, \underbrace{N, N, \ldots, N}_{2P}$$

$$F(1) = \underbrace{1, 2, \ldots, P}_{2P}, \underbrace{2, 3, \ldots, P, 1}_{2P}, \ldots, \underbrace{N, N+1, \ldots, N-1}_{2P}$$

As shown in Fig. 3, the first element 0 is special because it does not appear in other positions of SS and it corresponds to $F(0)$ once, while the other elements in SS correspond to each type of frames twice.

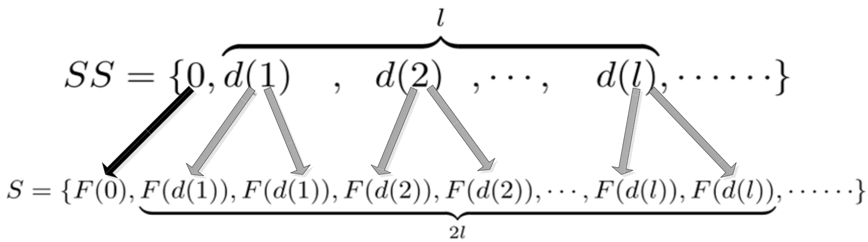

Fig. 3. Construction of Channel Hopping Sequence

Supposing users a and b run Alg. 3 with their local information (C_a, I_a) and (C_b, I_b) where $C_a \cap C_b \neq \emptyset, I_a \neq I_b$, let $n_a = |C_a|, n_b = |C_b|$, denote $d_a = \{d_a(1), d_a(2), \ldots, d_a(l)\}$, $d_b = \{d_b(1), d_b(2), \ldots, d_b(l)\}$ as the outputs of ID Scale function, denote SS_a, SS_b as the scale sequences (as constructed above), and denote $S_a = \{a_0, a_1, \ldots, a_t, \ldots\}$, $S_b = \{b_0, b_1, \ldots, b_t, \ldots\}$ as the Channel Hopping Sequences. Without loss of generality, suppose user b starts the process $\delta \geq 0$ time slots later than user a. we have the following Lemmas 2, 3 and 4. Due to the lack of space, the proofs are included only in the full version [14].

Lemma 2. *Consider SS_a, SS_b: $\forall \delta' \in Z$, there exists $i \geq 0, i + \delta' \geq 0$ such that:*

$$SS_a(i) \neq SS_b(i + \delta)$$

Lemma 3. *Consider S_a, S_b; for any pair (i, j) where $1 \leq i \leq n_a$, $1 \leq j \leq n_b$, if $0 \leq \delta < T$,*

$$\exists t \leq 2lT \quad s.t. \quad a_{(\delta+t)} = i \text{ and } b_t = j.$$

Lemma 4. *Consider S_a, S_b, for any pair (i, j) where $1 \leq i \leq n_a$, $1 \leq j \leq n_b$, if $\delta \geq T$,*

$$\exists t \leq T \quad s.t. \quad a_{(\delta+t)} = i \text{ and } b_t = j.$$

Theorem 3. *Alg. 3 guarantees rendezvous between two asynchronous users of ORP-2 in $MTTR = 4lNP = O(N^2 \log_N M)$ time slots, where $P \leq 2N$.*

Proof. As assumed, $G = C_a \cap C_b \neq \emptyset$, supposing $c^* \in G$ and there exists $1 \leq i \leq n_a$, $1 \leq j \leq n_b$ such that $c_a(i) = c^*, c_b(j) = c^*$. Without loss of generality, suppose user b starts the process δ time slots later. If $\delta < T$, from Lemma 3, rendezvous is guaranteed in $2lT$ time slots; if $\delta \geq T$, rendezvous is guaranteed in T time slots. Thus $MTTR \leq 2lT = 4lNP = O(N^2 \log_N M)$. □

Generally speaking, if M is (bounded by) a polynomial function of the total number of licensed channels N, the length of scaled bits is a constant and two users can be guaranteed to rendezvous in $O(N^2)$ time slots, which meets the lower bound of ORP-2. Moreover, this result is also comparable to even state-of-the-art non-oblivious rendezvous algorithms as shown in Table 1.

5 Algorithm for ORP-M

The algorithms for ORP-2 can be smoothly extended to handle ORP-M. We use the basic idea in [9,12,21]: once every two users achieve rendezvous on a common channel successfully, they can exchange their information over the channel and the local information such as the user's ID and the labels for the frequency bands (channels) can be synchronized. Therefore, they would generate the same sequence afterwards. We extend Alg. 3 to the multiuser multihop scenario as an example.

Algorithm 4. Algorithm for Multiuser Multihop Scenario

while Not terminated **do**
  Run Alg. 3 with local information (I, C);
  if Rendezvous with user - (I', C') **then**
    $I := \min(I, I')$;
    $C := C \cap C'$;
    Synchronize labels for the channels as the user with smaller ID;
  end if
end while

In Alg. 4, the user runs Alg. 3 with local information (I, C). Once rendezvous is achieved with another user with (I', C'), they exchange their information and three operations are executed:

- Change I to be the smaller value between I, I';
- Change C to be the intersection of C and C';
- Synchronize the labels for the available channels with the user with smaller I value such that $\forall i \in [1, |C|]$, $c(i) = c'(i)$.

After these three steps, the local information of the two users are the same and they access the channels with the same sequence until rendezvous with others. Supposing that the network diameter of the CRN in ORP-M is D, the $MTTR$ value can be guaranteed as in Theorem 4 (pleas refer to [14] for the proof).

Theorem 4. *Alg. 4 guarantees that all users can achieve rendezvous in* $MTTR = 4lNPD = O(N^2 D \log_N M)$ *time slots, where* D *is the diameter of the CRN.*

6 Oblivious Rendezvous Applications

Oblivious rendezvous is not only practical in a Cognitive Radio Network (CRN), but also suitable for several other (theoretical) problems. For example, the telephone coordination problem [2]: there are n telephones in each of two rooms, where the telephones are connected pairwisely by some unknown rules. Each room has a player who can pick up one telephone and say 'hello' in each time slot until they hear each other. They do not have any common labels of the telephones by which they can coordinate, and the aim is to minimize the time slots required for the players to meet. This problem only considers two synchronous users and each has exactly n telephones. In our settings, once each user is assigned a distinct identifier, a deterministic algorithm for this problem can be designed even for asynchronous users and some of these telephones are broken. Another problem is rendezvous search on the graph [3], where different users are placed on the graph and they attempt to meet each other as quickly as possible. Our oblivious rendezvous problem is a little different as we can consider the users in the CRN being restricted to walk in a given clique (the set of available channels), and thus the time to rendezvous can be easily extended. For other more general rendezvous search problems, the method in this paper could be used as a basis for their study.

7 Conclusion

We introduce the oblivious rendezvous problem which is believed to be more practical in constructing Cognitive Radio Networks. In contrast to existing, non-oblivious rendezvous problem, the users in our setting have different labels for the licensed frequency bands (channels), and we derive rendezvous algorithms that is based on each user's local information.

For oblivious rendezvous, we first derive an $\Omega(N^2)$ rendezvous time lower bound. Then we propose two deterministic distributed algorithms: the ID Hopping (IDH) algorithm which can achieve rendezvous between two users in $O(N \max(M, N))$ time slots, where M is number of users in the network; and the Multi-Step channel Hopping (MSH) algorithm which guarantees oblivious rendezvous in $O(N^2 \log_N M)$ time slots. The IDH algorithm works efficiently when M is small, while the MSH algorithm performs much better for larger M, which implies large scale networks with many users. The upper bounds of two algorithms match the presented lower bound if $M = O(N)$ for the IDH algorithm and if $M = N^c$ (c is a constant) for the MSH algorithm. Third, we extend these two algorithms to multiuser multihop networks. We have conducted extensive simulations for both two-user rendezvous and multihop multiusers rendezvous using our algorithms (details in the full version [14]).

Although our algorithms are designed for oblivious rendezvous, the simulation results show that they are comparable to the state-of-the-art non-oblivious rendezvous algorithms and they even perform much better under some circumstances. For oblivious rendezvous, our two proposed algorithms also outperform the MMC algorithm, and the MSH algorithm performs the best as the number of rendezvous users increases.

N, M are the number of licensed channels and users in the network respectively; one future direction is to design fully distributed rendezvous algorithms without knowing these values. We also want to explore randomized distributed algorithms which can achieve bounded rendezvous time with high probability.

Acknowledgement. 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 Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61103186, 61033001, 61361136003, Hong Kong RGC-GRF grant 714311, and the Shu Shengman Special Research Fund.

References

1. Akyildiz, I., Lee, W., Vuran, M., Mohanty, S.: NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. Computer Networks 50(13), 2127–2159 (2006)
2. Alpern, S., Pikounis, M.: The Telephone Coordination Game. Game Theory Appl. 5, 1–10 (2000)
3. Anderson, E.J., Weber, R.R.: The Rendezvous Problem on Discrete locations. Journal of Applied Probability 28, 839–851 (1990)
4. Bian, K., Park, J.-M., Chen, R.: A Quorum-Based Framework for Establishing Control Channels in Dynamic Spectrum Access Networks. In: Mobicom (2009)
5. Bian, K., Park, J.-M.: Asynchronous Channel Hopping for Establishing Rendezvous in Cognitive Radio Networks. In: IEEE INFOCOM (2011)
6. Bian, K., Park, J.-M.: Maximizing Rendezvous Diversity in Rendezvous Protocols for Decentralized Cognitive Radio Networks. IEEE Transactions on Mobile Computing 12(7), 1294–1307 (2013)
7. Cai, Z., Ji, S., He, J., Bourgeois, A.G.: Optimal Distributed Data Collection for Asynchronous Cognitive Radio Networks. In: ICDCS (2012)
8. Chen, S., Russell, A., Samanta, A., Sundaram, R.: Deterministic Blind Rendezvous in Cognitive Radio Networks. In: ICDCS (2014)

9. Chuang, I., Wu, H.-Y., Lee, K.-R., Kuo, Y.-H.: Alternate Hop-and-Wait Channel Rendezvous Method for Cognitive Radio Networks. In: INFOCOM (2013)
10. Dai, Y., Wu, J., Xin, C.: Virtual Backbone Construction for Cognitive Radio Networks without Common Control Channel. In: INFOCOM (2013)
11. DaSilva, L., Guerreiro, I.: Sequence-Based Rendezvous for Dynamic Spectrum Access. In: DySPAN (2008)
12. Gu, Z., Hua, Q.-S., Wang, Y., Lau, F.C.M.: Nearly Optimal Asynchronous Blind Rendezvous Algorithm for Cognitive Radio Networks. In: SECON (2013)
13. Gu, Z., Hua, Q.-S., Dai, W.: Local Sequence Based Rendezvous Algorithms for Cognitive Radio Networks. In: SECON (2014)
14. Gu, Z., Hua, Q.-S., Wang, Y., Lau, F.C.M.: Oblivious Rendezvous in Cognitive Radio Networks, `http://i.cs.hku.hk/~qshua/sirocco2014full.pdf`
15. Huang, X., Lu, D., Li, P., Fang, Y.: Coolest Path: Spectrum Mobility Aware Routing Metrics in Cognitive Ad Hoc Networks. In: ICDCS (2011)
16. Ji, S., Beyah, R., Cai, Z.: Minimum-Latency Broadcast Scheduling for Cognitive Radio Networks. In: SECON (2013)
17. Jia, J., Zhang, Q., Shen, X.: HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management. IEEE Journal on Selected Areas in Communications 26(1), 106–117 (2008)
18. Kondareddy, Y., Agrawal, P., Sivalingam, K.: Cognitive Radio Network Setup without a Common Control Channel. In: MILCOM (2008)
19. Lazos, L., Liu, S., Krunz, M.: Spectrum Opportunity-Based Control Channel Assignment in Cognitive Radio Networks. In: SECON (2009)
20. Lin, Z., Liu, H., Chu, X., Leung, Y.-W.: Enhanced Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks. IEEE Communications Letters 17(9), 1742–1745 (2013)
21. Liu, H., Lin, Z., Chu, X., Leung, Y.-W.: Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks. IEEE Transactions on Parallel and Distributed Systems 23(10), 1867–1881 (2012)
22. Perez-Romero, J., Salient, O., Agusti, R., Giupponi, L.: A Novel On-Demand Cognitive Pilot Channel enabling Dynamic Spectrum Allocation. In: DySPAN (2007)
23. Shin, J., Yang, D., Kim, C.: A Channel Rendezvous Scheme for Cognitive Radio Networks. IEEE Communications Letters 14(10), 954–956 (2010)
24. Stevenson, C.R., Chouinard, G., Lei, Z., Hu, W., Shellhammer, S.J., Caldwell, W.: IEEE 802.22: The First Cognitive Radio Wireless Regional Area Network Standard. IEEE Communications Magazine 47(1), 130–138 (2009)
25. Song, J., Xie, J., Wang, X.: A Novel unified Analytical Model for Broadcast Protocols in Multihop Cognitive Radio Ad Hoc Networks. IEEE Transaction on Mobile Computing (2013)
26. Theis, N.C., Thomas, R.W., DaSilva, L.A.: Rendezvous for Cognitive Radios. IEEE Transactions on Mobile Computing 10(2), 216–227 (2011)
27. Zhang, D., He, T., Ye, F., Ganti, R., Lei, H.: EQS: Neighbor Discovery and Rendezvous Maintenance with Extended Quorum System for Mobile Sensing Applications. In: ICDCS (2012)