# Local Sequence Based Rendezvous Algorithms for Cognitive Radio Networks

Zhaoquan Gu*, Qiang-Sheng Hua*, and Weiguo Dai*
*Institute for Theoretical Computer Science, Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing, 100084, P.R. China

***Abstract*—Rendezvous process plays an important role in constructing Cognitive Radio Networks (CRNs), through which a user establishes a link on a common licensed channel for communication with its neighbors. Generally, the licensed spectrum is divided into $N$ channels and most blind rendezvous algorithms are realized by the "channel hopping" method where each user repeats a *Global Sequence* constructed on top of all the $N$ channels. This global sequence based method may contain lots of redundant channels resulting in large rendezvous time especially when the number of available channels each user has only accounts for a small fraction of all the $N$ channels. In this paper, we introduce the *Local Sequence* based rendezvous algorithms where the local sequence is only constructed on top of each user's available channels and different user's local sequence could be different. Our first local sequence based algorithm called $LS$ can guarantee rendezvous in $O(N)$ time slots for symmetric users (both users have the same set of available channels) and in $O(N^2)$ time slots for asymmetric users, which matches the best known results [11]. Our major contribution is the Modified Local Sequence (MLS) based algorithm which can guarantee an exponentially shorter rendezvous time than the best known results when the number of available channels each user has is relatively small. Extensive simulation results comparing with the state-of-the-art rendezvous algorithms corroborate our theoretical analyses.**

***Index Terms*—Rendezvous, Time to Rendezvous, Local Sequence, Cognitive Radio Network**

## I. Introduction

The wireless spectrum has become very precious and scarce with the increasing demand for wireless services. The unlicensed spectrum has been overcrowded, while the utilization of licensed spectrum is pretty low. For example, the Industrial, Scientific and Medical (ISM) band is free for all users and they can resolve any interference problems in the band, causing the ISM band overcrowded with the increasing number of users, such as cordless phone and Bluetooth [10]. However, several licensed bands are currently underutilized. For example, frequencies from $470-698$ MHz are allocated to TV broadcasting in the United States but the utilization ranges from $15\%$ to $85\%$ [8], [25]. Cognitive Radio Network (CRN) is thus proposed to solve the spectrum scarcity problem where the secondary users (SUs), i.e. unlicensed devices are allowed to share the licensed spectrum causing no interference to the primary users (PUs), i.e. licensed owners. IEEE 802.22 [21] and IEEE 802.11af [9] are such two ongoing standards in spectrum sharing. Unless otherwise specified, "users" in this paper refer to SUs.

978-1-4799-4657-0/14/$31.00 © 2014 IEEE

TABLE I
MTTR Comparisons for blind rendezvous algorithms

| Algorithms | Symmetric | Asymmetric |
|---|---|---|
| GOS [7] | $N(N+1)=O(N^2)$ | $-$ |
| DRSEQ [23] | $2N+1=O(N)$ | $-$ |
| Jump-Stay [18] | $3P=O(N)$ | $3NP^2+3P=O(N^3)$ |
| CRSEQ [20] | $P(3P-1)=O(N^2)$ | $P(3P-1)=O(N^2)$ |
| DRDS [11] | $3P=O(N)$ | $3P^2=O(N^2)$ |
| AHW [5] | $3P\log M=O(N\log N)$ | $3P^2\log M=O(N^2\log N)$ |
| LS (this paper) | $2(l+1)P=O(N)$ | $2(l+1)P^2=O(N^2)$ |

Remarks: 1) The comparisons are based on the rendezvous process between two users; 2)"$-$" means DRSEQ and GOS are inapplicable to asymmetric users; 3)$P$ is the smallest prime number $P \geq N$, $P=O(N)$; 4) $l$ is a constant defined in Alg. 1.

Many interesting works have been studied in CRN, such as neighbor discovery [6], [24], data collection [4], broadcast [13], [22], and routing [12]. The fundamental process of constructing the CRN is to establish a link on a common channel for communication, which is referred to as the process of *rendezvous*. More specifically, the licensed spectrum is supposed to be divided into $N$ channels and the users are equipped with cognitive radios to sense the status of these channels. The users can access an available channel to attempt rendezvous at any time, where *available* means the channel is not occupied by any nearby PU. Rendezvous is assumed to be achieved once the users access the same channel at the same time for a period, without considering the practical implementation such as beaconing and handshaking. *Time to rendezvous* ($TTR$) is used to measure the efficiency of the rendezvous algorithms, which denotes the time cost during this process. The spectrum usage of the PUs varies temporally and geographically, each user may have different available channels. Thus the goal is to minimize the *Maximum Time to Rendezvous* ($MTTR$) for both symmetric and asymmetric users, where *symmetric users* means they have the same available channels and *asymmetric* means their available channels are different.

Some previous works use either a central controller or a Common Control Channel (CCC) [15], [19] to simplify the problem. However, it incurs a bottleneck with the increasing number of users and it's vulnerable to adversary attacks. *Blind rendezvous algorithms* are thus proposed with no centralization or the CCC. Most of these algorithms are based on the Channel Hopping (CH) method [11], [20], whereby the user hops among the available channels based on certain predefined sequence. They focus on the rendezvous between

TABLE II
MTTR COMPARISONS WITH STATE-OF-THE-ART RENDEZVOUS ALGORITHMS FOR DIFFERENT NUMBER OF AVAILABLE CHANNELS

| $n_B$ / $n_A$ | | $O(1)$ | | $O(\log N)$ | | $O(N^\epsilon)$ | | $O(N)$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Symmetric | Asymmetric | Symmetric | Asymmetric | Symmetric | Asymmetric | Symmetric | Asymmetric |
| $O(1)$ | JS [18] | $O(N)$ | $O(N^3)$ | - | $O(N^3)$ | - | $O(N^3)$ | - | $O(N^3)$ |
| | DRDS [11] | $O(N)$ | $O(N^2)$ | - | $O(N^2)$ | - | $O(N^2)$ | - | $O(N^2)$ |
| | AHW [5] | $O(N \log N)$ | $O(N \log N)$ | - | $O(N \log_N^2)$ | - | $O(N^{1+\epsilon} \log N)$ | - | $O(N^2 \log N)$ |
| | MLS(this paper) | $O(\log N)$ | $O(\log N)$ | - | $O(\frac{\log_N^3}{\log\log N})$ | - | $O(N^{2\epsilon})$ | - | $O(N^2)$ |
| $O(\log N)$ | JS | - | $O(N^3)$ | $O(N)$ | $O(N^3)$ | - | $O(N^3)$ | - | $O(N^3)$ |
| | DRDS | - | $O(N^2)$ | $O(N)$ | $O(N^2)$ | - | $O(N^2)$ | - | $O(N^2)$ |
| | AHW | - | $O(N \log^2 N)$ | $O(N \log N)$ | $O(N \log_N^2)$ | - | $O(N^{1+\epsilon} \log N)$ | - | $O(N^2 \log N)$ |
| | MLS | - | $O(\frac{\log^3 N}{\log\log N})$ | $O(\frac{\log^2 N}{\log\log N})$ | $O(\frac{\log^3 N}{\log\log N})$ | - | $O(N^{2\epsilon})$ | - | $O(N^2)$ |
| $O(N^\epsilon)$ | JS | - | $O(N^3)$ | - | $O(N^3)$ | $O(N)$ | $O(N^3)$ | - | $O(N^3)$ |
| | DRDS | - | $O(N^2)$ | - | $O(N^2)$ | $O(N)$ | $O(N^2)$ | - | $O(N^2)$ |
| | AHW | - | $O(N^{1+\epsilon} \log N)$ | - | $O(N^{1+\epsilon} \log N)$ | $O(N \log N)$ | $O(N^{1+\epsilon} \log N)$ | - | $O(N^2 \log N)$ |
| | MLS | - | $O(N^{2\epsilon})$ | - | $O(N^{2\epsilon})$ | $O(N^\epsilon)$ | $O(N^{2\epsilon})$ | - | $O(N^2)$ |
| $O(N)$ | JS | - | $O(N^3)$ | - | $O(N^3)$ | - | $O(N^3)$ | $O(N)$ | $O(N^3)$ |
| | DRDS | - | $O(N^2)$ | - | $O(N^2)$ | - | $O(N^2)$ | $O(N)$ | $O(N^2)$ |
| | AHW | - | $O(N^2 \log N)$ | - | $O(N^2 \log N)$ | - | $O(N^2 \log N)$ | $O(N \log N)$ | $O(N^2 \log N)$ |
| | MLS | - | $O(N^2)$ | - | $O(N^2)$ | - | $O(N^2)$ | $O(N)$ | $O(N^2)$ |

Remarks: 1) The comparisons are based on the rendezvous process between two users for both symmetric and asymmetric scenarios; 2) $n_A$ and $n_B$ represent the number of available channels for user $A$ and $B$ respectively; 3) $\epsilon$ is a constant in $(0, 1)$ ; 4)"$-$" means these situations can't be symmetric since $n_A \neq n_B$.

two users, which can be extended to multi-users networks smoothly [18]. The intuitive idea of these methods is to design a sequence based on the $N$ channels, which is called *Global Sequence*(GS) [11]. By assuming the network is time slotted, each user accesses the corresponding channel in each time slot by repeating the same global sequence until rendezvous. If the user hops to some channel in the global sequence which is unavailable, the user just randomly selects a replaced one in its available channel set [11], [18]. Thus, if the user's available channels only account for a small fraction of all the channels, there could be lots of this kind of randomly selected channels (redundant channels) which do not help rendezvous but greatly increase the rendezvous time (Fig. 2 as an example).

Thus, in this paper, we aim to design *Local Sequences* based rendezvous algorithms, which are constructed based on each user's available channels. Compared with the global sequence based methods, the local sequence based methods could avoid these redundant channels that do not help rendezvous but just increase the sequence length and rendezvous time.

Assuming each user has an available channel set $C'$ and a unique identifier (ID) ranging in $[1, M]$ where $M = N^c$ ($c$ is an arbitrary large constant), we propose two algorithms to generate different sequences for different users. Local Sequence (LS) based algorithm is the first introduced algorithm, generating a sequence of fixed length $T = O(N^2)$ for each user. This algorithm guarantees rendezvous for two symmetric users in $O(N)$ time slots, and for two asymmetric users in $O(N^2)$ time slots, which matches the best known results as Table I. Moreover, we propose a Modified Local Sequence (MLS) based algorithm to generate sequences of varying lengths for different users. MLS works significantly better than all extant algorithms when the number of available channels is small (exponentially shorter rendezvous time) and it's also comparable to state-of-the-art GS based rendezvous algorithms when the number is large as Table II. We also carry out extensive simulations to evaluate our proposed algorithms and the results show that our algorithms outperform the extant blind rendezvous algorithms.

The remainder of the paper is organized as follows. The next section gives the related work. Preliminaries are provided in Section III. We present the Local Sequence ($LS$) based algorithm and the Modified Local Sequence ($MLS$) based algorithm in Sections IV and V. Extensive simulations are conducted in Section VI and we conclude the paper in Section VII.

## II. RELATED WORK

Rendezvous algorithm can be divided into two categories: centralized and decentralized algorithms. Centralized algorithms assume a central controller or a dedicated Common Control Channel (CCC) exists [15], [19] and each user can communicate through the the central unit or the CCC. However, this method is vulnerable to adversary attacks and it's inefficient when the number of users is large. Thus decentralized algorithms are proposed without centralization. Some decentralized algorithms establish local CCCs for communication [14], [16], but incur too much overhead in constructing and maintaining them.

Therefore, *blind rendezvous algorithms* lead the research direction where no centralization or CCC exists. Many blind rendezvous algorithms boomed during the past several years and the main technique involved is Channel Hopping (CH). Assuming the network is time-slotted and each user can access an available channel in each time slot. The rendezvous process is considered as hopping among these available channels according to some pre-defined sequence. Most works construct a common sequence for all users based on all the channels' information, which is called *Global Sequence* (GS) in [11]. JS [18], CRSEQ [20], DRDS [11] are several state-of-the-art GS based algorithms. As mentioned, we prefer designing different sequences for different users to avoid the redundant channels in global sequence. To the best of our knowledge, only Alternate Hop-and-Wait (AHW) [5] generates different sequences by assuming each user has a unique identifier (ID), but this method still contains a lot of redundant channels.

Generated Orthogonal Sequence (GOS) [7] is a pioneering work by generating a sequence of length $N(N+1)$ on the basis of a random permutation of $\{1, 2, \cdots, N\}$. However, this algorithm is limited to the situation all channels are available. Quorum-based Channel Hopping [1], [2] works efficiently for only synchronous users (i.e. all users start at the same time), which generate the global sequence based on the quorum system. Asynchronous QCH [3] is modified for asynchronous users (i.e. the users' start time is different), but only applicable to two available channels.

Channel Rendezvous Sequence (CRSEQ) [20] is the first algorithm guaranteeing rendezvous in bounded time. It picks the smallest prime $P > N$ and generates the global sequence with $P$ periods, and each period consists of $3P - 1$ elements based on the triangle number and certain modular operation. However, it works badly when the users are symmetric, i.e. the users have the same available channels as Table I. Jump-Stay (JS) [18] can guarantee efficient rendezvous between symmetric users. The main idea is similar to CRSEQ, which generates the global sequence of $P$ periods and each period contains two *jump* frames and one *stay* frame (each frame contains $P$ numbers). However, JS works badly for the worst scenario of asymmetric users. This result is later improved in [17]. Disjoint Relaxed Difference Set (DRDS) [11] is the first algorithm guaranteeing quick rendezvous for both symmetric and asymmetric users. It reveals the equivalence between DRDS and global sequence. By constructing an appropriate DRDS and transforming it into a global sequence, rendezvous is achieved in $O(N^2)$ time slots for asymmetric users and $O(N)$ time slots for symmetric users.

Alternate Hop-and-Wait (AHW) [5] generates different sequences on the basis that each user has a distinct identifier (ID). Each user's ID can be represented as a unique binary string of length $\log M$ (M is the maximum ID value) and different sequences can be designed. AHW guarantees rendezvous between symmetric users in $O(N \log M)$ time slots and asymmetric users in $O(N^2 \log M)$ time slots. However, AHW still contains redundant channels and our goal in this paper is to design local sequence based algorithms without such redundancy.

## III. Preliminaries

### A. System Model

Consider a CRN with $m$ users (SUs) coexisting with some PUs. Each user is equipped with cognitive radios to sense the licensed spectrum, which is divided into $N$ non-overlapping channels with labels $\{1, 2, \cdots, N\}$. Assume each user has a unique identifier (ID) ranging in $[1, M]$, where $M = N^c$ ($c$ is a constant) is the upper bound of the ID. A channel is *available* to a user if it's not occupied by any nearby PU and each user can only access the sensed available channels for rendezvous. The users with same available channels are called *symmetric*, otherwise they are asymmetric.

Assume time is divided into slots of equal length $2t$, where $t$ is the time for establishing a link between users if they access the same channel. According to IEEE 802.22 [21], $t = 10ms$ and thus each time slot has a duration of $20ms$. Suppose the network is slot-aligned and each user can access an available channel in each time slot. (If two users' time slot is not aligned, an overlap of $t$ time length exists and thus it can be transformed to slot-aligned scenario as Fig. 1) Rendezvous is achieved if the users access the same channel in the same time slot. *Time to rendezvous* ($TTR$) denotes the time cost if all users have begun the process and we use *Maximum* $TTR$ to evaluate the performance of rendezvous algorithms.

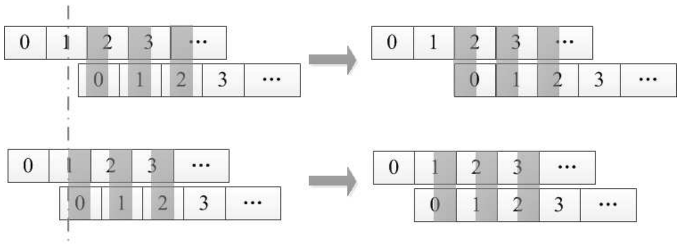



Fig. 1. Transform non-aligned slots to aligned ones

### B. Problem Formulation

In this paper, we focus on handling rendezvous problem for two asynchronous users (i.e. the users' start time may be different) and these algorithms can be extended to multi-users networks smoothly [11], [18].

Consider two users A and B, suppose the available channel sets for them are $C_A, C_B \subseteq C$ , where $C = \{1, 2, \cdots, N\}$ is the set of all channels, and the IDs are $I_A, I_B$, respectively. The rendezvous problem between two users is formulated as:

*Problem 1:* Given a channel set $C' \subseteq C, I \in [1, M]$, design a channel access strategy for each time slot $f_{C',I}(t) \in C'$ such that: $\forall C_A, C_B \subseteq C, C_A \bigcap C_B \neq \emptyset$ and $I_A \neq I_B$, $\forall \delta_t$:

$$\exists T \text{ s.t. } f_{C_A,I_A}(T + \delta_t) = f_{C_B,I_B}(T)$$

The $MTTR$ value of the algorithm $f$ is $MTTR_f = \max_{\forall \delta_t} T$.

In the problem formulation, two users are symmetric if $C_A = C_B$, otherwise they are asymmetric. In the paper, we propose local sequence based algorithms to guarantee rendezvous quickly for both symmetric users and asymmetric users.

For example, $C = \{1, 2, 3\}$, $C_A = \{1, 2\}, I_A = 1$ and $C_B = \{2, 3\}, I_B = 2$, suppose user B is $\delta = 1$ time slot later than user A. Global sequence (GS) based algorithms construct a unique sequence for all users and replace with an available channel randomly if the channel in the sequence is unavailable.

As illustrated in Fig. 2, user A replaces channel 3 by channel 1 or 2 randomly, while user B replaces channel 1 by channel 2 or 3. They can achieve rendezvous on the common channel 2 in time slot 9. As illustrated, GS based algorithms have redundant channels, such as channel 3 in the global sequence is useless for user A, and we tend to handle this problem by constructing different sequences for different users on the basis of available channels.

| Time | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GS | 1 | 1 | 1 | 1 | 3 | 2 | 1 | 3 | 2 | 2 | 2 | 2 | ... |
| User A | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | ... |
| User B | | 2 | 3 | 2 | 3 | 3 | 2 | 2 | 3 | 2 | 2 | 2 | ... |

Fig. 2. An example of global sequence based algorithm (DRDS [11])

*Remark 3.1:* If user A starts later than user B, $\delta_t < 0$ in the description of Problem 1.

## IV. Local Sequence Based Algorithm

### A. Algorithm Description

In this section, we present our Local Sequence (LS) based algorithm to design different sequences for different users. Suppose the user's identifer (ID) is $I \in [1, M]$ ($M = N^c$, $c$ is a constant) and denote the available channel set as $C' \subseteq C$. The intuitive idea is to convert the user's ID to certain fixed base number and different IDs have different representations, thus local sequences could be generated according to the different bits of the new numbers.

To begin with, the user's ID is scaled into $l = \lfloor \log_{P-1} M \rfloor + 1$ bits as in Alg. 1, where $P$ is the smallest prime number $P \geq N$ (Bertrand-Chebyshev Theorem shows that $P < 2N$). From the scaling steps, it's obvious that $\forall i \in [0, l), 1 \leq d(i) < P$, and different IDs have different representations. For example, when $N = 4, M = 16$, $I = 1$ is scaled as $\{1, 1, 2\}$ and $I = 16$ is scaled as $\{2, 1, 1\}$. Another preprocessing is to expand the available channels into $\overrightarrow{e}$ consisting of $P$ numbers. For example, $N = 6$ and $C' = \{2, 4, 5\}$ is expanded as $\overrightarrow{e} = \{2, 2, 2, 4, 5, 5, 5\}$.

Building on the preprocessing, Alg. 1 designs a sequence of length $T = 2(l + 1)P^2$ for the user. It can be thought of constructing $P$ periods of length $L = 2(l + 1)P$. Each period has a base number $x$ as Line 8, for example the $i$-th period has base number $x = i$ and it stays the same for the first $2P$ time slots, which is called *base stage*. The following $2lP$ numbers are generated on the basis of the ID's scaled bits and it's called *hop stage*. This stage consists of $l$ frames of length $2P$ and each frame relates to the scaled bit. For example, the $j$-th frame is generated as $(i + k \cdot d(j)) \mod P, \forall 0 \leq k < 2P$, here $d(j)$ is called *hopping step*. Then the corresponding channel can be accessed as Line 15 based on the expansion of $C'$. In order to guarantee rendezvous for asynchronous users, each frame contains $2P$ numbers and this is from the idea of transforming non-aligned time slots into aligned ones in Fig. 1. Moreover, base stage is designed to accelerate the algorithm.

**Algorithm 1** Local Sequence Based Algorithm

1: Find the smallest prime number $P \geq \max\{N, 3\}$;
2: $l := \lfloor \log_{P-1} M \rfloor + 1$;
3: ID Scale on $I$ to get $\overrightarrow{d} = \{d(0), d(1), \cdots, d(l-1)\}$;
4: Expansion on $C'$ to get $\overrightarrow{e} = \{e(0), e(1), \cdots, e(P-1)\}$;
5: $T := 2(l+1)P^2$, $t := 0$, $L := 2(l+1)P$;
6: **while** Not rendezvous **do**
7: $\quad t' := t \mod T$;
8: $\quad x := \lfloor t'/L \rfloor$, $y := t' \mod L$;
9: $\quad$ **if** $y < 2P$ **then**
10: $\quad\quad z := x$;
11: $\quad$ **else**
12: $\quad\quad y_1 := \lfloor (y - 2P)/(2P) \rfloor$, $y_2 := (y - 2P) \mod 2P$;
13: $\quad\quad z := (x + y_2 \cdot d(y_1)) \mod P$;
14: $\quad$ **end if**
15: $\quad$ Access channel $e(z)$;
16: $\quad t := t + 1$;
17: **end while**

**ID Scale on $I$**

1: **for** $i = l - 1$ to $0$ **do**
2: $\quad d(i) := I \mod (P-1) + 1$;
3: $\quad I := \lfloor I/(P-1) \rfloor$;
4: **end for**

**Expansion on $C'$**

1: Order the channels in $C'$ as $c_1 < c_2 < \cdots < c_{|C'|}$;
2: Construct $\overrightarrow{e} = \{e(0), e(1), \cdots, e(P-1)\}$;
3: $e(j) := c_1$, $\forall 0 \leq j \leq c_2 - 2$;
4: **for** $i = 2$ to $|C'| - 1$ **do**
5: $\quad e(j) := c_i$, $\forall c_i - 1 \leq j \leq c_{i+1} - 2$;
6: **end for**
7: $e(j) := c_{|C'|}$, $\forall c_{|C'|} - 1 \leq j \leq P - 1$;

For example, $N = 3, M = 9, I = 5$, the scaled bits are $\overrightarrow{d} = \{1, 2, 1, 2\}$ and three periods are constructed as Fig. 3. Then the corresponding channels can be accessed on the expansion of the available channel set.

| | | | | | | | hop stage | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | base stage | | | | | | Frame 0 | | | | | | Frame 1 | | | | | | Frame 2 | | | | | | Frame 3 | | | | | |
| Period 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 2 | 1 | 0 | 2 | 1 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 2 | 1 | 0 | 2 | 1 |
| Period 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 0 | 2 | 1 | 0 | 2 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 0 | 2 | 1 | 0 | 2 |
| Period 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 1 | 0 | 2 | 1 | 0 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 1 | 0 | 2 | 1 | 0 |

Fig. 3. An example of Local Sequence based algorithm

### B. Algorithm Performance

*Lemma 4.1:* Every $P$ continuous time slots in the same frame of the hop stage correspond to $P$ different $z$ values (Line 13 of Alg. 1), i.e. these corresponding $z$ values compose a permutation of $\{0, 1, \ldots, P-1\}$.

*Proof:* Consider the $j$-th frame of period $i$, the $2P$ numbers are generated as: $z_k = i + k \cdot d(j) \mod P, \forall 0 \leq k < 2P$. For any $0 \leq k_1, k_2 < 2P$ satisfying $|k_1 - k_2| < P$, $z_{k_1} - z_{k_2} = (k_1 - k_2) \cdot d(j) \neq 0 \mod P$ since $k_1 - k_2 \neq 0 \mod P$ and $0 < d(j) < P$. Thus every $P$ continuous $z$ values generated in the same frame of the hop stage are different from each other and they compose a permutation of $\{0, 1, \ldots, P-1\}$. ∎

Consider two users A and B with $C_A \bigcap C_B \neq \emptyset, I_A \neq I_B$, denote the variables used in Alg. 1 as: $(\overrightarrow{d_A}, \overrightarrow{e_A}, t_A, x(A), y_1(A), y_2(A))$ and $(\overrightarrow{d_B}, \overrightarrow{e_B}, t_B, x(B), y_1(B), y_2(B))$, respectively.

*Theorem 1:* Alg. 1 guarantees rendezvous in $MTTR = 2(l+1)P = O(N)$ time slots for two symmetric users.

*Proof:* Two symmetric users (A and B) means $C_A = C_B$, thus $\overrightarrow{e_A} = \overrightarrow{e_B}$ can be verified easily. Since $I_A \neq I_B$, there exists $0 \leq i < l$ such that $d_A(i) \neq d_B(i)$. Without loss of generality, suppose user B is $\delta \geq 0$ time slot later than user A. Define $\delta_T = \delta \mod T$ and $\delta_L = \delta \mod L$. According to different $\delta$ values, we prove the theorem from six cases.

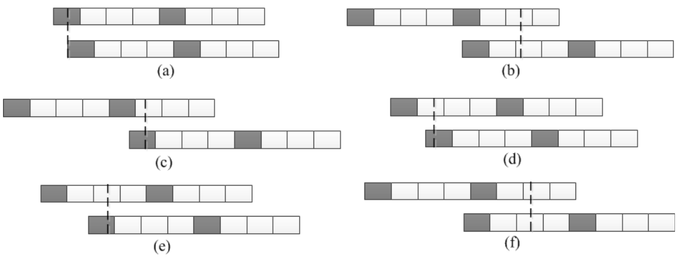


Fig. 4. Illustrations of Theorem 1's proof

**Case** 1: $0 \leq \delta_L < 2P$ and $0 \leq \delta_T < 2P$. User B can achieve rendezvous with user A in the first time slot as Fig. 4(a), because user A is accessing channel $e_A(0)$ in the base stage of Period 0 and user B's first attempt is $e_B(0) = e_A(0)$.

**Case** 2: $0 \leq \delta_L < P$ and $\delta_T \geq 2P$. Different from case 1, this situation means although user A is in base stage when user B starts, user A is accessing $e_A(k) \neq e_B(0), k > 0$, thus they don't rendezvous during this stage. Since there exists $0 \leq i < l$ such that $d_A(i) \neq d_B(i)$, they can achieve rendezvous in the $i$-th frame of the hop stage as Fig. 4(b). When $t_B \in [2(i+1)P, (2i+3)P)$ for user B, from Line 8 and Line 12 of Alg. 1: $x(B) = 0$, $y_1(B) = i$ and $0 \leq y_2(B) < P$. The corresponding $z$ values are generated as Line 13:

$$z_t(B) = [0 + y_2(B) \cdot d_B(i)] \mod P$$

For user A, $t_A = t_B + \delta$, from Line 8 and Line 12 of Alg. 1, $x(A) = \lfloor (t_B + \delta)/L \rfloor = \lfloor \delta_T / L \rfloor$, $y_1(A) = i$ and $0 \leq y_2(A) = y_2(B) + \delta_L < 2P$. Thus, the corresponding $z$ values are generated as:

$$z_t(A) = [x(A) + y_2(A) \cdot d_A(i)] \mod P$$

Let $z_t(A) = z_t(B)$ i.e. they access the same channel, we can derive:

$$[d_B(i) - d_A(i)] \cdot y_2(B) = x(A) + \delta_L \cdot d_A(i) \mod P \quad (1)$$

As $d_B(i) \neq d_A(i)$, such $y_2(B)$ exists and rendezvous is guaranteed in $t_B \leq (2i+3)P \leq (2l+1)P$ time slots.

**Case** 3: $P \leq \delta_L < 2P$ and $\delta_T \geq 2P$. Similar as case 2, user B cannot achieve rendezvous with user A both in hop stage. However, it's obvious that when $t_B \in [2P - \delta_L, 3P - \delta_L)$, user B is in base stage accessing channel $e_B(0)$, while user A is in the 0-th frame of the hop stage (in some period). From Lemma 4.1, the $P$ continuous $z_A$ numbers compose a permutation of $\{0, 1, \ldots, P-1\}$. Thus rendezvous can be guaranteed in $t_B \leq 2P$ time slots when $z_A = 0$ and user A accesses channel $e_A(0) = e_B(0)$ as Fig. 4(c).

**Case** 4: There exists $i' \in [0, l)$ such that $(2i'+2)P \leq \delta_L < (2i'+3)P$. As illustrated in Fig. 4(d), user B accesses channel $e_B(0)$ for the first $P$ time slots, while user A is in the same frame of the hop stage. Thus from the analysis of case 3, they can achieve rendezvous in $t_B \leq P$ time slots.

**Case** 5: There exists $i' \in [0, l-1)$ such that $(2i'+3)P \leq \delta_L < (2i'+4)P$. Different from case 4, when $t_B \in [0, P)$, user A isn't in the same frame, but rendezvous can be achieved when $t_B \in [P, 2P)$ as Fig. 4(e) (the corresponding $P$ time slots $t_A$ are in the same frame).

**Case** 6: $2lP \leq \delta_L < (1+2l)P$. The situation is different from case 5 because when $t_B \in [P, 2P)$, user A is in base stage and rendezvous may not happen. It's akin to case 2 that they can achieve rendezvous in the $i$-th frame where $d_A(i) \neq d_B(i)$ in $t_B \leq 2(l+1)P$ time slots as Fig. 4(f).

Combining these situations, rendezvous for symmetric users can be achieved in $2(l+1)P = O(N)$ time slots. ∎

*Theorem 2:* Alg. 1 guarantees rendezvous in $MTTR = 2(l+1)P^2 = O(N^2)$ time slots for two asymmetric users.

*Proof:* Two users A and B are asymmetric, and thus $C_A \neq C_B, I_A \neq I_B$. After the ID Scale and Expansion, the representations are different, i.e. $\overrightarrow{d_A} \neq \overrightarrow{d_B}$ and $\overrightarrow{e_A} \neq \overrightarrow{e_B}$. So there exists $0 \leq i < l$ such that $d_A(i) \neq d_B(i)$. As they share at least one common channel, there exists $0 \leq j < P$ such that $e_A(j) = e_B(j)$. The theorem can be proved based on the six cases in symmetric scenario.

Case 1: $0 \leq \delta_L < 2P$ and $0 \leq \delta_T < 2P$. Different from symmetric users, $e_A(0)$ may not equal to $e_B(0)$, thus rendezvous is not guaranteed in the base stage of the 0-th period. However, when time counts to the $j$-th period, it's clear that when $t_B \in [2(l+1)P \cdot j, 2(l+1)P \cdot +P)$, user A and B are both in base stage, accessing $e_A(j) = e_B(j)$. Thus $MTTR = t_B \leq 2(l+1)P^2$ time slots.

Case 2: $0 \leq \delta_L < P$ and $\delta_T \geq 2P$. When $t_B \in [k \cdot L + 2(i+1)P, k \cdot L + (2i+3)P)$, user B is in the $i$-th frame of the $k$-th period ($0 \leq k < P$), thus the corresponding $z_t(B)$ can be generated from Line 13:

$$z_t(B) = [k + y_2(B) \cdot d_B(i)] \mod P \quad (2)$$

From $t_A = t_B + \delta$, we can derive the $z_t(A)$ values as:

$$z_t(A) = [x(A) + k + y_2(A) \cdot d_A(i)] \mod P \quad (3)$$

where $x(A) = \lfloor \delta_T / L \rfloor$ is similar with case 2 of Theorem 1. Let $z_t(A) = z_t(B)$ to conclude the same result as Eqn. (1). Denote $\theta = d_B(i) - d_A(i)$, $\lambda = x(A) + \delta_L \cdot d_A(i)$, it can be figured out $y_2(B) = \lambda \cdot \theta^{-1}$, where $\theta^{-1} \cdot \theta = 1 \mod P$ exists. Plugging $y_2(B)$ into Eqn. (2), the corresponding $z_t(B)$ is computed. As $k$ ranges in $[0, P)$, it's obvious that there exists $0 \le k^* < P$ such that $k^* + y_2(B) \cdot d_B(i) = j \mod P$, which implies users A and B both access channel $e_B(j) = e_A(j)$ at the same time. Thus rendezvous is guaranteed in $t_B \le 2(l+1)P^2$ time slots.

For the other four cases discussed in Theorem 1, they can be proved in the similar way as case 1 or case 2. Thus we conclude that rendezvous for two asymmetric users is bounded by $2(l+1)P^2 = O(N^2)$ time slots. ∎

From Theorem 1 and Theorem 2, the LS based algorithm matches the best known results of Global Sequence based algorithms shown in Table I.

## V. Modified Local Sequence Based Algorithm

Although the LS based algorithm guarantees rendezvous between two users in short time, it seems inefficient as the length of each frame in Alg. 1 is fixed to be $2P$. (The users' IDs are converted to fixed base numbers and the length of each frame is related to the base.) When the number of available channels $n$ is small, we could convert the specific user's ID to a new base number where the base and the length of each frame relate to $n$ directly. The challenge is that different IDs may have same representations in different base systems, such as $(12)_6 = 8$ but $(12)_4 = 6$. Thus refined improvement should be made and the Modified Local Sequence (MLS) based algorithm is described in Alg. 2.

### A. Algorithm Description

Different from Alg. 1, Alg. 2 counts the number of available channels as $n = |C'|$ and finds the smallest prime number $p \ge n$. The preprocessing of ID Scale is similar to Alg. 1, the difference is that the ID is scaled by $p-1$ where $p$ relates to the number of available channels. Different users may have different $p$ values, thus the number of scaled bits for different users may be different since $l := \lfloor \log_{p-1} M \rfloor + 1$. For example, $N = 5, M = 25$, the scaled bits for the user with $n = 3, I = 5$ is $\overrightarrow{d} = \{1,1,2,1,2\}$, but for the user with $n = 4, I = 5$ is $\overrightarrow{d} = \{1,2,2\}$.

Another preprocessing is extraction on $C'$, which is different from expansion procedure in Alg. 1. Extraction procedure constructs $\overrightarrow{e}$ with $p$ numbers by ordering the available channels as $c_1 < c_2 < \cdots < c_n$. For example, $N = 7$ and channels $\{1,2,4,7\}$, and the extraction result is $\overrightarrow{e} = \{1,2,4,7,1\}$. The number of $\overrightarrow{e}$ is related to the number of available channels $n$, not all channels $N$.

Building on the preprocessing, Alg. 2 constructs a sequence of length $T = 2(l+1)p^2$, which also can be thought of constructing $p$ periods of length $L = 2(l+1)p$. There are also two stages in each period like LS based algorithm, *base stage* consists of $2p$ base values $x = i$ for the $i$-th period as Line 9, and *hop stage* contains $p$ frames. The $2p$ numbers of the $j$-th frame are generated as $z = (i + k \cdot d(j)) \mod p$, $\forall 0 \le k < 2p$. Then the corresponding channel $e(z)$ is accessed as Line 16. Alg. 2 is a modified version of Alg. 1, but it could be more efficient as the length of each user's sequence may be different. When the user has less available channels, the corresponding sequence is shorter.

**Algorithm 2** Modified Local Sequence Based Algorithm

1: Count the number of available channels $n = |C'|$;
2: Find the smallest prime number $p \ge \max\{n, 3\}$;
3: $l := \lfloor \log_{p-1} M \rfloor + 1$;
4: ID Scale on $I$ to get $\overrightarrow{d} = \{d(0), d(1), \cdots, d(l-1)\}$;
5: Extraction on $C'$ to get $\overrightarrow{e} = \{e(0), e(1), \cdots, e(p-1)\}$;
6: $T := 2(l+1)p^2$, $t := 0$, $L := 2(l+1)p$;
7: **while** Not rendezvous **do**
8:   $t' := t \bmod T$;
9:   $x := \lfloor t'/L \rfloor$, $y := t' \mod L$;
10:   **if** $y < 2p$ **then**
11:     $z := x$;
12:   **else**
13:     $y_1 := \lfloor (y-2p)/(2p) \rfloor$, $y_2 := (y-2p) \mod 2p$;
14:     $z := (x + y_2 \cdot d(y_1)) \mod p$;
15:   **end if**
16:   Access channel $e(z)$;
17:   $t := t + 1$;
18: **end while**

**ID Scale on $I$**

1: **for** $i = l-1$ to $0$ **do**
2:   $d(i) := I \bmod (p-1) + 1$;
3:   $I := \lfloor I/(p-1) \rfloor$;
4: **end for**

**Extraction on $C'$**

1: Order the channels in $C'$ as $c_1 < c_2 < \cdots < c_n$;
2: Construct $\overrightarrow{e} = \{e(0), e(1), \cdots, e(p-1)\}$;
3: **for** $j = 0$ to $p-1$ **do**
4:   $i := j \mod n + 1$;
5:   $e(i) := c_i$;
6: **end for**

### B. Correctness and Efficiency

Consider two users A and B with $C_A \bigcap C_B \neq \emptyset$ and $I_A \neq I_B$. Denote the number of available channels for two users as $n_A = |C_A|, n_B = |C_B|$ in the first line of Alg. 2. Similarly, denote the other variables during Alg. 2 as $(p_A, l_A, \overrightarrow{d_A}, \overrightarrow{e_A}, T_A, L_A, t_A, x(A), y_1(A), y_2(A), z_t(A))$ and $(p_B, l_B, \overrightarrow{d_B}, \overrightarrow{e_B}, T_B, L_B, t_B, x(B), y_1(B), y_2(B), z_t(B))$, respectively. Similar with Lemma 4.1, every $p_A$ continuous time slots for user A in the same frame of the hop stage generate $p_A$ different $z_A$ values in $[0, p_A)$ (the same situation for user B).

*Theorem 3:* Alg. 2 guarantees rendezvous in $MTTR = 2(l_A+1)p_A = O(l_A n_A)$ time slots for two symmetric users.

*Proof:* Two symmetric users ($C_A = C_B$) implies $n_A = n_B, p_A = p_B, l_A = l_B, \overrightarrow{e_A} = \overrightarrow{e_B}$. From the scaling on ID, there exists $0 \le i < l_A$, such that $d_A(i) \neq d_B(i)$. The length of two sequences are the same ($T_A = T_B$) and from the proof details of Theorem 1, it can be concluded similarly. ∎

When the number of available channels is small, MLS algorithm performs much better than LS algorithm. It's clear that $l_A = O(\log N/\log n_A)$ and thus the $MTTR$ value could be small. Such as $n_A = O(1), MTTR = O(\log N)$; $n_A = O(\log N), MTTR = O(\log^2 N/\log\log N)$; $n_A = O(N^\epsilon)(0 < \epsilon < 1), MTTR = O(N^\epsilon)$. When it comes to asymmetric users, the situation is much more complicated.

*Lemma 5.1:* For two asymmetric users ($C_A \neq C_B$), rendezvous is guaranteed in $MTTR = 2(l_B+1)p_B^2 = O(l_B n_B^2)$ time slots if $p_A = p_B$.

*Proof:* Two asymmetric users $C_A \neq C_B$ implies $\overrightarrow{e_A} \neq \overrightarrow{e_B}$. Since $p_A = p_B$, $I_A \neq I_B$, the number of scaled bits $l_A = l_B$ and there exists $0 \leq i < l_A$ such that $d_A(i) \neq d_B(i)$. From $C_A \bigcap C_B \neq \emptyset$, there exist $0 \leq j_1, j_2 < p_A$ suit $e_A(j_1) = e_B(j_2)$. The situation is similar with Theorem 2. For cases $1, 3, 4, 5$ in the proof of Theorem 1, user B can achieve rendezvous in base stage by accessing channel $e_B(j_2)$ in $t_B \in [2(1+l_B)p_B \cdot j_2, 2(1+l_B)p_B \cdot j_2 + 2p_B)$ . For the other two cases, rendezvous happens in the users' hop stage. The difference is in Eqn. (2) and Eqn. (3), let $z_t(A) = j_1$ and $z_t(B) = j_2$, it can be verified similarly that such $t_B < 2(l_B+1)p_B^2$ exists. ■

Without loss of generality, suppose $p_B > p_A$ and the following lemmas are concluded.

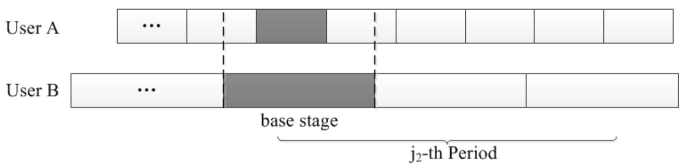



Fig. 5. Example of Lemma 5.2 when $p_B \geq 2p_A$

*Lemma 5.2:* Rendezvous is guaranteed in $MTTR = 2(l_B+1)p_B^2 = O(l_B n_B^2)$ time slots if $p_B \geq 2p_A$.

*Proof:* Since $C_A \bigcap C_B \neq \emptyset$, there exists $0 \leq j_1 < p_A, 0 \leq j_2 < p_B$ such that $e_A(j_1) = e_B(j_2)$. No matter who starts the algorithm firstly, user B can achieve rendezvous in the base stage of period $j_2$. This is because the base stage contains $2p_B > 4p_A$ numbers, which is large enough to cover $p_A$ continuous numbers from a same frame of user A's hop stage as illustrated in Fig. 5. Thus such $j_1 \in [0, p_A)$ exists and the $MTTR$ value is bounded by $2(1_B+1)p_B^2$ time slots. ■

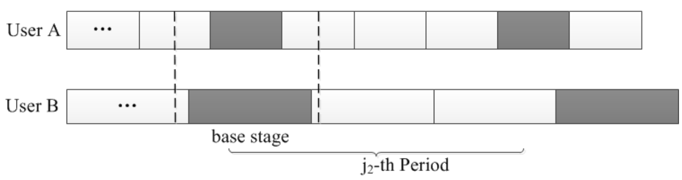



Fig. 6. Example of Lemma 5.3 when $p_A < p_B < 2p_A$

*Lemma 5.3:* Rendezvous is guaranteed in $MTTR = 2(l_B+1)p_B^2 p_A = O(l_B n_B^2 n_A)$ time slots if $p_A < p_B < 2p_A$.

*Proof:* Different from Lemma 5.2, $2p_B$ time slots are not large enough to assure any $p_A$ continuous numbers for the same frame of the hop stage exist. Thus we analyze the worst situation for two users. Suppose $0 \leq j_1 < p_A, 0 \leq j_2 < p_B$ exist such that $e_A(j_1) = e_B(j_2)$. Consider the base stage of the $j_2$-th period of user B (i.e. $t_B \in [\delta_B, \delta_B + 2p_B)$, where $\delta_B = 2(l_B+1)p_B \cdot j_2$). Denote the corresponding time for user A as $\delta_A$ and as illustrated in Fig. 6, the only situation that user B cannot rendezvous in the base stage is: $L_A - p_A < (\delta_A \mod L_A) < L_A$ and $0 < (\delta_A + 2p_B \mod L_A) < p_A$. Only when the two conditions are satisfied, user B may not achieve rendezvous in the base stage. Then user B repeats the sequence and we can figure out how many times needed to rendezvous. Denote $\epsilon = T_B \mod L_A$ and it's clear that $\epsilon \neq 0$. Only when $\epsilon \in (0, p_A)$ or $\epsilon \in (L_A - p_A, L_A)$, they may not rendezvous as user B repeats the sequence for the second time. However, if $\epsilon \in (0, p_A)$, after at most $\frac{p_A}{\epsilon}$ times, $(\delta_A + \frac{p_A}{\epsilon} \cdot T_B) \mod L_A \in [0, P)$ and rendezvous happens. If $\epsilon \in (L_A - p_A, L_A)$, rendezvous is also guaranteed after $\frac{p_A}{L_A - \epsilon}$ times. Thus $MTTR = 2(l_B+1)p_B^2 p_A$ time slots. ■

Lemma 5.3 reveals an extreme situation for the $MTTR$ values and it rarely happens. Thus we show the $MTTR$ values on the basis of $n_A, n_B$ for most cases in Table II. Concluding from Lemmas 5.1-5.3:

*Theorem 4:* Alg. 2 guarantees rendezvous in $MTTR = O(l_B n_B^2)$ time slots if $p_B \geq 2p_A$ or $p_B = p_A$ and in $MTTR = O(l_B n_B^2 n_A)$ time slots if $p_A < p_B < 2p_A$.

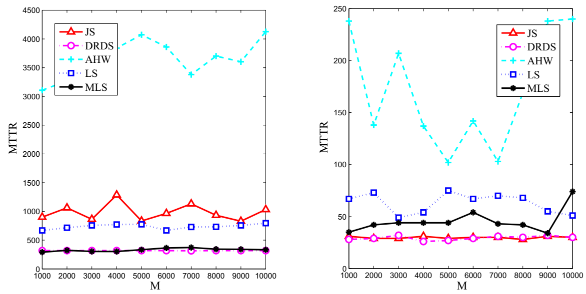
Combing Theorem 3 and Theorem 4, the MLS based algorithm is significantly better than the best known results in Table I when the number of available channels is small. Specifically, the MLS based algorithm can guarantee rendezvous in $O(l_A n_A)$ time slots for symmetric users, which is much smaller than $O(N)$ when $n_A = o(N)$. It also guarantees rendezvous for asymmetric users in less time than $O(N^2)$ time slots for most combinations in Table II.

## VI. Simulation

We evaluate the performance of our proposed algorithms under different circumstances and compare the results with several state-of-the-art algorithms. We choose Jump-Stay (JS) [18], DRDS [11] and AHW [5] for the $MTTR$ comparisons with our LS and MLS based algorithms.

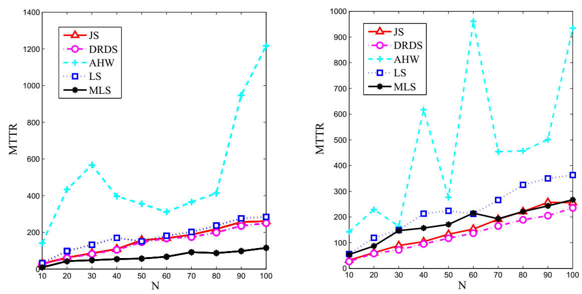
Define $\theta_A = \frac{n_A}{N}$, $\theta_B = \frac{n_B}{N}$, $n_G = |C_A \bigcap C_B|$ and $\theta_G = \frac{n_G}{N}$. In each simulation, the starting time of each user is random and the identifers (IDs) for the users are randomly generated in $[1, M]$. Based on different circumstances, the available channels are also generated randomly. Detailed parameters are described for the corresponding figures and the results provided are the means of 5000 separate time.

Since AHW and our proposed algorithms involve the users' IDs, we firstly evaluate the impact of the ID's maximum value $M$. Fix $N = 10, n_A = 5, n_B = 5$, when two users are asymmetric and $n_G = 1$, Fig. 7(a) shows our proposed algorithms don't increase too much as $M$ increases, moreover, the MLS based algorithm is as good as DRDS algorithm. When they are symmetric, i.e. $C_A = C_B$, Fig. 7(b) reveals that both LS and MLS based algorithms are stable and their performance is comparable to DRDS and JS. Although AHW



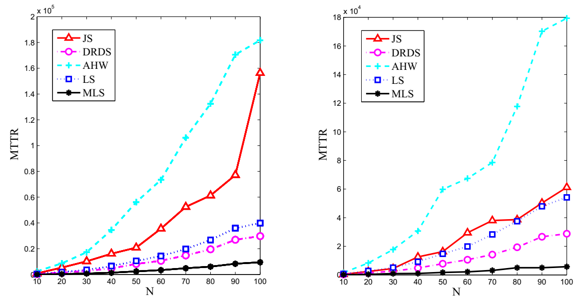

(a) $MTTR$ vs. $M$ (b) $MTTR$ vs. $M$

Fig. 7. $MTTR$ comparison as $M$ increases from 1000 to 10000 when $N = 10, n_A = 5, n_B = 5$: (a) $n_G = 1$; (b) $C_A = C_B$.



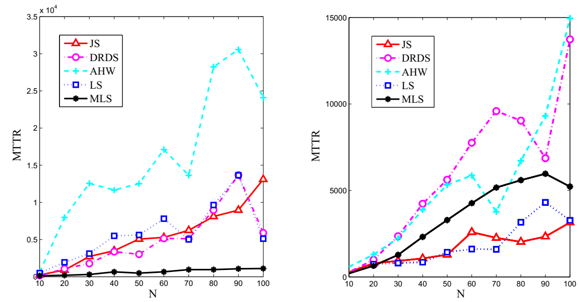

(a) $MTTR$ vs. $N$ (b) $MTTR$ vs. $N$

Fig. 8. $MTTR$ comparison for symmetric users when $N$ increases from 10 to 100. $M = 100$, (a) $\theta_A = \theta_B = 0.2$; (b) $\theta_A = \theta_B = 0.8$



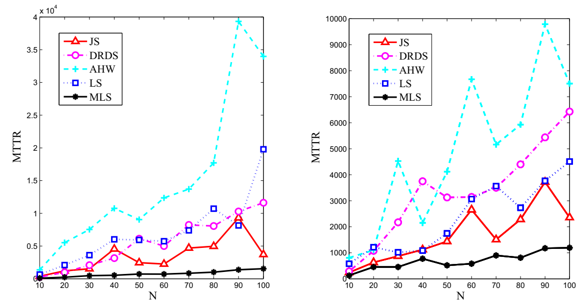

(a) $MTTR$ vs. $N$ (b) $MTTR$ vs. $N$

Fig. 9. $MTTR$ comparison when $N$ increases from 10 to 100. $M = 100$, (a) $\theta_A = \theta_B = 0.5, n_G = 1$; (b) $\theta_A = \theta_B = 0.3, n_G = 1$



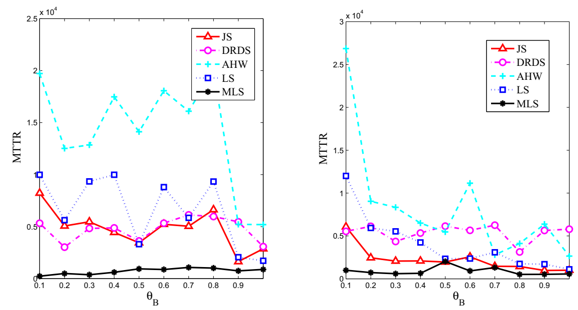

(a) $MTTR$ vs. $N$ (b) $MTTR$ vs. $N$

Fig. 10. $MTTR$ comparison for asymmetric users 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. $N$ (b) $MTTR$ vs. $N$

Fig. 11. $MTTR$ comparison for asymmetric users. $N \in [10, 100]$, $M = 100$, (a) $\theta_A = 0.5, \theta_B = 0.2, \theta_G = 0.1$; (b) $\theta_A = 0.5, \theta_B = 0.8$




(a) $MTTR$ vs. $\theta_B$ (b) $MTTR$ vs. $\theta_B$

Fig. 12. $MTTR$ comparison for asymmetric users when $\theta_B$ increases from 0.1 to 1. $M = 100, N = 50$, (a) $\theta_A = 0.2$; (b) $\theta_A = 0.5$

algorithm also uses the users' ID, it's affected by the increasing of $M$ and it's unstable. In the following scenarios, we set $M = 100$.

We evaluate these algorithms for two symmetric users. As shown in Fig. 8(a), when $\theta_A = \theta_B = 0.2$, the MLS based algorithm outperforms the others and LS based algorithm works well as JS and DRDS. When the number of available channels is larger, such as $\theta_A = \theta_B = 0.8$, when $N$ increases from 10 to 100, DRDS algorithm works best and the LS and MLS based algorithms are better than AHW as Fig. 8(b). It's because the MLS based algorithm suits the users with small number of available channels, as described in Table II.

In order the evaluate the $MTTR$ values for some extreme situations, set $\theta_A = \theta_B = 0.5$ (and 0.3) with only one common channel, i.e. $n_G = 1$. When $N$ increases from 10 to 100, Fig. 9 shows that the MLS based algorithm works best for the two extreme situations. JS and AHW algorithms work badly since JS algorithm can only guarantee rendezvous in $O(N^3)$ time slots for the worst case, while AHW algorithm is influenced by both $N$ and $M$ values.

For the comparison of two asymmetric users, we set $\theta_A = \theta_B = 0.2$ (and 0.8) in Fig. 10. In Fig. 10(a), $\theta_G = 0.1$ and it shows the MLS based algorithm has much smaller $MTTR$ values than others, while the LS based algorithm is as good as JS and DRDS algorithms. Fig. 10(b) reveals that the LS based algorithm performs better when the number of available channels is large and it corroborates our theoretical analyses.

We also evaluate the these algorithms' performance when the number of available channels for two users are different. In Fig. 11(a), set $\theta_A = 0.5, \theta_B = 0.2$ and $\theta_G = 0.1$, the MLS based algorithm is much better than other algorithms. In Fig. 11(b), when $\theta_A = 0.5, \theta_B = 0.8$, the MLS based algorithm also outperforms others. We also find that the LS based algorithm is comparable to both JS and DRDS algorithms.

In Fig. 12, we fix $N = 50$ and evaluate the $MTTR$ values when $\theta_B$ increases from 0.1 to 1. When $\theta_A = 0.2$, i.e. the number of available channels is small enough, the MLS based algorithm improves the previous best results significantly as Fig. 11(a). When $\theta_A = 0.5$, the MLS based algorithm also works best and the LS based algorithm has smaller $MTTR$ values than AHW algorithm.

Concluding from the extensive simulation results, our proposed LS and MLS based algorithms are less affected by the increasing $M$ values. For both symmetric and asymmetric comparisons, the LS and MLS based algorithm is comparable to the state-of-the-art rendezvous algorithms (JS and DRDS). Moreover, when the number of available channels is small, the MLS based algorithm works significantly better than others, which corroborates our theoretical analyses.

## VII. Conclusion

In this paper, we study the rendezvous problem in Cognitive Radio Networks from a new aspect. Most extant works design Global Sequences (GS) on the basis of $N$ channels and the best results guarantee rendezvous for two symmetric users in $O(N)$ time slots and two asymmetric users in $O(N^2)$ time slots. In this paper, we propose two algorithms based on the intuitive idea that different users have different local sequences building on the available channels and distinct identifiers (IDs). The first one is Local Sequence (LS) based algorithm which scales the user's ID and constructs sequences on the expansion of the available channels. The other is Modified Local Sequence (MLS) based algorithm which generates shorter sequences for the users with less available channels. The LS based algorithm guarantees rendezvous in $O(N)$ and $O(N^2)$ time slots for two symmetric and asymmetric users respectively, which matches the state-of-the-art GS based results. Our main contribution is the MLS based algorithm that guarantees rendezvous in $O(ln)$ time slots for two symmetric users, where $n$ is the number of available channels and $l = O(\log N / \log n)$. Moreover, it also guarantees rendezvous in shorter time as Table II for two asymmetric users. Through extensive simulations, these results also show that the LS based algorithm is comparable to the state-of-the-art algorithms and the MLS based algorithm is significantly better (exponentially shorter rendezvous time) when the number of available channels is small.

## VIII. Acknowledgement

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.

## References

[1] K. Bian, J.-M. Park, and R. Chen. A Quorum-Based Framework for Establishing Control Channels in Dynamic Spectrum Access Networks. In *Mobicom*, 2009.

[2] K. Bian, J.-M. Park, R. Chen. Control Channel Establishment in Cognitive Radio Networks using Channel Hopping. *IEEE Journal on Selected Areas in Communications*, 29(4):689-703, 2011.

[3] 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.

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

[5] 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.

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

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

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

[9] 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.

[10] A.G. Fragkiadakis, E.Z. Tragos, and I.G. Askoxylakis. A survey on Security Threats and Detection Techniques in Cognitive Radio Networks. *Communications Survey & Tutorials*, 15(1): 428-455, 2013.

[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] S. Ji, R. Beyah, and Z. Cai. Minimum-Latency Broadcast Scheduling for Cognitive Radio Networks. In *SECON*, 2013.

[14] J. Jia, Q. Zhang, and X. Shen. HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management. *IEEE Journal on Selected Areas in Communications*, 26(1):106-117, 2008.

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

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

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

[18] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung. Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(10):1867-1881, 2012.

[19] J. Perez-Romero, O. Salient, R. Agusti, and L. Giupponi. A Novel On-Demand Cognitive Pilot Channel enabling Dynamic Spectrum Allocation. In *DySPAN*, 2007.

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

[21] 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.

[22] 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.

[23] D. Yang, J. Shin, and C. Kim. Deterministic Rendezvous Scheme in Multichannel Access Networks. *Electronics Letters*, 46(20):1402-1404, 2010.

[24] 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.

[25] Q. Zhao, and B. M. Sadler. A Survey of Dynamic Spectrum Acess (signalprocessing, networking and regulatory policy). *IEEE Signal Processing Magazine*, 2007.