A Tutorial on Multichannel Rendezvous in Cognitive Radio Networks

Cheng-Shang Chang, Duan-Shin Lee
Institute of Communications Engineering,
National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
E-mail address: cschang@ee.nthu.edu.tw and lds@cs.nthu.edu.tw.
and
Wanjiun Liao
Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.
E-mail address: wjliao@ntu.edu.tw.

Abstract

Rendezvous search that asks two persons to find each other among a set of possible locations has regained tremendous research interest lately in the research community of cognitive radio networks (CRNs). In a CRN, there are two types of users: primary spectrum users (PUs) and secondary spectrum users (SUs). SUs are only allowed to share spectrum with PUs provided that they do not cause any severe interference to the PUs. To do this, SUs first sense a number of frequency channels. If a channel is not blocked by a PU, then that channel may be used for SUs to establish a communication link. One of the fundamental problems in a CRN is then for two SUs to find a common *unblocked* channel.

To find a common unblocked channel, channel hopping (CH) schemes are commonly used in the literature. In a CH scheme, time is divided into consecutive time slots and each SU hops to a channel in every time slot according to a specific CH sequence. Eventually, two SUs rendezvous when they both hop to a common unblocked channel. Such a rendezvous search problem in a CRN is known as the *multichannel rendezvous problem*.

The objective of this book chapter is to provide a tutorial on the multichannel rendezvous problem under various categories and assumptions, including asymmetric/symmetric roles, synchronous/asynchronous clocks, homogeneous/heterogeneous available channel sets, and oblivious/non-oblivious rendezvous. Instead of giving rigorous mathematical proofs of the results in the multichannel rendezvous problem, we will provide the needed insights/intuitions to understand these results. Though there are many mathematical theories associated with the multichannel rendezvous problem, including Galois fields, finite projective planes, orthogonal Latin squares, quorum systems, and difference sets, in our view the fundamental theorem for the multichannel rendezvous problem is the *Chinese Remainder Theorem* and the development of this tutorial will be focused on the Chinese Remainder Theorem.

I. Introduction

Rendezvous search that asks two persons to find each other among a set of possible locations is perhaps one of the most common problems in our daily life. Such a problem has been studied extensively in the literature (see e.g., the book [8] and references therein). Anderson and Weber [11] first studied the rendezvous problem on discrete locations. The rendezvous search problem was generalized to the compact metric space in [2]. Since then, the rendezvous search problem on various topologies has been studied extensively in the literature, e.g., the rendezvous search problem on the line [7], [9], [31], [40], the rendezvous search problem on the interval [51], the rendezvous search problem on a graph [3], [6], [35], [61], the two-dimensional rendezvous problem [4], [10], [59], [60], and rendezvous in higher dimensions [5].

The rendezvous search problem has regained tremendous research interest lately in the research community of cognitive radio networks (CRNs). As wireless networks used today are regulated by a fixed spectrum policy, such a policy leads to the problem of inefficient usage of radio spectrum [39]. To solve this problem, cognitive radio (CR) [62] was introduced to improve the spectrum efficiency. In a CRN, unlicensed users (called secondary users (SUs)) are allowed to use unused licensed spectrum without interfering with licensed users (called primary users (PUs)). With the support of software defined radio (SDR) technology, nodes equipped with cognitive radio transceivers (CR transceivers) can intelligently adjust the transmission characteristics (e.g., transmission power, carrier frequency, and modulation strategy) to achieve highly reliable communications and high spectrum efficiency throughout a wide range of spectrum. Therefore, they can quickly switch their operation spectrums and utilize the unused licensed spectrums efficiently.

In a CRN, each SU is associated with a set of channels for communications, and the availability of each channel is determined by the behavior of neighboring PUs. SUs located in different locations may have different available channel sets because their neighboring PUs may be different. In addition, the available channel set of an SU may change with time because the neighboring PUs may change their transmission states. The diverseness of available channel sets makes the problem of establishing a control channel very challenging in a CRN, especially in a fully distributed environment. For instance, as shown in Fig. 1, there are five channels in a CRN $\{0, 1, 2, 3, 4\}$. The PU on the left uses the two channels $\{1, 2\}$ and the PU on the right uses the three channels $\{0, 2, 4\}$. Thus, the available channel set of SU A is $\{0, 3, 4\}$ and the available channel set of SU B is $\{1, 3\}$. The common available channel for these two SUs is channel 3.

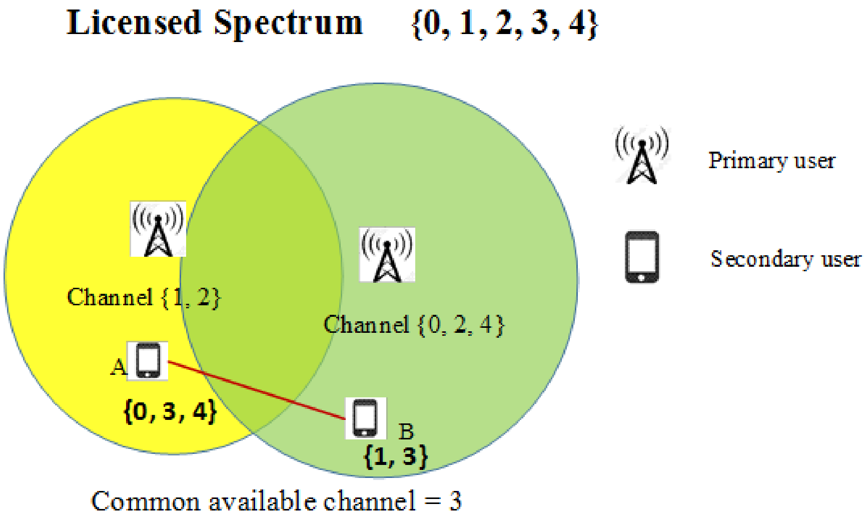

Fig. 1. An illustration of a cognitive radio network.

The most typical approach for control channel establishment is to use a dedicated global control channel among all SUs [30], [55], [77], [79]. However, the availability of channel sets among SUs may vary due to the fact that they might have different neighboring PUs. Hence, the likelihood of having a control channel globally available to all SUs is very slim. Even if SUs are able to find a globally available channel, the availability of this dedicated control channel may change over time. When the dedicated control channel is unavailable, the normal operations of SUs may be disrupted. In particular, new data packets cannot be transmitted because the control messages cannot be exchanged even though there are other common available channels. Once a PU starts using its channel, it is very likely that the PU will continue to use this channel for a long time. Thus, all the control messages will be "blocked" during this long duration. Such a problem is known as the *PU long-time blocking problem* (see e.g., [25], [67]). Moreover, using one single control channel may introduce a bottleneck in the operation and may further cause the *control channel saturation problem* in a high node-density environment (see e.g., [67] and references therein).

To cope with the control channel saturation problem and the PU long-time blocking problem, channel hopping (CH) schemes are commonly used in the literature (see e.g., [13]–[15], [27], [34], [50], [53], [57], [63], [67]–[69], [74], [78]). In a CH scheme, time is usually divided into consecutive time slots and each SU hops to a channel in every time slot according to a specific CH sequence. Eventually, two SUs rendezvous when they both hop to a common unblocked channel. Such a rendezvous search problem in a CRN is known as the *multichannel rendezvous problem*.

The multichannel rendezvous problem is mathematically equivalent to the rendezvous problem on a *labelled* complete graph (Chapter 13 of [8]) as the channels in the multichannel rendezvous problem are generally assumed to be *labelled* and known to both users. Since the channels are labelled, both users could simply use the FOCAL strategy [8] and hop to channel 0 when the rendezvous process is started. However, this is not a desirable thing to do in a CRN as all the SUs will hop to the same channel and that cause severe interference among themselves. Such a problem is the same as the control channel saturation problem. As such, it is much more desirable to have all the pairs of SUs rendezvous on different channels and that puts a channel loading constraint to limit the probability for a user to search a certain channel (or location). In view of this, the multichannel rendezvous problem is different from the classical rendezvous search problem in the additional channel loading constraint.

As discussed in [27], CH schemes can be classified into various categories depending on their assumptions.

Asymmetric vs. symmetric: A CH scheme is called *asymmetric* if one SU can be identified as the *sender* and the other SU can be identified as the *receiver*. For asymmetric CH schemes, the sender and the receiver can use different strategies to rendezvous. On the other hand, both SUs in a *symmetric* CH scheme have to follow the same strategy. As such, the performance of asymmetric CH schemes is better than that of symmetric CH schemes.

Synchronous vs. asynchronous: A CH scheme is *synchronous* if the indices of time slots of both SU are the same. Synchronous CH schemes can achieve better performance than asynchronous CH schemes as both SUs know when to start their CH sequences simultaneously.

Homogeneous vs. heterogeneous: A CH scheme is called *homogeneous* if the available channel sets of the two SUs are the same. On the other hand, it is called *heterogeneous* if the available channel sets of the two SUs are different. Clearly, rendezvous in a heterogeneous environment is much more difficult than that in a homogeneous environment.

Oblivious vs. non-oblivious: Oblivious rendezvous is the most challenging setting of the multichannel rendezvous problem, where (i) there are no distinguishable roles of users, (ii) users' clocks are

Table 1
An illustration of the Chinese Remainder Theorem

t	0	1	2	3	4	5
$X_1(t)$	0	3^*	4	0	3	4
$X_2(t)$	1	3^*	1	3	1	3

not synchronized, (iii) users may have different available channel sets, and (iv) there is no universal labelling of the channels. In such an environment, nothing can be learned from a failed rendezvous attempt.

The objective of this book chapter is to provide a tutorial on the multichannel rendezvous problem under various categories/assumptions. Instead of giving rigorous mathematical proofs of the results in the multichannel rendezvous problem, we will provide the needed insights/intuitions to understand these results. The road map of this tutorial is as follows: we give the formal formulation of the rendezvous problem and introduce various assumptions, constraints, and performance metrics in Section 2. We then start from the easiest setting, i.e., the asymmetric setting in Section 3, where we introduce the wait-for-mommy strategy and the modular clock algorithm. In Section 4, we address the symmetric setting with time synchronization, where we introduce the synchronous modular clock algorithm and the associated mathematical tools, including Galois field, orthogonal Latin squares, finite projective planes and quorum systems. Next, we address the symmetric setting without time synchronization in Section 5, where we introduce the sawtooth sequence and the disjoint relaxed difference sets. In Section 6, we address the oblivious rendezvous problem with the assumption of unique user IDs, where we introduce the strong symmetrization mapping of IDs and the two-prime modular clock algorithm. In Section 7, we consider the oblivious rendezvous problem without the assumption of unique user IDs. There we introduce a method to assign a user with exactly two available channels an ID and the complete symmetrization mapping that uses the assigned ID for guaranteed rendezvous. Finally, we conclude the tutorial in Section 8 by discussing some further extensions.

Though there are many mathematical theories associated with the multichannel rendezvous problem, in our view the fundamental theorem for the multichannel rendezvous problem is the *Chinese Remainder Theorem* that was first discovered in the 3rd century AD by the Chinese mathematician Sunzi in Sunzi Suanjing. The Chinese remainder theorem asserts that if n_1 and n_2 are coprime, and if r_1 and r_2 are integers such that $0 \leq r_i < n_i$ for $i = 1$ and 2, then there is one and only one integer x, such that $0 \leq x < n_1 \cdot n_2$ and $(x \bmod n_i) = r_i$ for $i = 1$ and 2. In other words, if user i has the available channel set $\mathbf{c_i} = \{c_i(0), c_i(1), \ldots, c_i(n_i - 1)\}$, $i = 1$ and 2, then each user can simply cycle through its available channel set *repeatedly* and every pair of available channels for user 1 and 2 will appear exactly once in a period of $n_1 \cdot n_2$ time slots if n_1 and n_2 are coprime. In Table 1, we illustrate this by using the available channel sets for the two SUs in Fig. 1, i.e., user 1 has the available channel set $\{0, 3, 4\}$ and user 2 has the available channel set $\{1, 3\}$. The CH sequence of user 1 (resp. user 2) is $X_1(t)$ (resp. $X_2(t)$) in this table. Clearly, all the six pairs of available channels for users 1 and 2 appear exactly once in six time slots and these two users rendezvous on channel 3 at $t = 1$. In view of this, the development of this tutorial will be focused on the Chinese Remainder Theorem and the techniques for solving the coprime problem.

II. Mathematical formulation of the multichannel rendezvous problem

To formulate the multichannel rendezvous problem, let us consider a CRN with N channels (with $N \geq 2$), indexed from 0 to $N-1$. There are two (secondary) users who would like to rendezvous on a common unblocked channel by hopping over these N channels with respect to time. We assume that time is slotted (the discrete-time setting) and indexed from $t = 0, 1, 2, \ldots$. The length of a time slot, typically in the order of 10ms, should be long enough for the two users to establish their communication link on a common unblocked channel. In the literature, slot boundaries of these two users are commonly assumed to be aligned. In the case that slot boundaries of these two users are not aligned, one can double the size of each time slot so that the overlap of two misaligned time slots is not smaller than the original length of a time slot. Denote by $X_1(t)$ (resp. $X_2(t)$) the channel selected by user 1 (resp. user 2) at time t. Let $B(t)$ be the set of channels that are blocked at time t. If a channel is blocked at time t, then the two users will not rendezvous even though they both hop to that channel at time t. Then the time-to-rendezvous (TTR), denoted by T, is the number of time slots (steps) needed for these two users to select a common unblocked channel, i.e.,

$$T = \inf\{t \geq 0 : X_1(t) = X_2(t) \notin B(t)\} + 1, \tag{1}$$

where we add 1 in (1) as we start from $t = 0$.

In the literature, various assumptions are used for speeding up the rendezvous process in the multichannel rendezvous problem. In this tutorial, we will focus on the following assumptions:

(A1) *Asymmetric roles*: these two users play two different roles, e.g., user 1 is the transmitter and user 2 is the receiver.

(A2) *Universal labelling of channels*: these two users have the same labelling of the N channels.

(A3) *No blocked channels*: every channel is available to both users.

(A4) *Nonempty intersection of available channel sets*: the available channel set for user i, $i = 1, 2$, is

$$\mathbf{c}_i = \{c_i(0), c_i(1), \ldots, c_i(n_i - 1)\},$$

where $n_i = |\mathbf{c}_i|$ is the number of available channels to user i, $i = 1, 2$. Then there is at least one channel that is commonly available to the two users, i.e.,

$$\mathbf{c}_1 \cap \mathbf{c}_2 \neq \phi. \tag{2}$$

(A5) *Time synchronization*: the two users have the same indexing of the time slots.

(A6) *Unique ID*: each user is assigned with an L-bit unique ID.

In addition to these assumptions, there are also various constraints in the literature on how the channel hopping (CH) sequences $\{X_1(t), t \geq 0\}$ and $\{X_2(t), t \geq 0\}$ are selected.

(C1) *Independence constraint*: before rendezvous, both users are not able to communicate with each other. It is natural to assume that they select their CH sequences independently, i.e., $\{X_1(t), t \geq 0\}$ and $\{X_2(t), t \geq 0\}$ are two independent random sequences.

(C2) *Uniform load constraint*: the channel load is defined as the maximum probability that a user hops to a particular channel in a particular time slot. To avoid the control channel saturation problem, the channel load of a CH sequence needs to be constrained. The uniform load constraint requires that each available channel of a user is selected with an *equal* probability at any time t.

(C3) *Identical distribution constraint*: in the symmetric and homogeneous setting, both users need to follow the same strategy to select their CH sequences. In such an environment, it is natural to assume that $\{X_1(t), t \geq 0\}$ and $\{X_2(t), t \geq 0\}$ have the same joint distribution.

(C4) *Maximum rendezvous diversity constraint*: every channel in the intersection of the available channel sets of the two users is a rendezvous channel. It is also known as the maximum degree of overlapping constraint in the literature. Such a constraint is needed for solving the PU long-time blocking problem.

There are three commonly used metrics for evaluating the performance of TTR:

(i) expected time-to-rendezvous (ETTR) : the expected time for two users to rendezvous,

(ii) maximum time-to-rendezvous (MTTR): the maximum time for two users to rendezvous when all the channels are available to both users, and

(iii) maximum conditional time-to-rendezvous (MCTTR): the maximum time for two users to rendezvous when there is at least one common available channel between two users.

Note that MTTR is a special case of MCTTR when all the channels are available to both users. In the literature, people sometimes use MTTR and MCTTR interchangeably to denote the maximum time-to-rendezvous for the setting considered in their papers.

The simplest way to generate the CH sequences to satisfy the four constraints [C1-C4] is the random algorithm that selects a channel uniformly at random in a user's available channel set in every time slot. The random algorithm performs amazingly well in terms of ETTR and its ETTR is quite close to the lower bound in oblivious rendezvous (see e.g., [24]). As such, it outperforms many CH algorithms proposed in the literature in terms of ETTR. However, the random algorithm does not have bounded MTTR. Thus, for theoretical analysis, researchers in the field focus mostly on MTTR/MCTTR.

III. Asymmetric Setting

A. Homogeneous available channel sets

Let us also start from the easiest setting by assuming the asymmetric role assumption in (A1) and the assumption of a universal labelling of channels in (A2), and the assumption of no blocked channels in (A3).

Since these two users are allowed to play two different roles, let us consider the "wait-for-mommy" strategy in [11], [69] for constructing the two CH sequences $\{X_1(t), t \geq 0\}$ and $\{X_2(t), t \geq 0\}$. Specifically, user 1 plays the role of "mommy" who uniformly selects at random a channel to start and then cycles through the N channels. On the other hand, user 2 plays the role of "child" who also uniformly selects at random a channel and stays there until it is found by the "mommy." Clearly, T is also uniformly distributed over $[1, \ldots, N]$, i.e., for $\tau = 1, \ldots, N$,

$$\mathsf{P}(T = \tau) = \frac{1}{N}. \tag{3}$$

Thus, the expected TTR (ETTR) is $(N + 1)/2$. Also, as both users starts from a uniformly selected channel, the uniform load constraint in (C2) is satisfied. It was further shown in [22]

Table 2
The wait-for-mommy CH sequences

t	0	1	2	3	4	5
mommy	1	2^*	0	1	2^*	0
child	2	2^*	2	2	2^*	2

that the "wait-for-mommy" strategy minimizes the ETTR among all the CH sequences that satisfy the two constraints in (C1) and (C2).

Theorem 1 (Wait-for-mommy (Theorem 1 and Example 1 in [22])) *Under (A1), (A2) and (A3), the "wait-for-mommy" strategy minimizes the ETTR among all the CH sequences that satisfy the two constraints in (C1) and (C2). It also guarantees that the two users will rendezvous within N time slots, i.e., its MTTR is N.*

In Table 2, we show an example of the CH sequences of the "wait-for-mommy" strategy. In this example, we assume that $N = 3$ and there are three channels, $\{0, 1, 2\}$. The mom cycles through the three channels in the order of channels 1,2 and 0. On the other hand, the child stays on channel 2 all the time. They rendezvous on channel 2 at $t = 1$ and $t = 4$.

To satisfy the maximum rendezvous diversity constraint in (C4), we let the child stay on the same channel for a duration of N time slots and then switch to another channel. By doing so, the assumption in (A3) can be removed. The detailed algorithm for such a generalization is shown in Algorithm 1.

Algorithm 1 The wait-for-mommy strategy with maximum rendezvous diversity

```
1. For any $t$, let $q(t) = \lfloor t/N \rfloor$ and $r(t) = (t \bmod N)$.
2. (Mommy) User 1 generates independently a uniformly distributed random variable $U_1$ over $[0, N-1]$. Then construct its CH sequence as follows:
       $X_1(t) = (r(t) + U_1) \bmod N.$    (4)
3. (Child) User 2 also generates independently a uniformly distributed random variable $U_2$ over $[0, N-1]$. Then construct its CH sequence as follows:
       $X_2(t) = (q(t) + U_2) \bmod N.$    (5)
```

Theorem 2 (The wait-for-mommy strategy with maximum rendezvous diversity (Algorithms 3 and 4 and Theorem 16 in [23])) *Under (A1), and (A2), the wait-for-mommy strategy with maximum rendezvous diversity in Algorithm 1 satisfy the three constraints in (C1), (C2) and (C4). It also guarantees that the two users will rendezvous within N^2 time slots, i.e., its MCTTR is N^2.*

In Table 3, we show an example of the CH sequences of the "wait-for-mommy" strategy with maximum rendezvous diversity. In this example, we assume that $N = 3$ and there are three channels, $\{0, 1, 2\}$. The mom cycles through the three channels in the order of channels 1,2 and

Table 3
The wait-for-mommy CH sequences with maximum rendezvous diversity

t	0	1	2	3	4	5	6	7	8
mommy	1	2^*	0	1	2	0^*	1^*	2	0
child	2	2^*	2	0	0	0^*	1^*	1	1

0. On the other hand, the child stays on channel 2 for three time slots and then switch to another channel. They rendezvous on channel 2 at $t = 1$, channel 0 at $t = 5$, and channel 1 at $t = 6$.

A careful examination of the "wait-for-mommy" strategy reveals that it does not really need the universal labelling of channels. What it needs is that both users have the same channel available sets, i.e., a homogeneous view between the two users. This shows that the asymmetric role assumption in (A1) is the most crucial assumption in simplifying and speeding up the rendezvous process in a CRN.

B. Heterogeneous available channel sets

In this section, we consider the heterogeneous setting where the two users have different available channel sets. In such a setting, we assume that the asymmetric role assumption in (A1) and the assumption on the nonempty intersection of available channel sets in (A4).

Under the heterogeneous environment, using the "wait-for-mommy" strategy is a little bit tricky as the "child" does not know how long it has to stay on the same channel. If it stays on a channel that is not in the available channel set of the "mommy," then it will not be found. To solve such a problem, the solution is for the two users to cycle through their available channels like the "mommy" in the "wait-for-mommy" strategy, but do so with coprime periods. For this, let us introduce the modular clock algorithm in [69] (see Algorithm 2) that cycles through the available channel set with a period p not smaller than the size of its available channel set. When the modular clock k is larger than the size of the available channel set, the algorithm selects a channel uniformly at random from the available channel set. It is easy to see that the modular clock algorithm satisfies both the independence constraint (C1) and the uniform load constraint (C2).

Now suppose user 1 uses p_1 and user 2 uses p_2 in the modular clock algorithm. Let

$$\mathbf{c_1} \times \mathbf{c_2} = \{(c_1(\tau_1), c_2(\tau_2)), \tau_1 = 0, 1, \ldots, n_1 - 1, \tau_2 = 0, 1, \ldots, n_2 - 1\}$$

be the set that contains all the pairs of available channels for users 1 and 2. If p_1 and p_2 are coprime, we know from the Chinese Remainder Theorem that every pair of two channels in $\mathbf{c_1} \times \mathbf{c_2}$ appears at least once in $p_1 \cdot p_2$ time slots. Thus, the modular clock algorithm will guarantee that the two users will rendezvous within $p_1 \cdot p_2$ time slots (with maximum rendezvous diversity) if p_1 and p_2 are coprime. Since we assume the asymmetric role assumption in (A1), one easy way to make sure that p_1 and p_2 are coprime is for user 1 to select an odd number that is not smaller than n_1 and user 2 to select a power of 2 that is not smaller than n_2. Clear, $p_1 \leq n_1 + 1$. On the other hand, $p_2 \leq 2n_2$. This then leads to the following theorem.

Theorem 3 (Modular clock (cf. Theorem 4 in [69])) *Under (A1) and (A4), suppose both users use the modular clock algorithm in Algorithm 2 to generate their CH sequences. If* p_1 *and* p_2

Table 4
The CH sequences of the modular clock algorithm

t	0	1	2	3	4	5	6	7	8	9
$X_1(t)$	1	2	3	4	R	1	2^*	3	4	R
$X_2(t)$	2	5	2	5	2	5	2^*	5	2	5

are coprime, then the modular clock algorithm will guarantee that the two users will rendezvous within $p_1 \cdot p_2$ time slots (with maximum rendezvous diversity). In particular, if user 1 selects an odd number that is not smaller than n_1 and user 2 selects a power of 2 that is not smaller than n_2, then both the independence constraint (C1) and the uniform load constraint (C2) are satisfied and these two users are guaranteed to rendezvous within $2(n_1+1)n_2$ time slots, i.e., its MCTTR is bounded above by $2(n_1+1)n_2$.

Algorithm 2 The modular clock algorithm

```
Input An available channel set $\mathbf{c} = \{c(0), c(1), \ldots, c(n-1)\}$, a period $p \geq n$, a slope $r > 0$ that is relatively prime to $p$, and a bias $0 \leq b \leq p-1$.
Output A CH sequence $\{X(t), t = 0, 1, \ldots\}$ with $X(t) \in \mathbf{c}$.
1. For each $t$, let $k = ((r * t + b) \bmod p)$.
2. If $k \leq n-1$, let $X(t) = c(k)$.
3. Otherwise, select $X(t)$ uniformly at random from the available channel set $\mathbf{c}$.
```

In Table 4, we show an example of the CH sequences of the modular clock algorithm. In this example, we assume that the available channels for user 1 are $\{1,2,3,4\}$ and the available channels of user 2 are $\{2,5\}$. Thus, channel 2 is the common available channel of these two users, and the number of available channels for user 1 (resp. 2) is $n_1 = 4$ (resp. $n_2 = 2$). Since user 1 selects an odd number that is not smaller than n_1 and user 2 selects a power of 2 that is not smaller than n_2, we have $p_1 = 5$ and $p_2 = 2$. Suppose that both users choose $r = 1$ and $b = 0$ in the modular clock algorithm. As p_1 and p_2 are coprime, they will rendezvous in 10 time slots. In this example, they rendezvous on channel 2 at $t = 6$. A channel marked with R indicates that it is a channel selected at random from the available channel set of a user.

IV. Symmetric setting with time synchronization

Now we remove the asymmetric role assumption in (A1) and consider the symmetric setting. The symmetric setting is much more difficult than the asymmetric setting in the previous section. To alleviate the difficulty of the multichannel rendezvous problem a bit, we assume (A2) and (A5) hold, i.e., there is a universal labelling of channels and the clocks are synchronized. Such a setting is called the symmetric setting with time synchronization in this tutorial. Moreover, CH sequences constructed for this setting are generally referred to as *synchronous* CH sequences in the literature.

A. Homogeneous available channel sets without blocked channels

In this section, we also assume (A3) hold, i.e., there are no blocked channels.

1) The synchronous modular clock algorithm in a Galois field: For the multichannel rendezvous problem in the symmetric setting with time synchronization, we introduce the synchronous modular clock algorithm (see Algorithm 3). The CH sequence from this algorithm is based on the mathematical theory of Galois fields [44]. A Galois field $GF(N)$ is a set of N elements with two operations $\oplus$ (addition) and $\otimes$ (multiplication) that satisfy various algebraic properties, including the associative law, the commutative law and the distributive law. Moreover, there exists an identity element for addition $\oplus$, called the zero element, and for every element in $GF(N)$, its additive inverse exists. Similarly, there exists an identity element for multiplication $\otimes$, called the one element, and for every nonzero element, its multiplicative inverse exists. Intuitively, we can add, subtract, multiply and divide in a Galois field as in rational numbers.

It is well-known that a Galois field $GF(N)$ exists if and only if N is a prime power. In particular, if $N = 2$, the addition in $GF(2)$ is the exclusive-OR operation and the multiplication in $GF(2)$ is the AND operation. When N is a prime, the addition is the usual addition with the modulo N operation and the multiplication is the usual multiplication with the modulo N operation, i.e.,

$$(a \oplus b) = ((a + b) \bmod N),$$
$$(a \otimes b) = ((a * b) \bmod N).$$

The operations for $GF(2^m)$ are more involved, but they can be easily implemented by using combinatorial logic circuits and have a lot of applications in error correcting codes and network coding.

Here we assume that N is a prime power. Hence, a Galois field $GF(N)$ with the two operations $\oplus$ and $\otimes$ exist. Denote the N elements in $GF(N)$ as $\{0, 1, 2, \ldots, N-1\}$, where 0 is the zero element (the identity element for $\oplus$) and 1 is the one element (the identity element for $\otimes$). We will use $-a$ to denote the inverse element of a under $\oplus$ and a^{-1} to denote the inverse element of a under $\otimes$. As we can treat these two operations as usual addition and multiplication, it is well-known that $-(a \oplus b) = (-a) \oplus (-b)$, $a \otimes 0 = 0 \otimes a = 0$ and $a \otimes (-b) = (-a) \otimes b = -(a \otimes b)$ for the Galois field $GF(N)$.

Algorithm 3 The synchronous modular clock algorithm (in a Galois field)

```
Input A Galois field $GF(N)$ with the $N$ elements $\{0, 1, 2, \ldots, N-1\}$, where 0 is the zero element (the identity element for $\oplus$) and 1 is the one element (the identity element for $\otimes$), a slope $r \in GF(N)$, and a bias $b \in GF(N)$.
Output A CH sequence $\{X(t), t = 0, 1, \ldots, N\}$ with $X(t) \in \{0, 1, 2, \ldots, N-1\}$
1. For $t = 0$, let $X(t) = r$.
2. For each $t = 1, \ldots, N$, let $X(t) = (r \otimes t) \oplus b$.
```

Clearly, if N is a prime, then $(r \otimes t) \oplus b = ((r * t + b) \bmod N)$ and Algorithm 3 is reduced to the modular clock algorithm in Algorithm 2 for $1 \leq t \leq N$. Such a CH sequence was first proposed in SSCH [13].

For $N = 5$, the CH sequences for $t = 0, 1, 2, 3, 4, 5$ generated with $r = 1, 2, 3, 4$ and $b = 0, 1, 2, 3, 4$ are shown in the $(b+1)^{th}$ row in the following four matrices:

$$\left[\begin{array}{c|ccccc} 1 & 1 & 2 & 3 & 4 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \\ 1 & 3 & 4 & 0 & 1 & 2 \\ 1 & 4 & 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 & 3 & 4 \end{array}\right], \quad \left[\begin{array}{c|ccccc} 2 & 2 & 4 & 1 & 3 & 0 \\ 2 & 3 & 0 & 2 & 4 & 1 \\ 2 & 4 & 1 & 3 & 0 & 2 \\ 2 & 0 & 2 & 4 & 1 & 3 \\ 2 & 1 & 3 & 0 & 2 & 4 \end{array}\right], \tag{6}$$

$$\left[\begin{array}{c|ccccc} 3 & 3 & 1 & 4 & 2 & 0 \\ 3 & 4 & 2 & 0 & 3 & 1 \\ 3 & 0 & 3 & 1 & 4 & 2 \\ 3 & 1 & 4 & 2 & 0 & 3 \\ 3 & 2 & 0 & 3 & 1 & 4 \end{array}\right], \quad \left[\begin{array}{c|ccccc} 4 & 4 & 3 & 2 & 1 & 0 \\ 4 & 0 & 4 & 3 & 2 & 1 \\ 4 & 1 & 0 & 4 & 3 & 2 \\ 4 & 2 & 1 & 0 & 4 & 3 \\ 4 & 3 & 2 & 1 & 0 & 4 \end{array}\right]. \tag{7}$$

Similarly, the CH sequences with $r = 0$ are shown in the following matrix:

$$\left[\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 & 2 & 2 \\ 0 & 3 & 3 & 3 & 3 & 3 \\ 0 & 4 & 4 & 4 & 4 & 4 \end{array}\right]. \tag{8}$$

Theorem 4 *(cf. Theorems 2 and 3 in [22]) Assume that N is a prime power so that $GF(N)$ exists. Suppose that user 1 (resp. user 2) chooses its slope r_1 (resp. r_2) independently and uniformly over $\{0, 1, 2, \dots, N-1\}$ and its bias b_1 (resp. b_2) independently and uniformly over $\{0, 1, 2, \dots, N-1\}$ and both users generate their CH sequences by using the synchronous modular clock algorithm in a Galois field in Algorithm 3. Then under (A2), (A3) and (A5), the two users will rendezvous within $N+1$ time slots and ETTR is $\frac{N+1}{2} + \frac{1}{2} - \frac{1}{2N}$. Moreover, the constraints in (C1), (C2) and (C3) are satisfied.*

Now we explain the intuition behind Theorem 4. For $1 \leq t \leq N$, the CH sequence is simply a "line" in the field $GF(N)$ with the slope r and the bias b. If these two users select the same slopes, they rendezvous at time 0 and then hop as two "parallel lines" afterwards. On the other hand, if these two users select different slopes, then these two lines intersect each other within $[1, N+1]$. To compute the ETTR, one needs to consider the following two cases:(i) both users select the same slope with probability $1/N$, and (ii) both users select different slopes with probability $1 - 1/N$. The ETTR for this first case is 1 and that for the second case is $1 + (N+1)/2$.

As each user selects its slope and its bias uniformly at random in $[0, N-1]$, it is easy to see that the probability that a user hops on a particular channel is $1/N$ and thus the uniform load constraint in (C2) is satisfied. In view of the lower bounds in Theorems 2 of [22], the synchronous modular clock algorithm in a Galois field in Algorithm 3 minimizes the MTTR and ETTR under the three constraints (C1), (C2) and (C3).

In Table 5, we show an example of the CH sequences of the synchronous modular clock algorithm. In this example, we assume that $N = 5$ and there are five channels $\{0, 1, 2, 3, 4\}$. Suppose that user 1 (resp. user 2) chooses $r_1 = 1$ and $b_1 = 0$ (resp. $r_2 = 2$ and $b_2 = 0$) in the

Table 5
The CH sequences of the synchronous modular clock algorithm

t	0	1	2	3	4	5	6	7	8	9	10	11
$X_1(t)$	1	1	2	3	4	0^*	1	1	2	3	4	0^*
$X_2(t)$	2	2	4	1	3	0^*	2	2	4	1	3	0^*

synchronous modular clock algorithm. Then the CH sequences can be generated from the first rows in (6). These two users rendezvous on channel 0 at $t = 5$ and $t = 11$.

2) Orthogonal Latin squares: In this section, we discuss the connection between the synchronous modular clock algorithm and orthogonal Latin squares.

Definition 5 ***(Latin square and orthogonal Latin squares)*** *A Latin square with the set of symbols S is an $|S| \times |S|$ matrix such that every symbol appears exactly once in every row and every column. Two $N \times N$ Latin squares $A = (a_{i,j})$ and $B = (b_{i,j})$ are said to be* orthogonal *if the ordered pairs $(a_{i,j}, b_{i,j})$ are all different for all $i, j = 1, 2, \ldots, N$.*

In the following, we show two 4×4 orthogonal Latin squares with $S = \{0, 1, 2, 3\}$.

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \end{bmatrix}. \tag{9}$$

Merging these two matrices together yields

$$\begin{bmatrix} \{0,0\} & \{1,1\} & \{2,2\} & \{3,3\} \\ \{1,3\} & \{0,2\} & \{3,1\} & \{2,0\} \\ \{2,1\} & \{3,0\} & \{0,3\} & \{1,2\} \\ \{3,2\} & \{2,3\} & \{1,0\} & \{0,1\} \end{bmatrix}. \tag{10}$$

Thus, each of the 16 ordered pairs appears exactly once.

The number of mutually orthogonal Latin squares of order N is not greater than $N - 1$. It is known that if N is a prime power, then there exist $N - 1$ mutually orthogonal Latin squares. These $N - 1$ mutually orthogonal Latin squares in fact correspond to the CH sequences from the synchronous modular clock algorithm with nonzero slopes. For instance, if $N = 5$, there are four mutually orthogonal Latin squares. The $(b+1)^{th}$ row $(b = 0, 1, 2, 3, 4)$ of the r^{th} $(r = 1, 2, 3, 4)$ Latin square is generated by the CH sequence from the synchronous modular clock algorithm with the nonzero slope r and bias b for $t = 1, 2, 3, 4, 5$. This can be seen from (6) and (7) by removing the first columns of these four matrices. Note that even the matrix generated from $r = 0$ is not a Latin square, it is still orthogonal to the four matrices in the sense that the ordered pairs (i, j) are all different for all $i, j = 1, 2, \ldots, N$.

Remark 6 The study of the existence of two orthogonal Latin squares (also known as the Graeco-Latin square) was first proposed by L. Euler in 1782 [38]. He was not able to construct two orthogonal Latin squares of order 6 (known as the 36 officers problem) and then conjectured that there does not exist two orthogonal Latin squares of order N for $N = 4m + 2$, where m

is a nonnegative integer. It was confirmed later by G. Tarry via exhaustive enumeration that there does not exist two orthogonal Latin squares of order 6. However, via extensive computer enumeration, two orthogonal Latin squares of order 10 and order 22 were found and it was later shown that Euler's conjecture is false for all $N \geq 10$. We now know that two orthogonal Latin squares exist for all $N \geq 3$ except $N = 6$.

3) Finite projective planes: In this section, we discuss the connection between the synchronous modular clock algorithm and finite projective planes. Readers interested in this topic may find additional material in [54] and its references. The main point of introducing finite projective planes is that the synchronous CH sequences constructed this way have the ability to learn from each unsuccessful rendezvous optimally.

Definition 7 *A finite projective plane of order* N *is a collection of* $N^2 + N + 1$ *lines and* $N^2 + N + 1$ *points such that*

(P1) *every line contains* $N + 1$ *points,*
(P2) *every point is on* $N + 1$ *lines,*
(P3) *any two distinct lines intersect at exactly one point, and*
(P4) *any two distinct points lie on exactly one line.*

The finite projective plane of order 1 is simply a triangle that consists of three points and three lines. In Fig. 2, we show a finite projective plane of order 2, known as the Fano plane, where points are shown as dots and lines are shown as lines or circles. In this figure, there are 7 lines:

$$\begin{aligned} L_0 &= \{0,1,2\}, \\ L_1 &= \{0,3,4\}, \\ L_2 &= \{0,5,6\}, \\ L_3 &= \{1,3,5\}, \\ L_4 &= \{1,4,6\}, \\ L_5 &= \{2,3,6\}, \quad \text{and} \\ L_6 &= \{2,4,5\}. \end{aligned} \tag{11}$$

If N is a prime power, then there exists a systematic method to construct a finite projective plane of order N via projective geometry over the Galois field $GF(N)$ [16], [17]. However, a finite projective plane may not exist for arbitrary N. It was shown by Bose [16] that there is no projective plane of order 6. Moreover, a much more general theorem by Bruck and Ryser [18] provided a necessary condition for the existence of a finite projective plane of order N when $N = 4m + 1$ or $4m + 2$ for some nonnegative integer m. The necessary condition requires that N to be a sum of two integer squares. However, such a necessary condition is not sufficient. In particular, when $N = 10 = 1^2 + 3^2$, it was shown in [54] by computer enumeration that there is no projective plane of order 10.

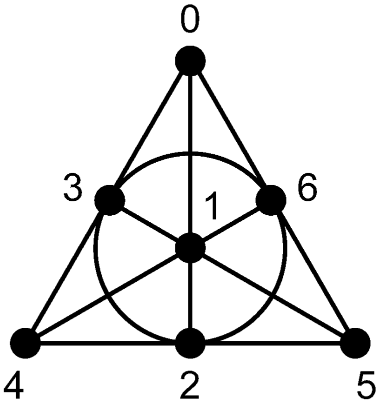

Fig. 2. A finite projective plane of order 2.

Now we show how to construct the synchronous CH sequence from a finite projective plane of order N. Without loss of generality, we index the N^2+N+1 points from 0 to N^2+N. Choose point 0 and select the $N+1$ lines that contain point 0. For each one of these $N+1$ lines, we form a set by removing point 0 in that line. This gives us $N+1$ sets $S_1, S_2, \ldots, S_{N+1}$, and each of them contains exactly N points. Suppose that S_i contains the N points $\{S_{i,0}, S_{i,1}, \ldots, S_{i,N-1}\}$ for $i = 1, 2, \ldots, N+1$. From (P1)-(P4) for a finite projective plane, it is easy to see that the sets S_i, $i = 1, 2, \ldots, N+1$, have the following properties.

(i) These $N+1$ sets are *disjoint* and the union of these $N+1$ sets is the set of points from 1 to N^2+N (excluding point 0).

(ii) Consider a point $S_{1,j_1} \in S_1$ and another point $S_{2,j_2} \in S_2$. As these two points are distinct, there is a unique line, denoted by L_{j_1,j_2}, that contains these two points. Moreover, L_{j_1,j_2} is one of the N^2 lines that does not contain point 0, and it intersects with S_i at exactly one point for $i = 1, 2\ldots, N+1$, i.e., $|L_{j_1,j_2} \cap S_i| = 1$.

(iii) Consider two lines L_{j_1,j_2} and $L_{j_1',j_2'}$. If either $j_1 \neq j_1'$ or $j_2 \neq j_2'$, then these two lines intersect at exactly one point.

(iv) For every point z in the union of these $N+1$ sets, there are exactly N lines in the N^2 lines that contain point z.

Using these four properties, we can construct synchronous CH sequences from Algorithm 4.

Algorithm 4 The finite projective plane algorithm

```
1. Each user selects a point in $S_1$ and another point in $S_2$ uniformly and independently.
2. Suppose that the line determined by these two points is line
       $L = \{S_{i,j_i}, i = 1, 2, \ldots, N+1\}.$
3. Construct the CH sequence $\{X(t), 0 \le t \le N\}$ by assigning $X(t) = j_{t+1}$.
```

Example 8 ($N = 2$) *In this example, we consider the case with $N = 2$. As shown in (11), the three lines L_0, L_1 and L_2 contain point 0. By removing point 0 from these three lines, we form*

$S_1 = \{S_{1,0}, S_{1,1}\} = \{1, 2\}$, $S_2 = \{S_{2,0}, S_{2,1}\} = \{3, 4\}$ and $S_3 = \{S_{3,0}, S_{3,1}\} = \{5, 6\}$. For a point in S_1 and another point in S_2, we have the following four lines:

$$\begin{aligned} L_{1,1} &= \{1, 3, 5\} = L_3, \\ L_{1,2} &= \{1, 4, 6\} = L_4, \\ L_{2,1} &= \{2, 3, 6\} = L_5, \\ L_{2,2} &= \{2, 4, 5\} = L_6. \end{aligned}$$

If $L_{1,1}$ is selected, then the corresponding CH sequence is 000 as $1 = S_{1,0}$, $3 = S_{2,0}$ and $5 = S_{3,0}$. The other three sequences corresponding to $L_{1,2}$, $L_{2,1}$ and $L_{2,2}$ are 011, 101, and 110. Each of these four sequences is selected with probability $1/4$ and both 0 and 1 appear with probability $1/2$ at any time.

As in the synchronous modular clock algorithm, there are two cases to discuss for Algorithm 4: (i) both users select the same point in S_1 with probability $1/N$, and (ii) both users select two different points in S_1 with probability $1 - 1/N$. The ETTR for the first case is 1 and that for the second case is $1 + (N + 1)/2$. If the two users select two different points in S_1, then they are on two different lines and these two lines must intersect at exactly one point later. This corresponds to the orthogonal Latin squares discussed in the previous section.

A fascinating thing for the finite project plane algorithm is its ability to learn from each unsuccessful rendezvous. Suppose that user 1 selects a line L_{j_1,j_2} and it has not met user 2 in $\tau - 1$ time slots. During these $\tau - 1$ time slots, user 1 learns that the line of user 2 does not contain any points that user 1 has searched. Note that for each point z in the union of these $N + 1$ sets, there are exactly N lines in the N^2 lines that contain that point. Excluding line L_{j_1,j_2}, there are $N - 1$ lines containing each point it has searched. Thus, the total number of lines in the N^2 lines that contain a point it has searched is $1 + (\tau - 1)(N - 1)$. As such, user 1 knows that user 2 must be in one of the $N^2 - 1 - (\tau - 1)(N - 1)$ lines that do not contain any point it has searched. Such a learning ability of the finite project plane algorithm was used in [22] to show its optimality in minimizing the TTR.

B. Maximum rendezvous diversity for homogeneous available channel sets

1) The rotation trick: To satisfy the maximum rendezvous diversity constraint in (C4), we use the rotation trick in [23], [67]. In Algorithm 5, we introduce the synchronous modular clock algorithm with maximum rendezvous diversity. The CH sequence generated from Algorithm 5 consists of $N(N + 1)$ time slots. These $N(N + 1)$ time slots are partitioned into N intervals, each with $N + 1$ time slots. The CH sequence in the 0^{th} interval is generated by using Algorithm 3. The CH sequence in the q^{th} interval is the modulo sum of the CH sequence in the 0^{th} interval and q. This is similar to the trick for the wait-for-mommy strategy with maximum rendezvous diversity in Algorithm 1.

In Table 6, we show an example of the CH sequences of the synchronous modular clock algorithm with maximum rendezvous diversity. In this example, we assume that $N = 5$ and there are five channels $\{0, 1, 2, 3, 4\}$. Suppose that user 1 (resp. user 2) chooses $r_1 = 1$ and $b_1 = 0$ (resp. $r_2 = 2$ and $b_2 = 0$) in the synchronous modular clock algorithm with maximum

```
Algorithm 5 The synchronous modular clock algorithm with maximum rendezvous diversity
Input A Galois field $GF(N)$ with the $N$ elements $\{0, 1, 2, \ldots, N-1\}$, a slope $r \in GF(N)$, and a bias $b \in GF(N)$.
Output A CH sequence $\{X(t), t = 0, 1, \ldots, N(N+1)-1\}$ with $X(t) \in \{0, 1, 2, \ldots, N-1\}$
1. For any $t$, let $q(t) = \lfloor t/(N+1) \rfloor$ and $\gamma(t) = (t \bmod (N+1))$.
2. Let $\alpha(t)$ be the output of the synchronous modular clock algorithm in Algorithm 3 at time $\gamma(t)$ by using the Galois field $GF(N)$, the slopr $r$ and the bias $b$ as its input.
3. (Rotation trick) Set $X(t) = ((\alpha(t) + q(t)) \bmod N)$.
```

Table 6
The CH sequences of the synchronous modular clock algorithm with maximum rendezvous diversity

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$q(t)$	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2
$\gamma(t)$	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2
$X_1(t)$	1	1	2	3	4	0^*	2	2	3	4	0	1^*	3	3	4
$X_2(t)$	2	2	4	1	3	0^*	3	3	0	2	4	1^*	4	4	1

t	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
$q(t)$	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
$\gamma(t)$	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
$X_1(t)$	0	1	2^*	4	4	0	1	2	3^*	0	0	1	2	3	4^*
$X_2(t)$	3	0	2^*	0	0	2	4	1	3^*	1	1	3	0	2	4^*

rendezvous diversity. As in Table 5, the CH sequences can be generated from the first rows in (6) with the additional rotation trick. The sequence length is $5 \cdot 6 = 30$. These two users rendezvous on channel 0 at $t = 5$, channel 1 at $t = 11$, channel 2 at $t = 17$, channel 3 at $t = 23$, and channel 4 at $t = 29$.

2) Relaxation of the load constraint: In this section, we address the tradeoff between the channel load and the TTR. One problem of the synchronous modular clock algorithm with maximum rendezvous diversity is that the TTR might be very long when the number of channels N is very large. To solve the long TTR problem for a large number of channels N, one can relax the uniform load constraint in (C2) to a *weaker* channel load constraint below:

(C2') (Channel load constraint) The maximum probability that a user hops to a particular channel at a particular time slot is not greater than $1/u$, i.e., for all $i = 1$ and 2, $j = 0, 1, \ldots, N-1$ and $t \geq 0$,

$$\mathsf{P}(X_i(t) = j) \leq 1/u. \tag{12}$$

Under the channel load constraint in (C2'), there is a lower bound on the MTTR.

Theorem 9 *(Theorem 2 in [23]) For a* symmetric *CH scheme with channel load not greater than $1/u$, if u is an integer and $2 \leq u \leq N$, then its MTTR cannot be smaller than $u+1$.*

Intuitively, relaxing the load constraint speeds up the rendezvous process and thus reduces the MTTR and ETTR. But Increasing the load also increases the congestion level of a control channel. Thus, there is a tradeoff between TTR and congestion level. In [23], the Cycle-Adjustable Channel Hopping (CACH) scheme was introduced to convert a CH sequence satisfying the uniform load constraint to another sequence satisfying a weaker load constraint. The key idea of CACH is to create another layer of logical channels and have the two users rendezvous on logical channels. By using the rotation trick to create a mapping between logical channels and physical channels, CACH still achieves the maximum rendezvous diversity and thus it can still be used for solving the PU long-time blocking problem. The detailed CACH algorithm is shown in Algorithm 6, in which $\alpha(t)$ is the sequence of logical channels and $X(t)$ is the sequence of physical channels. Note that the CACH algorithm is reduced to the synchronous modular clock algorithm with maximum rendezvous diversity in Algorithm 5 when $u = N$. As CACH uses the synchronous modular clock algorithm in Algorithm 3 to construct its CH for the first $u + 1$ time slots, its MTTR is bounded above by $u + 1$ and thus achieves the lower bound in Theorem 9.

Algorithm 6 The cycle adjustable channel hopping (CACH) algorithm

```
Input A system with $N$ channels and a Galois field $GF(u)$ with the $u \leq N$ elements $\{0, 1, 2, \ldots, u - 1\}$, a slope $r \in GF(u)$, and a bias $b \in GF(u)$.
Output A CH sequence $\{X(t), t = 0, 1, \ldots, N(u + 1) - 1\}$ with $X(t) \in \{0, 1, 2, \ldots, N - 1\}$
1. For any $t$, let $q(t) = \lfloor t/(u + 1) \rfloor$ and $\gamma(t) = (t \bmod (u + 1))$.
2. Let $\alpha(t)$ be the output of the synchronous modular clock algorithm in Algorithm 3 at time $\gamma(t)$ by using the Galois field $GF(u)$, the slopr $r$ and the bias $b$ as its input.
3. (Rotation trick) Set $X(t) = ((\alpha(t) + q(t)) \bmod N)$.
```

In Table 7, we show an example of the CH sequences of the cycle adjustable channel hopping (CACH) algorithm. In this example, we assume that $N = 7$ and there are seven channels $\{0, 1, 2, 3, 4, 5, 6\}$. To relax the uniform load constraint, i.e., $1/7$, we choose $u = 5$ so that the load is not greater than $1/5$. Suppose that user 1 (resp. user 2) chooses $r_1 = 1$ and $b_1 = 0$ (resp. $r_2 = 2$ and $b_2 = 0$) in the synchronous modular clock algorithm. As in Table 5, the CH sequences can be generated from the first rows in (6) with the additional rotation trick. The sequence length of each user is $N(u + 1) = 42$. These two users rendezvous on channel 0 at $t = 5$, channel 1 at $t = 11$, channel 2 at $t = 17$, channel 3 at $t = 23$, channel 4 at $t = 29$, channel 5 at $t = 35$, and channel 6 at $t = 41$.

3) Quorum systems: In this section, we introduce the Quorum-based Channel Hopping (QCH) sequence in [15].

Definition 10 *Given a finite universal set $Z_n = \{0, 1, 2, \ldots, n - 1\}$ of n elements, a quorum system H under Z_n is a collection of non-empty subsets of Z_n that satisfies the intersection property:*

$$S' \cap S'' \neq \phi, \; \forall \, S', S'' \in H.$$

Each $S' \in H$ is called a quorum.

Clearly, a finite projective plane of order N is a quorum system under Z_{N^2+N+1}, where each of the $N^2 + N + 1$ lines is a quorum.

Now consider a quorum system $H = \{h_0, h_1, \ldots, h_{m-1}\}$ under Z_n in a CRN with N channels, indexed from $0, 1, 2, \ldots, N - 1$. Each user first randomly selects one of the m quorums from H. Without loss of generality, suppose that h_0 is selected. For the first n times slot (from

Table 7
The CH sequences of the cycle adjustable channel hopping (CACH) algorithm with $N = 7$ and $u = 5$

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$q(t)$	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2
$\gamma(t)$	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2
$X_1(t)$	1	1	2	3	4	0^*	2	2	3	4	5	1^*	3	3	4
$X_2(t)$	2	2	4	1	3	0^*	3	3	5	2	4	1^*	4	4	6

t	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
$q(t)$	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
$\gamma(t)$	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
$X_1(t)$	5	6	2^*	4	4	5	6	0	3^*	5	5	6	0	1	4^*
$X_2(t)$	3	5	2^*	5	5	0	4	6	3^*	6	6	1	5	0	4^*

t	30	31	32	33	34	35	36	37	38	39	40	41			
$q(t)$	5	5	5	5	5	5	6	6	6	6	6	6			
$\gamma(t)$	0	1	2	3	4	5	0	1	2	3	4	5			
$X_1(t)$	6	6	0	1	2	5^*	0	0	1	2	3	6^*			
$X_2(t)$	0	0	2	6	1	5^*	1	1	3	0	2	6^*			

$t = 0, 1, \ldots, n-1$), this user will hop on channel 0 for the time slots in h_0 and randomly select one of the N channels for the time slots not in h_0. As clocks are synchronized, the intersection property of the quorum system ensures that any two users (who might choose different quorums) will rendezvous within n times slots on channel 0. To achieve the maximum rendezvous diversity, QCH partitions time into intervals with each interval consisting of n time slots. It then uses the rotation trick for the channel selected in every interval as described in Section IV-B1

Now suppose we use a finite projective plane of order N to construct a QCH. As there are N^2+N+1 quorums (lines) and every point is on $N+1$ lines, the probability that a user will hop to a particular channel is at least $(N+1)/(N^2+N+1)$. Thus, the load of such a QCH (called L-QCH in [15]) is at least $(N+1)/(N^2+N+1)$. Since the MTTR for such a QCH is N^2+N+1, its MTTR is far from the lower bound in Theorem 9. As such, in the symmetric and synchronous setting, CACH in [23] can achieve a smaller MTTR than L-QCH [15] under the same channel load.

In Table 8, we show an example of the Quorum-based Channel Hopping (QCH) sequences. In this example, we consider the quorum system under Z_7 from the finite projective plane of order 2, i.e., the Fano plane. As show in (11), there are seven channels $\{0, 1, 2, 3, 4, 5, 6\}$ and the quorum system H consists of the following seven sets $\{h_0, h_1, \ldots, h_6\}$:

$$\begin{aligned} h_0 &= \{0,1,2\}, \\ h_1 &= \{0,3,4\}, \\ h_2 &= \{0,5,6\}, \\ h_3 &= \{1,3,5\}, \\ h_4 &= \{1,4,6\}, \\ h_5 &= \{2,3,6\}, \quad \text{and} \\ h_6 &= \{2,4,5\}. \end{aligned} \tag{13}$$

Suppose that user 1 selects $h_0 = \{0, 1, 2\}$ and user 2 selects $h_1 = \{0, 3, 4\}$ in this example. Then the length of such a QCH is 49. These two users rendezvous on channel 0 at $t = 0$, channel 1

Table 8
The Quorum-based Channel Hopping (QCH) sequences

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$X_1(t)$	0^*	0	0	R	R	R	R	1^*	1	1	R	R	R	R
$X_2(t)$	0^*	R	R	0	0	R	R	1^*	R	R	1	1	R	R
t	14	15	16	17	18	19	20	21	22	23	24	25	26	27
$X_1(t)$	2^*	2	2	R	R	R	R	3^*	3	3	R	R	R	R
$X_2(t)$	2^*	R	R	2	2	R	R	3^*	R	R	3	3	R	R
t	28	29	30	31	32	33	34	35	36	37	38	39	40	41
$X_1(t)$	4^*	4	4	R	R	R	R	5^*	5	5	R	R	R	R
$X_2(t)$	4^*	R	R	4	4	R	R	5^*	R	R	5	5	R	R
t	42	43	44	45	46	47	48							
$X_1(t)$	6^*	6	6	R	R	R	R							
$X_2(t)$	6^*	R	R	6	6	R	R							

at $t = 7$, channel 2 at $t = 14$, channel 3 at $t = 21$, channel 4 at $t = 28$, channel 5 at $t = 35$, and channel 6 at $t = 42$. As before, a channel marked with R indicates that it is a channel selected at random from the available channel set of a user.

V. Symmetric setting without time synchronization

In both the asymmetric setting and the symmetric setting with time synchronization, the CH sequences are optimal in the sense of minimizing MTTR and ETTR. This is because these CH sequences have the ability to learn "perfect" information from each unsuccessful rendezvous. For instance, the wait-for-mommy strategy in the asymmetric setting is able to eliminate one channel among the N channels after a failed rendezvous. On the other hand, the finite projective plane algorithm is able to eliminate $N - 1$ lines among the N^2 lines for each channel it has searched.

In this section, we further remove the clock synchronization assumption in the symmetric setting. This is a new territory as learning from a failed rendezvous is much more difficult than before.

A. DRSEQ for homogeneous available channel sets without blocked channels

For the symmetric setting without time synchronization, we introduce the sawtooth sequences (known as DRSEQ in [74]).

Definition 11 ***(Sawtooth sequence [74])*** *A CH sequence $\{f(t), t \geq 0\}$ is called a sawtooth sequence of order N if*

$$f(t) = \begin{cases} t, & \textit{for } t = 0, 1, \dots, N-1 \\ 2N - 1 - t, & \textit{for } t = N, N+1, \dots, 2N-1 \end{cases}, \tag{14}$$

and

$$f(t) = f(t \textit{ mod } (2N + 1)). \tag{15}$$

for $t > 2N + 1$ and $(t \textit{ mod } (2N + 1)) \neq 2N$. On the other hand, for $(t \textit{ mod } (2N + 1)) = 2N$, $f(t)$ is selected at random from the N channels.

Table 9
A sawtooth sequence of order 4

t	0	1	2	3	4	5	6	7	8
$f(t)$	0	1	2	3	3	2	1	0	R
$f(t+1)$	1	2	3	3^*	2	1	0	R	0
$f(t+2)$	2	3	3	2	1	0	R	0^*	1
$f(t+3)$	3	3	2^*	1	0	R	0	1	2
$f(t+4)$	3	2	1	0	R	0	1^*	2	3
$f(t+5)$	2	1^*	0	R	0	1	2	3	3
$f(t+6)$	1	0	R	0	1	2^*	3	3	2
$f(t+7)$	0^*	R	0	1	2	3	3	2	1
$f(t+8)$	R	0	1	2	3^*	3	2	1	0

In view of (15), it is periodic with period $2N+1$ for the rest of time t (except the randomly selected channels). Also, every channel i, $i = 0, 1, \ldots, N-1$, appears exactly twice for $t = 0, 1, \ldots, 2N-1$.

For example, a sawtooth sequence of order 4 is

$$01233210R01233210R\ldots,$$

where R denotes a random channel. One key property of a sawtooth sequence of order N is that for any time shift d, there exists some $0 \le \tau \le 2N-1$ such $f(\tau) = f(\tau+d)$.

In Table 9, we illustrate such a property by considering the case for $N = 4$. Here we show all the time shifted sequences $f(t+d)$, $d = 0, 1, 2, \ldots, 8$. In the first row, we show the sawtooth sequence of order 4 (the case with $d = 0$). A channel marked with R indicates that it is a channel selected at random from the four channels $\{0, 1, 2, 3\}$. Those channels marked with an $*$ in all the subsequent rows are the channels that match their counterparts in the first row.

Theorem 12 *(DRSEQ [74]) Consider a CRN with N channels. Assume that both (A2) and (A3) hold, i.e., the two users have the same labelling of the N channels and all the N channels are available to both users. If both users select their hopping sequences by using the sawtooth sequence of order N, then both users rendezvous within $2N+1$ time slots for any arbitrary clock drift d between these two users. Thus, its MTTR is $2N+1$.*

In addition to the MTTR result, it was shown in [22] by a lengthy and detailed calculation that for $N \ge 2$,

$$\mathsf{E}[T] = N - \frac{1}{6} + \frac{2N^2 + 11N - 4}{6N(2N+1)^2} < N,$$

if both users select their sawtooth sequences *independently*. As such, the ETTR is also better than using the random CH sequences.

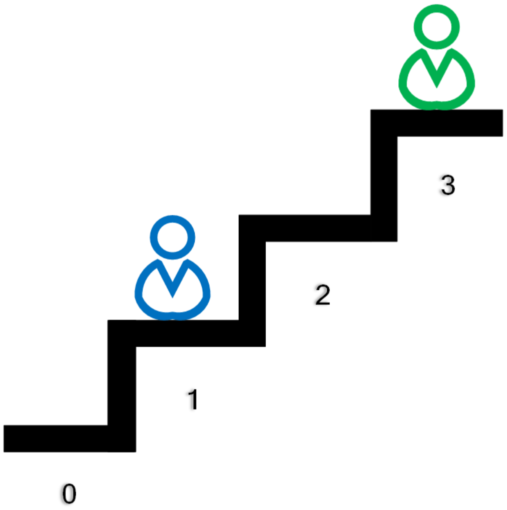

Fig. 3. An illustration of the rendezvous process with the sawtooth sequences.

To see the intuition behind this, let us assume that the random channel is always channel 0. Now view the N channels as a staircase of N steps, and image that both users are climbing up/going down a staircase of N steps according to the sawtooth sequence of order N (see Fig. 3). When one user is climbing up and the other is going down, either they rendezvous on a common step or they just miss each other. In the latter case, the user who is going down will stay an extra time slot on channel 0. This extra time slot changes the "phase" of the user and when it climbs up again, it will rendezvous the other user who is going down from the staircase.

A careful examination of Table 9 reveals that a user that uses the sawtooth sequence (the sequence in the first row) will rendezvous the other user that uses the sawtooth sequence with a specific clock drift d on a specific channel at a specific time. This implies that each unsuccessful rendezvous (except the randomly selected channel) eliminates one possible clock drift among the $2N$ possible clock drifts. In other words, the sawtooth sequence is also able to learn from each unsuccessful rendezvous. However, it is not perfect in the sense that it has an extra time slot for a randomly selected channel.

The sawtooth sequence [74] has the smallest MTTR among all the works in the literature when (A2) and (A3) hold and the CH sequences satisfy the uniform load constraint in (C2). When there is a single permanently blocked channel, it was shown in [22] that one can time-interleave a sawtooth sequence and an inverted sawtooth sequence to guarantee rendezvous within $4N + 2$ time slots.

B. Maximum rendezvous diversity for homogeneous available channel sets

1) Disjoint Relaxed Difference Set (DRDS): Recall that a periodic CH sequence is said to achieve the maximum rendezvous diversity for a CRN with N channels if the two (asynchronous) users rendezvous within the period of the sequence even if there are $N - 1$ blocked channels. In the following definition, we state formally the mathematical properties for an Asynchronous Channel Hopping sequence with Maximum rendezvous diversity (MACH).

Table 10
The $(2, 6)$-MACH

t	0	1	2	3	4	5
$c(t)$	0	0	1	0	1	1
$c(t+1)$	0^*	1	0	1	1^*	0
$c(t+2)$	1	0^*	1^*	1	0	0
$c(t+3)$	0^*	1	1^*	0^*	0	1^*
$c(t+4)$	1	1	0	0^*	1^*	0
$c(t+5)$	1	0^*	0	1	0	1^*

Definition 13 *An (N, p)-MACH sequence $\{c(t), t \geq 0\}$ satisfies the following properties:*

(i) *(Periodicity) $c(t) = c(t + p)$ for all t.*

(ii) *(Maximum rendezvous diversity) For any time shift $0 \leq d \leq p - 1$ and any channel $0 \leq i \leq N - 1$, there exists $\tau(i, d)$ such that $0 \leq \tau(i, d) \leq p - 1$ and $c(\tau(i, d)) = c(\tau(i, d) + d) = i$.*

From the maximum rendezvous diversity property in (ii), we know if there is a time shift d between two users, then user 1 with $X_1(t) = c(t)$ and user 2 with $X_2(t) = c(t + d)$ will rendezvous at channel i at time $\tau(i, d)$. Such a property is also known as the rotation closure property for channel i in the literature (see e.g., [14], [50], [68]). Clearly, for an (N, p)-MACH sequence, its MCTTR is bounded above by its period p.

In [46], the connection between the MACH and the Disjoint Relaxed Difference Set (DRDS) was first made. Let $Z_p = \{0, 1, 2, \ldots, p - 1\}$ be the set of nonnegative integers not larger than p.

Definition 14 *A set $D = \{a_1, a_2, \ldots, a_m\} \subset Z_p$ is called a* Relaxed Difference Set (RDS) *if for every $(d \; mod \; p) \neq 0$, there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j = (d \; mod \; p)$, where $a_i, a_j \in D$.*

It is easy to see that $D = \{1, 2, 4\}$ is a RDS in Z_7. Moreover, if D is an RDS under Z_p, then $D_d = \{(a_i + d) \text{ mod } p | a_i \in D\}$ is also an RDS under Z_p.

Definition 15 *A set $S = \{D_0, D_1, \ldots, D_{N-1}\}$ is called a* Disjoint Relaxed Difference Set (DRDS) *under Z_p if (i) for all $D_i \in S$, D_i is an RDS under Z_p, and (ii) for all $D_i, D_j \in S$, $i \neq j$, $D_i \cap D_j = \phi$.*

In view of Definition 13 and Definition 15, there is a one-to-one mapping between an (N, p)-MACH sequence and a DRDS under Z_p with size N. To see this, one simply maps $c(t) = i$ for $0 \leq t \leq p - 1$ if $t \in D_i$. For $N = 2$, $S = \{\{0, 1, 3\}, \{2, 4, 5\}\}$ is a DRDS under Z_6. Thus, the corresponding $(2, 6)$-MACH is 001011. In Table 10, we illustrate that the maximum rendezvous property of the $(2, 6)$-MACH. Those channels marked with an $*$ in all the subsequent rows are the channels that match their counterparts in the first row.

In general, finding the maximum DRDS is hard. When N is a prime, a linear-time algorithm was proposed in [46] to find a DRDS under Z_p with size N, where $p = 3N^2$. This then leads to an $(N, 3N^2)$-MACH when N is a prime. A lower bound for the period p of any (N, p)-MACH sequence was also shown in Theorem 3 of [46].

Table 11
The $(8, 73)$-MACH sequence in [50]

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$X_1(t)$	R	1	1*	2	1	3	2	3	1	4	3	5	2	6	3
$X_2(t)$	4	R	1*	1	2	1	3	2	3	1	4	3	5	2	6
t	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
$X_1(t)$	5	1	7	4	2	3	5	5*	2	2*	0	6	0	3	6
$X_2(t)$	3	5	1	7	4	2	3	5*	5	2*	2	0	6	0	3
t	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
$X_1(t)$	5	6	1	7	7*	0	4	1	2	3	3*	4	5	6	5
$X_2(t)$	6	5	6	1	7*	7	0	4	1	2	3*	3	4	5	6
t	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
$X_1(t)$	7	2	5	2	0	0*	6	6*	7	0	1	3	4	6	7
$X_2(t)$	5	7	2	5	2	0*	0	6*	6	7	0	1	3	4	6
t	60	61	62	63	64	65	66	67	68	69	70	71	72		
$X_1(t)$	5	0	6	7	1	4	7	0	7	4	0	4	4*		
$X_2(t)$	7	5	0	6	7	1	4	7	0	7	4	0	4*		

Theorem 16 *(Theorem 3 of [46]) For any (N, p)-MACH sequence, its period p has the following lower bound:*

$$p \geq \begin{cases} N^2 + N & \textit{if } N \leq 2 \\ N^2 + N + 1 & \textit{if } N \geq 3 \textit{ and N is a prime power} \\ N^2 + 2N & \textit{otherwise} \end{cases}.$$

The lower bound is not always tight. Finding the minimum period is equivalent of finding the maximum DRDS. As it is hard to find the maximum DRDS, it is also hard to find the tight lower bound for MACH. Via extensive computer enumeration, it was shown in [46] that the lower bound is tight when $N = 1, 2, 5, 6$. For $N = 8$, an explicit $(8, 73)$-MACH sequence was shown in [50]. If N is a prime, the CRSEQ scheme in [68] is an $(N, N(3N - 1))$-MACH sequence. Other constructions of MACHs can be found in the following papers: JS [57], ASYNC-ETCH [78], FRCH [25], SARAH [27], HH [71], T-CH [26] and S-QCH [66].

In Table 11, we show the $(8, 73)$-MACH sequence in [50]. In this example, there are eight channels $\{0, 1, 2, 3, 4, 5, 6, 7\}$ and the length of such a sequence is 73. The sequence of user 1 is the $(8, 73)$-MACH sequence in [50] and the sequence of user 2 is circularly shifted by 1 (to emulate the clock drift of 1). These two users rendezvous on channel 0 at $t = 50$, channel 1 at $t = 2$, channel 2 at $t = 24$, channel 3 at $t = 40$, channel 4 at $t = 72$, channel 5 at $t = 22$, channel 6 at $t = 52$, and channel 7 at $t = 34$. As before, a channel marked with R indicates that it is a channel selected at random from the available channel set of a user.

2) Hierarchical construction of MACH sequences: For $N = 1, 2, 5, 6$ and 8, we know there exist optimal MACH sequences. In this section, we show how one can use these optimal MACH sequences to construct *good* MACH sequences that can have a shorter period than the best-known MACH sequence in the literature. The idea for this is the hierarchical construction that constructs a MACH sequence by using two smaller MACH sequences as shown in the following theorem in [23].

Theorem 17 *(Theorem 19 of [23]) Consider two sequences: an* (N_1, p_1)*-MACH sequence* $\{c_1(t), t \geq 0\}$ *and an* (N_2, p_2)*-MACH sequence* $\{c_2(t), t \geq 0\}$*. For any time* t*, let* $q(t)$ *be the quotient of* t *divided by* $2p_1 - 1$ *and* $r(t)$ *be the corresponding remainder, i.e.,*

$$q(t) = \lfloor t/(2p_1 - 1) \rfloor, \tag{16}$$

and

$$r(t) = (t \textit{ mod } (2p_1 - 1)). \tag{17}$$

Let

$$c(t) = c_1(r(t)) + c_2(q(t)) * N_1. \tag{18}$$

Then the CH sequence $\{c(t), t \geq 0\}$ *is an* $(N_1 * N_2, (2p_1 - 1) * p_2)$*-MACH sequence.*

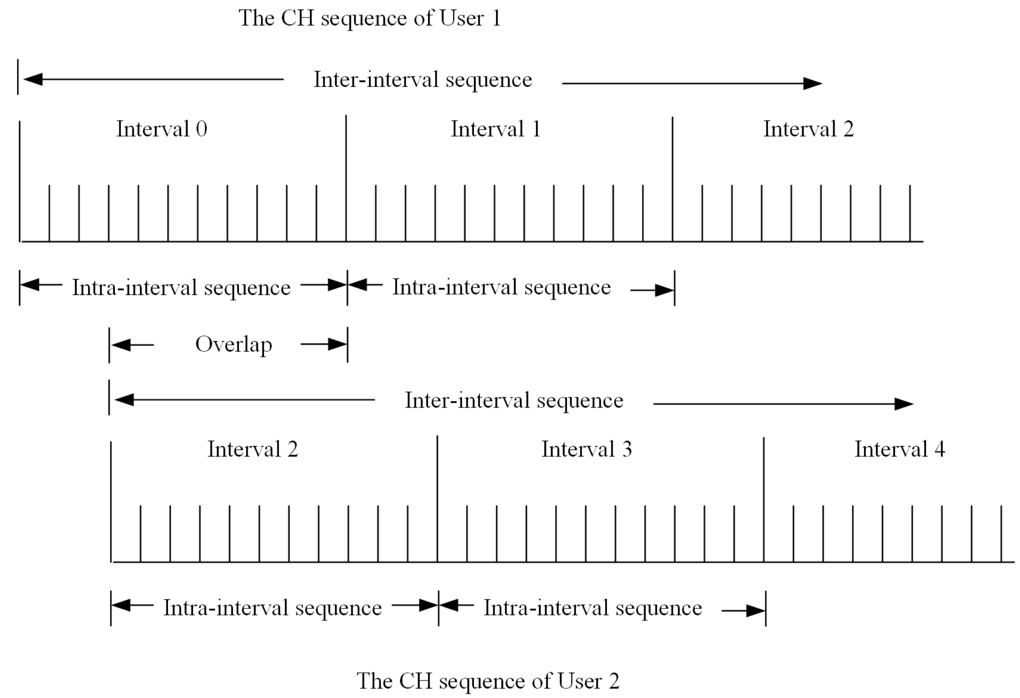

Fig. 4. An illustration of the hierarchical construction of an MACH sequence. As the clocks of the two users are not synchronized, the interval boundaries may not be aligned. However, the overlap of two misaligned intervals is not smaller than p_1 time slots.

To see the intuition of the hierarchical construction (see Fig. 4), one may group every $2p_1 - 1$ slots into an *interval*. Then the hierarchical construction uses the (N_1, p_1)-MACH sequence as the *intra-interval* sequence and uses the (N_2, p_2)-MACH sequence as the *inter-interval* sequence. Clearly, the constructed $(N_1 * N_2, (2p_1 - 1) * p_2)$-MACH is periodic with the period $(2p_1 - 1) * p_2$. Note that every channel in the $\{0, 1, 2, \ldots, N_1 * N_2 - 1\}$ channels can be represented uniquely by $c_1 + c_2 \times N_1$ for some integer c_1 in $\{0, 1, \ldots, N_1 - 1\}$ and some integer c_2 in $\{0, 1, \ldots, N_2 - 1\}$. As such, the property of maximum rendezvous diversity then follows from that of the (N_1, p_1)-MACH sequence and that of the (N_2, p_2)-MACH sequence. The reason why each interval consists

Table 12
A hierarchical construction of $(4, 66)$-MACH sequence by using two $(2, 6)$-MACH seqeunces

t	0	1	2	3	4	5	6	7	8	9	10
$X_1(t)$	0	0^*	1	0	1	1^*	0	0^*	1	0	1
$X_2(t)$	3	0^*	0	1	0	1^*	1	0^*	0	1	0

t	11	12	13	14	15	16	17	18	19	20	21
$X_1(t)$	0	0^*	1	0	1	1^*	0	0^*	1	0	1
$X_2(t)$	1	0^*	0	1	0	1^*	1	0^*	0	1	0

t	22	23	24	25	26	27	28	29	30	31	32
$X_1(t)$	2	2^*	3	2	3	3^*	2	2^*	3	2	3
$X_2(t)$	1	2^*	2	3	2	3^*	3	2^*	2	3	2

t	33	34	35	36	37	38	39	40	41	42	43
$X_1(t)$	0	0^*	1	0	1	1^*	0	0^*	1	0	1
$X_2(t)$	3	0^*	0	1	0	1^*	1	0^*	0	1	0

t	44	45	46	47	48	49	50	51	52	53	54
$X_1(t)$	2	2^*	3	2	3	3^*	2	2^*	3	2	3
$X_2(t)$	1	2^*	2	3	2	3^*	3	2^*	2	3	2

t	55	56	57	58	59	60	61	62	63	64	65
$X_1(t)$	2	2^*	3	2	3	3^*	2	2^*	3	2	3
$X_2(t)$	3	2^*	2	3	2	3^*	3	2^*	2	3	2

of $2p_1 - 1$ slots is to ensure that the overlap of two misaligned intervals is not smaller than p_1 time slots.

As an illustrating example, one can use the $(8, 73)$-MACH sequence in [50] as the two MACH sequences in Theorem 17 to construct a $(64, 10585)$-MACH sequence. On the other hand, one can also use the CRSEQ scheme in [68] to construct a $(67, 13400)$-MACH sequence for a system of 64 channels. For a system of 64 channels, the hierarchical construction yields a better MCTTR than the CRSEQ scheme. In view of this, one can search for good MACH sequences with a moderate number of channels and then use the hierarchical construction to construct a MACH sequence for systems with a large number of channels.

In Table 12, we show the hierarchical construction of a $(4, 66)$-MACH sequence by using two $(2, 6)$-MACH sequences. In this case, we have $N_1 = N_2 = 2$ and $p_1 = p_2 = 6$. Moreover, both $c_1(t)$ and $c_2(t)$ are periodic with period 6 and

$$\{c_1(t), t \geq 0\} = \{c_2(t), t \geq 0\} = 001011001011...$$

The sequence of user 1, i.e., $X_1(t)$, is the constructed $(4, 66)$-MACH sequence, and the sequence of user 2, i.e., $X_2(t)$, is circularly shifted by 1 (to emulate the clock drift of 1). These two users rendezvous on channel 0 at $t = 1, 7, 12, 18, 34, 40$, channel 1 at $t = 5, 16, 38$, channel 2 at $t = 23, 29, 45, 51, 56, 62$, and channel 3 at $t = 27, 49, 60$.

VI. Oblivious rendezvous with unique user IDs

Now we consider the most challenging setting of the multichannel rendezvous problem, i.e., is the oblivious rendezvous problem in heterogeneous CRNs, where (i) there are no distinguishable roles of users, (ii) users' clocks are not synchronized, (iii) users may have different available channel sets, and (iv) there is no universal labelling of the channels. In such an environment, nothing can be learned from a failed rendezvous attempt. Thus, in order to have a guaranteed rendezvous, we need additional assumptions. In this section, we assume that the intersection of available channel sets is nonempty in (A4) and each user is assigned with an L-bit unique ID in (A6).

A. Symmetrization mapping

One common approach for solving the oblivious rendezvous problem is to use the Chinese Remainder Theorem for the modular clock algorithm as described in Section 3.2. In order for the Chinese Remainder Theorem to work, the periods of the modular clock algorithm selected by the two users have to be coprime. If we had the asymmetric role assumption in (A1), then this is rather easy as shown in Theorem 3, where user 1 can select an odd number that is not smaller than n_1 and user 2 can select a power of 2 that is not smaller than n_2.

In view of this, the key insight is that we need something to break the symmetry between these two users. For this, we will use the unique ID assumption of each user. For instance, we can obtain the binary representation of an ID and ask a user to play one role (resp. another role) in a certain time slot if the bit of the ID in that time slot is 1 (resp. 0). As an ID is assumed to be unique, there must be some time slots in which these two users play different roles (if users' clocks are synchronized). Since users' clocks are not synchronized, we need an extra step to map a unique ID into a cyclically unique codeword. This is called the symmetrization mapping in [28].

Definition 18 (Rotation) *Consider an M-vector $\mathbf{w} = \big(w(0), w(1), \dots, w(M-1)\big)$. Define Rotate$(\mathbf{w}, d)$ be the M-vector that is obtained from rotating the elements in $\mathbf{w}$ d times, i.e.,*

$$\textit{Rotate}(\mathbf{w}, d) = \big(w(d), w(d+1), \dots, w((M-1+d) \textit{ mod } M)\big).$$

Definition 19 (Symmetrization mapping [28]) *A set of M-bit codewords $\{\mathbf{w}_i = (w_i(0), w_i(1), \dots, w_i(M-1)), i = 1, 2, \dots, K\}$ is called an M-*symmetrization class *if it satisfies the cyclically unique property, i.e., for any integer d and $i \neq j$,*

$$\mathbf{w}_i \neq \textit{Rotate}(\mathbf{w}_j, d).$$

*A one-to-one mapping from the set of L-bit unique IDs to an M-symmetrization class is called an M-*symmetrization mapping.

For instance, a 4-symmetrization class consists of the following six codewords:

$$\{0000, 1111, 0110, 0101, 1110, 0001\}.$$

For a 48-bit MAC address, it was proposed in [14] that adding another 48 bits of 1's and 48 bits of 0's to the MAC address results in a 144-bit codeword. It is easy to see such a mapping is indeed a 144-symmetrization mapping from a 48 bit ID. A mapping algorithm that requires $M = L + \lceil\sqrt{L}\rceil(2 + \lceil\log_2 L\rceil) + 3$ was proposed in [28]. For $M = 48$, this requires L to be 107.

Now we show how symmetrization mapping can be used with the modular clock algorithm in Algorithm 2 to construct CH sequences for oblivious rendezvous. For this, each user constructs two sequences from the modular clock algorithm: the 0-sequence with the period p_0 and the 1-sequence with the period p_1. The slopes (resp. biases) of both 0/1-sequences are set to 1 (resp.

0). The final CH sequence of a user is constructed by interleaving M 0/1-sequences according to the binary value of its M-bit codeword. Specifically, let $\alpha_0(t)$ (resp. $\alpha_1(t)$) be the 0-sequence (resp. 1-sequence) at time t. If $w(\tau) = 0$ (resp. 1) for $0 \le \tau \le M-1$, then we set the CH sequence $X(\tau+qM) = \alpha_0(q)$ (resp. $X(\tau+qM) = \alpha_1(q)$) for $q = 0, 1, 2, \ldots$. From the cyclically unique property of a symmetrization mapping, we know that the M-bit codeword of user 1 has at least one bit that is different from that of user 2 for any clock drift. Thus, as long as the period of any 0-sequence is relatively prime to the period of any 1-sequence, the Chinese Remainder Theorem guarantees the rendezvous of any two users within $Mp_{0,\max}p_{1,\max}$ time slots, where $p_{0,\max}$ (resp. $p_{1,\max}$) is the maximum of the periods of the 0-sequences (resp. 1-sequences). As in Theorem 3, one simple choice is to set the period $p_{i,0}$ of the 0-sequence of user i to be a power of 2 that is not smaller than n_i, i.e.,

$$p_{i,0} = 2^{\lceil \log_2 n_i \rceil}, \tag{19}$$

(where $\lceil x \rceil$ is the ceiling function that represents the smallest integer that is not less than x), and the period $p_{i,1}$ of the 1-sequence of user i to an odd number that is not smaller than n_i. Clearly, for such a choice, $p_{1,0}$ and $p_{2,1}$ are coprime, and $p_{2,0}$ and $p_{1,1}$ are also coprime. Moreover, for $i = 1$ and 2,

$$n_i \le p_{i,1} \le n_i + 1, \tag{20}$$

$$n_i \le p_{i,0} < 2n_i. \tag{21}$$

From (20) and (21), any two users will rendezvous within $2Mn_{\max}(n_{\max}+1)$ time slots, where $n_{\max} = \max[n_1, n_2]$ is the maximum number of available channels.

In Table 13, we show an example how symmetrization mapping can be used with the modular clock algorithm in Algorithm 2 to construct CH sequences for oblivious rendezvous. In this example, we assume that the available channels of user 1 are $\{1, 2\}$ and the available channels of user 2 are $\{2, 3\}$. Thus, channel 2 is the common available channel of these two users, and the number of available channels for user 1 (resp. 2) is $n_1 = 2$ (resp. $n_2 = 2$). Moreover, we have from (19) and (20) that $p_{1,0} = p_{2,0} = 2$ and $p_{1,1} = p_{2,1} = 3$. The 0-sequence (resp. 1-sequence) of user 1 is then $121212\ldots$ (resp. $12R12R\ldots$) assuming the slope $r = 1$ and the bias $b = 0$ in the modular clock algorithm. On the other hand, the 0-sequence (resp. 1-sequence) of user 2 is then $232323\ldots$ (resp. $23R23R\ldots$). Now assume that $M = 4$ and that user 1 (resp. user 2) uses the 4-bit codeword 0110 (resp. 0101) from the 4-symmetrization class. In this case, we have $Mp_{0,\max}p_{1,\max} = 4 \cdot 2 \cdot 3 = 24$. In this example, $w_1(t)$ (resp. $w_2(t)$) is the sequence that indicates whether the 0-sequence or 1-sequence should be used by user 1 (resp. user 2), and $X_1(t)$ (resp. $X_2(t)$) is the interleaved CH sequence of user 1 (res. user 2). These two users rendezvous on channel 2 at $t = 15, 18$.

B. Strong symmetrization mapping

There are still two shortcomings of the above approach:

(i) The symmetrization mapping from the L-bit unique ID into another M-bit cyclic unique codeword in the literature is not easy to implement if we would like to keep M close to L [28].

Table 13
The CH sequences of generated by using symmetrization mapping

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$w_1(t)$	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
$X_1(t)$	1	1	1	1	2	2	2	2	1	R	R	1	2	1	1	2^*	1	2	2^*	1	2	R	R	2
$w_2(t)$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$X_2(t)$	2	2	2	2	3	3	3	3	2	R	2	R	3	2	3	2^*	2	3	2^*	3	3	R	3	R

Table 14
The 4B5B encoding table

4B data	5B code	4B data	5B code
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

(ii) Even though the MCTTR is bounded, the ETTR is rather poor in comparison with the random algorithm in [69]. This is because $p_{i,0}$ and $p_{j,0}$ in (19) are not relatively prime to each other and a lot of time slots are wasted when both users are using their 0-sequences.

To address these two problems, we need a stronger property than the symmetrization class in [28].

Definition 20 (Strong symmetrization mapping) *Consider a set of M-bit codewords $\{\mathbf{w}_i = (w_i(0), w_i(1), \ldots, w_i(M-1)), i = 1, 2, \ldots, K\}$. Suppose Rotate$(\mathbf{w}_i, d) = (\tilde{w}_i(0), \tilde{w}_i(1), \ldots, \tilde{w}_i(M))$, $i = 1, 2, \ldots, K$. Then this set of codewords is called a* strong M-symmetrization class *if for any integer d and $i \neq j$, (at least) one of the following two properties is satisfied:*

(i) *There exist $0 \le \tau_1, \tau_2 \le M-1$ such that $(w_i(\tau_1), \tilde{w}_j(\tau_1)) = (1, 0)$ and $(w_i(\tau_2), \tilde{w}_j(\tau_2)) = (0, 1)$.*

(ii) *There exist $0 \le \tau_1, \tau_2 \le M-1$ such that $(w_i(\tau_1), \tilde{w}_j(\tau_1)) = (0, 0)$, and $w_i(\tau_2) \neq \tilde{w}_j(\tau_2)$.*

A one-to-one mapping from the set of L-bit unique IDs to a strong M-symmetrization class is called a strong M-symmetrization mapping.

Clearly, a strong M-symmetrization class is an M-symmetrization class in [28]. To construct a *strong M-symmetrization mapping*, the $\mathcal{C}$-transform in [21], [32] was used in [24]. A much simpler strong symmetrization mapping than the $\mathcal{C}$-transform is to use the standard 4B5B coding. The 4B5B encoding scheme is widely used in computer networks (see e.g., [65]). In such an encoding scheme, each piece of 4 bits is uniquely mapped to a 5-bit codeword (see Table 14).

One salient feature of the 4B5B encoding scheme is that each 5-bit codeword has at most one leading 0 as well as at most two trailing 0's. Thus, encoding the L-bit ID results in a $\lceil L/4 \rceil * 5$-bit codeword that does not have 4 consecutive 0's. Now we add the 6-bit delimiter 100001 in front of the $\lceil L/4 \rceil * 5$-bit codeword to construct an $M = \lceil L/4 \rceil * 5 + 6$ codeword. The details of the mapping from an L-bit ID to an M-bit codeword is shown in Algorithm 7. It was shown in [24] that Algorithm 7 is indeed a strong symmetrization mapping. The intuition behind this is that the substring of 4 consecutive 0's only appears in the 6-bit delimiter 100001 and thus it appears exactly once in the M-bit codeword for any cyclic shift d. The two conditions in Definition 20 correspond to the two cases: (i) the 6-bit delimiters are not aligned and (ii) the 6-bit delimiters are aligned.

Algorithm 7 The 4B5B strong symmetrization mapping

```
Input An $L$-bit unique ID.
Output An $M$-bit codeword $\big(w(0), w(1), \ldots, w(M-1)\big)$, where $M = \lceil L/4 \rceil * 5 + 6$.
1. If $L$ is not an integer multiple of 4, append $4 - (L \bmod 4)$ 0's to the unique ID to form a $\lceil L/4 \rceil * 4$-bit ID.
2. Use the 4B5B encoding scheme to encode the $\lceil L/4 \rceil * 4$-bit ID into a $\lceil L/4 \rceil * 5$-bit codeword.
3. Add the 6-bit delimiter 100001 in front of the $\lceil L/4 \rceil * 5$-bit codeword to form a $(\lceil L/4 \rceil * 5 + 6)$-bit codeword.
```

C. Two-prime modular clock algorithm

Now we combine the modular clock algorithm in Algorithm 2 and the strong symmetrization mappings in Algorithm 7 to construct a CH that can provide guaranteed rendezvous under the assumption that each user is assigned with a unique ID. Such an algorithm is called the *two-prime modular clock algorithm* in [24] and its detail is shown in Algorithm 8. The idea, as described before, is to interleave M 0/1-sequences according to the binary value of its M-bit codeword from the strong symmetrization mapping of the L-bit ID. For user i, we select two *primes* $p_{i,0}$ and $p_{i,1}$ such that $n_i \leq p_{i,0} < p_{i,1}$. A 0-sequence (resp. 1-sequence) of user i is then constructed by using the modular clock algorithm with the prime $p_{i,0}$ (resp. $p_{i,1}$). The slope parameter and the bias parameter are determined by pseudorandom number generators so that these two parameters appear to be "random." Then the CH sequence of a user is constructed by interleaving M 0/1-sequences according to its M-bit codeword.

We note that it is possible that $p_{1,0} = p_{2,1}$ (or $p_{2,0} = p_{1,1}$) and thus the previous coprime argument for interleaving M 0/1-sequences according to the cyclically unique property for a symmetrization mapping fails. Fortunately, the two properties for a strong symmetrization mapping are much stronger than the cyclic unique property for a symmetrization mapping and we can use them to prove guaranteed rendezvous in the following theorem.

Theorem 21 *(Theorem 9 in [24]) Suppose the assumptions in (A4) and (A6) hold and all the two users use Algorithm 8 to generate their CH sequences. Then user* 1 *and user* 2 *will rendezvous on every common available channel at least once within* $M \max[p_{1,0}p_{2,1}, p_{1,1}p_{2,0}]$ *time slots.*

Since there is a prime between $[n, 2n]$ [37] and another prime in $[2n, 3n]$ [12], we then have the following corollary.

Corollary 22 *Suppose the assumptions in (A4) and (A6) hold and the two users use Algorithm 8 to generate their CH sequences. Furthermore, user i chooses $p_{i,0}$ as the smallest prime not smaller than n_i and $p_{i,1}$ as the smallest prime larger than $p_{i,0}$, $i = 1$ and 2. Then user 1 and user 2 will rendezvous on every common available channel at least once within $6Mn_1n_2$ time slots. In particular, for the 4B5B strong symmetrization mapping, we have $M = (\lceil L/4 \rceil * 5 + 6)$ for an L-bit ID.*

Algorithm 8 The two-prime modular clock algorithm

```
Input An available channel set $\mathbf{c} = \{c(0), c(1), \ldots, c(n-1)\}$, two primes $p_1 > p_0 \geq n$, and an $L$-bit ID.
Output A CH sequence $\{X(t), t = 0, 1, \ldots\}$ with $X(t) \in \mathbf{c}$.
1. Use a strong $M$-symmetrization mapping (such as Algorithm 7) to construct an $M$-bit codeword $(w(0), w(1), \ldots, w(M-1))$ from the $L$-bit ID.
2. Let $h_1(t,p)$, $h_2(t,p)$ and $h_3(t,p)$ be three pseudorandom number generators that return "independent" and "uniformly distributed" integers in $[0, p-1]$ with the seed $t$.
3. For each $t$, compute the following variables:
4. $q = \lfloor t/M \rfloor$.
5. $s = (t \bmod M)$.
6. $p = p_{w(s)}$.
7. $r = h_1(s, p-1) + 1$.
8. $b = h_2(s, p)$.
9. $k = ((r * q + b) \bmod p)$.
10. If $k \leq n-1$, let $X(t) = c(k)$.
11. Otherwise, let $X(t) = c(h_3(t, n))$ .
```

Now we comment on the ETTR of the two-prime modular clock algorithm. Recall that $h_1(t,p)$, $h_2(t,p)$ and $h_3(t,p)$ are three pseudorandom number generators that return "independent" and "uniformly distributed" integers in $[0, p-1]$ with the seed t. If the same p is used for $t = 0, 1, 2, \ldots, M-1$, then the integers $k(t)$ in Line 9 of the two-prime modular clock algorithm are clearly independent and uniformly distributed in $[0, p-1]$. Even though we use two different primes p_0 and p_1, one can still argue that $X(t)$, $t = 0, 1, 2, \ldots, M-1$, are independent random variables as they are deterministic functions of the three independent random variables, $h_1(s, p-1)$, $h_2(s,p)$ and $h_3(t,n)$. It is also not hard to see that $X(t)$, $t = 0, 1, 2, \ldots, M-1$, are chosen from the available channel set $\mathbf{c} = \{c(0), c(1), \ldots, c(n-1)\}$ with equal probabilities. Thus, for $t = 0, 1, 2, \ldots, M-1$, the two-prime modular clock algorithm behaves as if it were the random algorithm. Thus, if M is large, its ETTR is close to that of the random algorithm. On the other hand, we note that $X(t)$ and $X(t+qM)$ are not independent as they both have the same s and thus the same r and b. Such a correlated property ensures that the MCTTR is bounded as shown in Theorem 21. In view of this, the two-prime modular clock algorithm not only has a bounded MCTTR, but also has the ETTR close to that of the random algorithm.

In the original modular clock algorithm, there is no guarantee that two users can rendezvous within a finite number of time slots (see e.g., Proposition 5 of [69]). The two-prime modular clock algorithm fixes this problem and guarantees that the MCTTR for users 1 and 2 is upper bounded

Table 15
The CH sequences of generated by using the 4B5B strong symmetrization mapping

w_1	1	0	0	0	0	1	0	1	1	1	0
w_2	1	0	0	0	0	1	0	1	0	1	1
t	0	1	2	3	4	5	6	7	8	9	10
$X_1(t)$	1	1	1	1	1	1	1	1	1	1	1
$X_2(t)$	2	2	2	2	2	2	2	2	2	2	2
t	11	12	13	14	15	16	17	18	19	20	21
$X_1(t)$	2	2	2	2	2	2	2	2	2	2	2
$X_2(t)$	3	3	3	3	3	3	3	3	3	3	3
t	22	23	24	25	26	27	28	29	30	31	32
$X_1(t)$	R	1	1	1	1	R	1	R	R	R	1
$X_2(t)$	R	2	2	2	2	R	2	R	2	R	R
t	33	34	35	36	37	38	39	40	41	42	43
$X_1(t)$	1	2	2	2	2	1	2	1	1	1	2^*
$X_2(t)$	2	3	3	3	3	2	3	2	3	2	2^*
t	44	45	46	47	48	49	50	51	52	53	54
$X_1(t)$	2	1	1	1	1	2	1	2	2^*	2	1
$X_2(t)$	3	2	2	2	2	3	2	3	2^*	3	3
t	55	56	57	58	59	60	61	62	63	64	65
$X_1(t)$	R	2	2	2	2	R	2	R	R	R	2
$X_2(t)$	R	3	3	3	3	R	3	R	3	R	R

by $6Mn_1n_2$ for the 4B5B encoding, where $M = (\lceil L/4 \rceil * 5 + 6)$. In particular, when $2^L = K$, the MCTTR upper bound is $O((\log_2 K)n_1n_2)$, which is comparable to the Conversion Based Hopping (CBH) algorithm in [49] and the Advanced Rendezvous Protocol [28]. Moreover, by conducting extensive simulations, its ETTR is almost the same as that of the random algorithm and is much better than other CH algorithms in the literature, including Modified Modular Clock Algorithm [69], FRCH [25], CBH [49] and Advanced Rendezvous Protocol [28].

In Table 15, we show an example how strong symmetrization mapping can be used with the two-prime modular clock algorithm in Algorithm 8 to construct CH sequences for oblivious rendezvous. In this example, we also assume that the available channels of user 1 are $\{1, 2\}$ and the available channels of user 2 are $\{2, 3\}$. Thus, channel 2 is the common available channel of these two users, and the number of available channels for user 1 (resp. 2) is $n_1 = 2$ (resp. $n_2 = 2$). Moreover, we choose $p_{1,0} = p_{2,0} = 2$ and $p_{1,1} = p_{2,1} = 3$. In the two-prime modular clock algorithm, we simply assume that the pseudorandom numbers $h_1(s, p-1) = h_2(s, p) = 0$ so that the slope $r = 1$ and the bias $b = 0$. As such, the 0-sequence (resp. 1-sequence) of user 1 is then $121212\ldots$ (resp. $12R12R\ldots$). On the other hand, the 0-sequence (resp. 1-sequence) of user 2 is then $232323\ldots$ (resp. $23R23R\ldots$). Now assume that user 1 (resp. user 2) has the 4-bit ID 0110 (resp. 0101). By using the 4B5B strong symmetrization mapping in Algorithm 7, the 11-bit codeword of user 1 (resp. user 2) is 10000101110 (resp. 10000101011). In this case, we have $M \max[p_{1,0}p_{2,1}, p_{1,1}p_{2,0}] = 11 \cdot 6 = 66$. The sequence of user 1, i.e., $X_1(t)$, is the interleaved CH sequence by using the 11-bit codeword 10000101110, and the sequence of user 2, i.e., $X_2(t)$, the interleaved CH sequence by using the 11-bit codeword 10000101011. These two users rendezvous on channel 2 at $t = 43, 52$.

VII. Oblivious rendezvous with mapped user IDs

In the previous section, we need the unique ID assumption for oblivious rendezvous. In the oblivious rendezvous problem, it is assumed that there is no universal labelling of the channels. However, in practice, there is a universal labelling of all the channels, even though the total number of channels might be very large. For this, we consider the multichannel rendezvous problem in heterogeneous CRNs with only the two assumptions in (A2) and (A4), i.e., the two users have the same labelling of the N channels and the intersection of available channel sets is nonempty. We do not assume that each user is assigned with an L-bit unique ID in (A6).

A. Complete symmetrization mapping

Our approach for the multichannel rendezvous problem with only the two assumptions in (A2) and (A4) is to extend the strong symmetrization mapping to the complete symmetrization mapping defined below.

Definition 23 (Complete symmetrization mapping) *Consider a set of M-bit codewords*

$$\{\mathbf{w}_i = (w_i(0), w_i(1), \ldots, w_i(M-1)), i = 1, 2, \ldots, K\}.$$

Let

$$\begin{aligned} \textit{Rotate}(\mathbf{w}_i, d) &= (w_i(d), w_i(d+1), \ldots, w_i((d+M-1) \textit{ mod } M)) \\ &= (\tilde{w}_i(0), \tilde{w}_i(1), \ldots, \tilde{w}_i(M)), \end{aligned}$$

be the vector obtained by cyclically shifting the vector $\mathbf{w}_i$ d times. Then this set of codewords is called a complete M-symmetrization class *if either the time shift $(d \textit{ mod } M) \neq 0$ or $i \neq j$, there exist $0 \leq \tau_1, \tau_2, \tau_3, \tau_4 \leq M-1$ such that*

(i) $(w_i(\tau_1), \tilde{w}_j(\tau_1)) = (0, 0)$,
(ii) $(w_i(\tau_2), \tilde{w}_j(\tau_2)) = (1, 1)$,
(iii) $(w_i(\tau_3), \tilde{w}_j(\tau_3)) = (0, 1)$*, and*
(iv) $(w_i(\tau_4), \tilde{w}_j(\tau_4)) = (1, 0)$.

A one-to-one mapping from the set of integers $[1, \ldots, K]$ to a complete M-symmetrization class is called a complete M-symmetrization mapping.

We note that if $i = j$ and $(d \bmod M) = 0$, then $w_i(\tau) = \tilde{w}_i(\tau)$ for all τ and thus conditions (i) and (ii) hold trivially. As such, we know that conditions (i) and (ii) always hold for a complete symmetrization mapping. Moreover, these four conditions in a complete symmetrization mapping are much stronger than those in the strong symmetrization mapping. Thus, we have

$$\text{complete symmetrization} \Rightarrow \text{strong symmetrization} \Rightarrow \text{symmetrization}.$$

In Algorithm 7, we have shown how one can use the standard 4B5B coding to construct a strong M-symmetrization mapping. Here we show how one can construct a complete M-symmetrization mapping by using the Manchester coding (that replaces a bit 0 by the two bits 01 and a bit 1 by the two bits 10).

```
**Algorithm 9** The Manchester complete symmetrization mapping
**Input** An integer $0 \leq x \leq 2^L - 1$.
**Output** An $M$-bit codeword $\big(w(0), w(1), \dots, w(M-1)\big)$ with $M = 2 * L + 10$.
1. Let $\big(\beta_1(x), \beta_2(x), \dots, \beta_L(x)\big)$ be the binary representation of $x$, i.e., $x = \sum_{i=1}^{L} \beta_i(x) 2^{i-1}$.
2. Use the Manchester encoding scheme to encode $x$ into a $2L$-bit codeword, $\big(\beta_1(x), \bar{\beta}_1(x), \beta_2(x), \bar{\beta}_2(x), \dots, \beta_L(x), \bar{\beta}_L(x)\big)$, where $\bar{\beta}_i(x)$ is the (binary) inverse of $\beta_i(x)$.
3. Add the 10-bit delimiter 0100011101 in front of the $2L$-bit codeword to form a ($2L + 10$)-bit codeword.
```

In view of the Manchester mapping in Algorithm 9, we have

$$\begin{aligned}
&(w_x(0), w_x(1), \dots, w_x(M-1)) \\
&= (0, 1, 0, 0, 0, 1, 1, 1, 0, 1, \beta_1(x), \bar{\beta}_1(x), \\
&\qquad \beta_2(x), \bar{\beta}_2(x), \dots, \beta_L(x), \bar{\beta}_L(x)).
\end{aligned}$$

As such, we know that the substring of 3 consecutive 0's only appears in the 10-bit delimiter 0100011101 and thus it appears exactly once in the M-bit codeword for any cyclic shift d. This also holds for the substring of 3 consecutive 1's. Using this, it is easy to show that the Manchester mapping in Algorithm 9 is indeed a complete symmetrization mapping by examining the following four cases: (i) the delimiters of two different codewords are aligned, (ii) the substring of 3 consecutive 0's in one codeword is aligned with the substring of 3 consecutive 1's of the other cyclically shifted codeword, (iii) the substring of 3 consecutive 1's in one codeword is aligned with the substring of 3 consecutive 0's of the other cyclically shifted codeword, and (iv) the substring of 3 consecutive 0's in one codeword is neither aligned with the substring of 3 consecutive 0's nor aligned with the substring of 3 consecutive 1's in the other cyclically shifted codeword. A detailed proof is given in [20].

B. Each user has exactly two channels

Now consider the two-user rendezvous problem with a common channel labelling of the N channels, indexed from 0 to $N-1$. Suppose that each user has exactly two available channels. Since there is a common channel labelling of the N channels, without loss of generality we assume that $c_i(0) < c_i(1)$, $i = 1$ and 2. Let

$$(\beta_1(z), \beta_2(z), \dots, \beta_{\lceil \log_2 N \rceil}(z))$$

be the binary representation of $z \in [0, 1, \dots, N-1]$. Based on the available channel set, we assign user i an integer x_i with

$$x_i = \max\{k \geq 1 : \beta_k(c_i(0)) < \beta_k(c_i(1))\} - 1. \tag{22}$$

The integer $x_i + 1$ is the largest bit that the binary representations of the two available channels $c_i(0)$ and $c_i(1)$ differ. Note that $0 \leq x_i \leq \lceil \log_2 N \rceil - 1$ and thus the binary representation of x_i requires at most $\lceil \log_2(\lceil \log_2 N \rceil) \rceil$ bits.

The integer x_i can be viewed as the mapped ID of user i based on its two available channels. Also, it is possible that these two users are assigned with the same mapped ID. Such an ID

Table 16
The CH sequences of generated by using the Manchester complete symmetrization mapping

w_1	0	1	0	0	0	1	1	1	0	1	1	0
w_2	0	1	0	0	0	1	1	1	0	1	0	1
w_3	0	1	0	0	0	1	1	1	0	1	1	0

t	0	1	2	3	4	5	6	7	8	9	10	11
$X_1(t)$	1	2	1	1	1	2	2	2	1	2	2	1
$X_2(t)$	2	3	2	2	2	3	3	3	2	3	2	3
$X_3(t)$	0	2	0	0	0	2	2	2	0	2	2	0

assignment was previously used in [29], [48]. In the following theorem, we use the Manchester complete symmetrization mapping to improve the MTTR from $16(\lceil \log_2 \log_2 N \rceil + 1)$ in [48] to $2\lceil \log_2(\lceil \log_2 N \rceil) \rceil + 10$ when $n_i = 2$ for all i.

Theorem 24 *(Theorem 4 in [20]) Suppose that the assumption (A2) and the assumption (A4) in (2) hold and $n_i = 2$ for all i. User i uses the Manchester complete symmetrization mapping in Algorithm 9 with the integer x_i in (22) to generate an M-bit codeword $\big(w_{x_i}(0), w_{x_i}(1), \ldots, w_{x_i}(M-1)\big)$ with $M = 2\lceil \log_2(\lceil \log_2 N \rceil) \rceil + 10$. At time t, user i hops on channel $c_i(0)$ (resp. $c_i(1)$) if $w_{x_i}(t \text{ mod } M) = 0$ (resp. 1). Then both users rendezvous within M time slots.*

Since Algorithm 9 is indeed a complete symmetrization mapping, it then follows from the assumption (A4) in (2) and the four conditions (i), (ii), (iii) and (iv) in Definition 23 for a complete symmetrization mapping that these two users rendezvous within M time slots if $x_1 \neq x_2$. Thus, we only need to consider the case that $x_1 = x_2$.

If $x_1 = x_2 = k - 1$ for some k, then the k^{th} bit of the binary representation of $c_1(0)$ and that of $c_2(0)$ are 0, i.e., $\beta_k(c_1(0)) = \beta_k(c_2(0)) = 0$. On the other hand, the k^{th} bit of the binary representation of $c_1(1)$ and that of $c_2(1)$ are 1, i.e., $\beta_k(c_1(1)) = \beta_k(c_2(1)) = 1$. This implies that $c_1(0) \neq c_2(1)$ as their binary representations are different. Similarly, $c_1(1) \neq c_2(0)$. Thus, under the assumption (A4) in (2), we have either $c_1(0) = c_2(0)$ or $c_1(1) = c_2(1)$. Thus, the two conditions (i) and (ii) in Definition 23 suffice to ensure that these two users rendezvous within M time slots.

To further illustrate the physical meaning of the mapped ID x_i in (22), let us consider a CRN with four channels $\{0, 1, 2, 3\}$. Since $N = 4$, $\lceil \log_2 \log_2 N \rceil = 1$ In this CRN, there are three users. The available channel set of user 1 (resp. user 2 and user 3) is $\{1, 2\}$ (resp. $\{2, 3\}$ and $\{0, 2\}$). In this case, we have $x_1 = 1$, $x_2 = 0$, and $x_3 = 1$. It then follows from the Manchester complete symmetrization mapping in Algorithm 9 that the 12-bit codeword for user 1 (resp. user 2 and user 3) is 010001110110 (resp. 010001110101 and 010001110110). The CH sequences of these three users are shown in Table 16. In this example, user 1 and user 2 rendezvous on channel 2 at $t = 10$. Also, user 1 and user 3 rendezvous on channel 2 at $t = 1, 5, 10$.

C. Each user has an arbitrary number of available channels

To extend the result in Theorem 24 to the setting where each user can have an arbitrary number of available channels, we follow the two-prime method in [29]. Partition the time into a sequence of intervals with each interval consisting of $2M - 1$ time slots. The $2M - 1$ time slots in an interval ensure that the overlap between an interval of a user and the corresponding interval of another user consists of at least M time slots even when the clocks of the two users are not synchronized. This is similar to the hierarchical construction of MACH sequences in Section 5.2.2. Within an interval, user i runs the channel hopping sequence in Theorem 24 for a pair of two channels in its available channel set. To select the two channels in an interval, user i first selects two primes $p_{i,1} > p_{i,0} \geq n_i$ (as in the two-prime modular clock algorithm in Algorithm 8). For the t^{th} interval, user i selects one channel according to the modular clock algorithm in Algorithm 2 at time t by using the prime $p_{i,0}$. It also selects the other channel by using the same algorithm and the other prime $p_{i,1}$. Replace the second channel by an arbitrary available channel if these two selected channels are identical. As a direct consequence of Theorem 24 and Theorem 3, we conclude that user i and user j rendezvous within $(2M-1)p_{1,1}p_{2,1}$ time slots under (A2) and (A4). As $p_{i,1} \leq 3n_i$, the MTTR is bounded above $18Mn_1n_2$, where n_i is the number of available channels for user i, $i = 1$ and 2, and $M = 2\lceil \log_2(\lceil \log_2 N \rceil) \rceil + 10$ (with N being the total number of channels).

VIII. Conclusion

In this tutorial, we have addressed the multichannel rendezvous problem with two users under various assumptions. One of the fundamental challenges of the multichannel rendezvous problem is to investigate how much a user can learn from a failed rendezvous attempt and how that information can be used to speed up the rendezvous process. For instance, the wait-for-mommy strategy in the asymmetric setting is able to eliminate one channel among the N channels after a failed rendezvous, and it is optimal in minimizing both ETTR and MTTR among all the CH sequences satisfying the two constraints (C1) and (C2). On the other hand, the finite projective plane algorithm is able to eliminate $N-1$ lines among the N^2 lines for each channel it has searched, and it is optimal in minimizing both ETTR and MTTR among all the CH sequences satisfying the three constraints (C1), (C2) and (C3). The sawtooth sequence (DRSEQ) is also able to eliminate one of the $2N$ clock drifts after a failed rendezvous from a non-random channel, and thus achieves the best MTTR among all the CH sequences in the literature when all the channels are available to both users. Moreover, its ETTR is even better than that of the random algorithm. In the oblivious rendezvous setting, nothing can be learned from a failed rendezvous. As such, the best a user can do is to cycle through its available channels periodically for some period p. As long as its period is relatively prime to the period of the other user, the Chinese Remainder Theorem guarantees the rendezvous of these two users. Both the unique ID assumption and the method of mapped IDs are then used to construct interleaved periodic CH sequences in which there are some coprime periods between these two users.

There are two possible extensions of the multichannel rendezvous problem: (i) multiuser rendezvous and (ii) multiple radios.

A. Multiuser rendezvous

To motivate the study of multiuser rendezvous, it was argued in [1] that a group of users need to rendezvous on the same channel periodically so as to update their common time-dependent group key for secure communication. Such a time-dependent key update cannot be achieved by a sequence of pairwise rendezvous. Other applications and generalizations, including local broadcast and data aggregation, were addressed in [43]. As in [47], [57], when a group of users rendezvous on a channel, a leader is elected and both the state information and clocks are synchronized to those of the leader. The key challenge for the multiuser rendezvous is then how to adjust the CH sequences after a successful rendezvous of a subset of users so as to speed up the rendezvous process. Two different extensions were considered in [24] for the oblivious rendezvous problem with unique user IDs: (i) the stick together algorithm and (ii) the spread out algorithm. In the stick together algorithm, all the users following a leader hop along with the leader. On the other hand, in the spread out algorithm, a user following a leader may hop on one of its available channels that is different from the channel selected by its leader. The simulation results in [24] showed that the ETTR of the stick together algorithm is still almost the same as that of the random algorithm. Moreover, the spread out algorithm is not always better than the stick together algorithm as commonly claimed in the literature (see e.g., [41]). In particular, when the number of common channels among a set of multiple users is very small, the spread out algorithm that hops on one of its available channels does not improve the rendezvous probability.

B. Multiple radios

In this tutorial, we only consider the setting where each user has only one radio. It is possible to extend the results in this tutorial to the setting where each user has more than one radio. For this, let us consider the oblivious rendezvous problem with mapped user IDs in Section 7. There we assume that (A2) and (A4) hold, i.e., the two users have the same labelling of the N channels and the intersection of available channel sets is nonempty. In addition to these two assumptions, we also assume that user i has m_i radios, where $m_i \geq 1$, $i = 1$ and 2. As such, user i can hop on m_i channels in a time slot. Denote by $X_1(t)$ (resp. $X_2(t)$) the set of channels selected by user 1 (resp. user 2) on its m_i radios at time t. Then the time-to-rendezvous (TTR), denoted by T, is the number of time slots needed for these two users to select a common available channel, i.e.,

$$T = \inf\{t \geq 0 : X_1(t) \cap X_2(t) \neq \phi\} + 1, \tag{23}$$

where we add 1 in (23) as we start from $t = 0$.

As now each user has more than one radio, one can convert the single radio CH sequences for multiple radios to speed up the rendezvous process. As in [75], there are two straightforward ways of doing this: (i) *independent sequence* that applies an existing CH algorithm in each radio to generates CH sequence independently for each radio, and (ii) *parallel sequence* that divides an existing CH sequence in the round robin fashion to the multiple ratios. For the independent sequence approach, the MCTTR is still the same as that for a single radio (Theorem 2 of [75]). On the other hand, for the parallel sequence approach, the MCTTR can be reduced to $1/m$ of that for a single radio if both users have the same number of radios m (Theorem 3 of [75]). If both users do not have the same number of radios, then the parallel sequence approach does not guarantee rendezvous. For the setting that each user has more than two radios, the rendezvous search is rather simple as one can simply use the two-prime modular clock algorithm as previously proposed in [56]. But the problem arises when each user only has a single radio. Then the Chinese Remainder Theorem needed for the two-prime modular clock algorithm may not be applicable as they may select the same prime. To address the problem that some users in a CRN might only have a single radio, it was recently proposed in [20] to emulate each radio of a user as it were two radios. Specifically, one constructs CH sequences that group several slots into an *interval* and within an interval each radio selects two channels according to the two-prime modular clock algorithm. This is similar to that discussed in Sections 7.2 and 7.3. When a user has more than one radio, one simply divides its available channels as evenly as possible to its radio(s). By doing so, the number of channels assigned to each radio is much smaller and thus the primes selected by each radio are also smaller. As a result, the MCTTR can be greatly reduced.

REFERENCES

[1] M. J. Abdel-Rahman, H. Rahbari, and M. Krunz, “Multicast rendezvous in fast-varying DSA network,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1449–1462, 2015.

[2] S. Alpern, “The rendezvous search problem,” *SIAM J. Control Optim.*, vol. 33, pp. 673–683, 1995.

[3] S. Alpern, “Rendezvous search on labeled networks,” *Naval Research Logistics*, vol. 49, pp. 256–274, 2002.

[4] S. Alpern and V. Baston, “Rendezvous on a planar lattice,” *Operations Research*, vol. 53, pp. 996–1006, 2005.

[5] S. Alpern and V. Baston, “Rendezvous in higher dimensions,” *SIAM J. Control Optim.*, vol. 44, pp. 2233–2252, 2006.

[6] S. Alpern, V. Baston, and S. Essegaier, “Rendezvous search on a graph,” *J. Appl. Probab.*, vol. 36, pp. 223–231, 1999.

[7] S. Alpern and S. Gal, “Rendezvous search on the line with distinguishable players,” *SIAM J. Control Optim.*, vol. 33, pp. 1270–276, 1995.

[8] S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous*. Dordrecht: Kluwer Academic Publishers, 2003.

[9] E. J. Anderson and S. Essegaier, “Rendezvous search on the line with indistinguishable players,” *SIAM J. Control Optim.*, vol. 33, pp. 1637–1642, 1995.

[10] E. J. Anderson and S. P. Fekete, “Two dimensional rendezvous search,” *Operations Research*, vol. 49, pp. 107–118, 2001.

[11] E. J. Anderson and R. Weber, “The rendezvous problem on discrete locations,” *Journal of Applied Probability,* vol. 28, pp. 839–851, 1990.

[12] M. El Bachraoui, “Primes in the interval [2n, 3n],” *Int. J. Contemp. Math. Sci.*, vol. 1, no. 13-16, pp. 617–621, 2006.

[13] P. Bahl, R. Chandra, J. Dunagan, “SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks,” *ACM MobiCom’04*, 2004.

[14] K. Bian and J.-M. Park, “Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1294-1307, 2013.

[15] K. Bian, J.-M. Park, and R. Chane, “A quorum-based framework for establishing control channels in dynamic spectrum access networks,” *ACM MobiCom’09*, 2009.

[16] R. C. Bose, “On the application of the properties of Galois fields to the problem of construction of hyper-graeco-latin squares,” *Sankhyā: The Indian Journal of Statistics*, pp. 323–338, 1938.

[17] R. C. Bose and K. R. Nair, “On complete sets of latin squares,” *Sankhyā: The Indian Journal of Statistics*, pp. 361–382, 1941

[18] R. H. Bruck and H. J. Ryser, “The nonexistence of certain finite projective planes,” *Canad. J. Math*, **1**(191), p. 9, 1949.

[19] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[20] C.-S. Chang, “A fast multi-radio rendezvous algorithm in heterogeneous cognitive radio networks,” *preprint*, 2017.

[21] C.-S. Chang, J. Cheng, T.-K. Huang and D.-S. Lee, ”Explicit constructions of memoryless crosstalk avoidance codes via C-transform,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 2030–2033, September 2014.

[22] C.-S. Chang, W. Liao and C.-M. Lien, “On the multichannel rendezvous problem: fundamental limits, optimal hopping sequences, and bounded time-to-rendezvous,” *Mathematics of Operations Research,* vol. 40, no. 1, pp. 1-23, 2015.

[23] C.-S. Chang, W. Liao, and T.-Y. Wu, “Tight lower bounds for channel hopping schemes in cognitive radio networks,” *IEEE/ACM Transactions on Netowrking*, vol. 24, no. 4, pp. 2343–2356, 2016.

[24] C.-S. Chang, C.-Y. Chen, D.-S. Lee, and W. Liao, “Efficient encoding of user IDs for nearly optimal expected time-to-rendezvous in heterogeneous cognitive radio networks,” to appear in *IEEE/ACM Transactions on Networking*.

[25] G.-Y. Chang and J.-F. Huang, “A fast rendezvous channel-hopping algorithm for cognitive radio networks,” *IEEE Communications Letters*, vol. 17, no. 7, pp. 1475–1478, 2013.

[26] G. Y. Chang, J. F. Huang, and Y. S. Wang, “Matrix-based channel hopping algorithms for cognitive radio networks,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2755–2768, 2015.

[27] G.-Y. Chang, W.-H. Teng, H.-Y. Chen, and J.-P. Sheu, "Novel channel-hopping schemes for cognitive radio networks," *IEEE Transactions on Mobile Computing,* vol. 13, pp. 407–421, Feb. 2014.

[28] L. Chen, K. Bian, L. Chen, C. Liu, J. M. J. Park, and X. Li, "A group-theoretic framework for rendezvous in heterogeneous cognitive radio networks," In *Proc. ACM MobiHoc*, pp. 165-174, 2014.

[29] S. Chen, A. Russell, A. Samanta, and R. Sundaram, "Deterministic blind rendezvous in cognitive radio networks." *IEEE 34th International Conference on Distributed Computing Systems (ICDCS)*, pp. 358–367, 2014.

[30] T. Chen, H. Zhang, G. M. Maggio, and I. Chlamtac, "CogMesh: a cluster-based cognitive radio network," in *Proc. IEEE DySPAN'07.*

[31] E. J. Chester and R. H. Tütüncü, "Rendezvous search on the labeled line," *Operations Research*, vol. 52, pp. 330–334, 2004.

[32] C.-C. Chou, C.-S. Chang, D.-S. Lee and J. Cheng, "A necessary and sufficient condition for the construction of 2-to-1 optical FIFO multiplexers by a single crossbar switch and fiber delay lines," *IEEE Transactions on Information Theory*, vol. 52, pp. 4519–4531, 2006.

[33] I. H. Chuang, H.-Y. Wu, and Y.-H. Kuo, "A fast blind rendezvous method by alternate hop-and-wait channel hopping in cognitive radio networks," *IEEE Transactions Mobile Computing*, vol. 13, no. 10, pp. 2171–2184, 2014.

[34] L. DaSilva and I. Guerreiro, "Sequence based rendezvous for dynamic spectrum access," *Proc. IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN'08)*, pp. 1-7, Oct. 2008.

[35] A. Dessmark, P. Fraigniaud, D. R. Kowalski, A. Pelc, "Deterministic rendezvous in graphs," *Algorithmica*, vol. 46, pp. 69-96, 2006.

[36] E. O. Elliott, "Estimates of error rates for codes on burst noise channels," *Bell System Technical Journal*, vol. 42, no. 5 pp. 1977–1997, 1963.

[37] P. Erdös, "Beweis eines satzes von tschebyschef," *Acta Litt. Univ. Sci., Szeged, Sect. Math.*, vol. 5, pp. 194–198, 1932.

[38] L. Euler, "Recherches sur une nouvelle espece de quarres magiques," *Zeeuwsch Genootschao*, 1782.

[39] Federal Comm. Commission, "Spectrum policy task force report," Washington, DC, *FCC 02-155*, 2002.

[40] S. Gal, "Rendezvous search on the line," *Operations Research*, vol. 47, pp. 974–976, 1999.

[41] R. Gandhi, C. C. Wang, and Y. C. Hu, "Fast rendezvous for multiple clients for cognitive radios using coordinated channel hopping," In *Proc. IEEE SECON*, pp. 434–442, 2012.

[42] E. N. Gilbert, "Capacity of a burst noise channel," *Bell system technical journal*, vol. 39, no. 5 pp. 1253–1265, 1960.

[43] S. Gilbert, F. Kuhn, C. Newport, and C. Zheng, "Efficient communication in cognitive radio networks," In *Proc. ACM PODC*, pp. 119-128, 2015.

[44] R. P. Grimaldi. *Discrete and Combinational Mathematics: An Applied Introduction*. Addison Wesley 2004.

[45] Z. Gu, Q.-S. Hua, and W. Dai, "Local sequence based rendezvous algorithms for cognitive radio networks," In *Proc. IEEE SECON*, pp. 194–202, 2014.

[46] Z. Gu, Q.-S. Hua, Y. Wang, and F. C. M. Lau, "Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks," in *Proc. IEEE SECON*, 2013.

[47] Z. Gu, Q. S. Hua, Y. Wang, and F. C. M. Lau, "Oblivious Rendezvous in Cognitive Radio Networks," in *International Colloquium on Structural Information and Communication Complexity*, pp. 165–179, 2014.

[48] Z. Gu, H. Pu, Q.-S. Hua, and F. C. M. Lau, "Improved rendezvous algorithms for heterogeneous cognitive radio networks," In *Proc. IEEE INFOCOM*, pp. 154–162, 2015.

[49] Z. Gu, Q.-S. Hua, and W. Dai, "Fully distributed algorithm for blind rendezvous in cognitive radio networks," In *Proc. ACM MobiHoc*, pp. 155–164, 2014.

[50] F. Hou, L. X. Cai, X. Shen, and J. Huang, "Asynchronous multichannel MAC design with difference-set-based hopping sequences," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 1728–1739, 2011.

[51] J. V. Howard, "Rendezvous search on the interval and the circle," *Operations research*, vol. 47, pp. 550–558, 1999.

[52] J. R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," *Mobile Networks and Applications*, vol. 10, no. 1-2, pp. 169-181, 2005.

[53] Y. R. Kondareddy and P. Agrawal, "Synchronized MAC protocol for multi-hop cognitive radio networks," in *Proc. IEEE ICC'08.*

[54] C. W. H. Lam, "The search for a finite projective plane of order 10," *The American mathematical monthly*, vol. 98, no. 4, pp. 305–318, 1991.

[55] L. Le and E. Hossain, "OSA-MAC: a MAC protocol for opportunistic spectrum access in cognitive radio networks," in *Proc. IEEE WCNC'08.*

[56] G. Li, Z. Gu, X. Lin, H. Pu, and Q.-S. Hua, "Deterministic distributed rendezvous algorithms for multi-radio cognitive radio networks," In *Proc. ACM Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp. 313-320, 2014.

[57] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE INFOCOM 2011.*

[58] Z, Lin, H. Liu, X. Chu, and Y.-W. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1742–1745, 2013.

[59] J. Lin, A. S. Morse and B. D. O. Anderson, "The multi-agent rendezvous problem. Part 1: the synchronous case," *SIAM J. Control Optim.*, vol. 46, pp. 2096–2119, 2007.

[60] J. Lin, A. S. Morse and B. D. O. Anderson, "The multi-agent rendezvous problem. Part 2: the asynchronous case," *SIAM J. Control Optim.*, vol. 46, pp. 2120–2147, 2007.

[61] G. D. Marcoa, L. Garganoa, E. Kranakisb, D. Krizancc, A. Pelcd, and U. Vaccaroa, "Asynchronous deterministic rendezvous in graphs," *Theoretical Computer Science*, vol. 355, pp. 315–326, 2006.

[62] J. Mitola III and G. Q. Maguire Jr.,"Cognitive radio: making software radios more personal," *IEEE Personal Communications*, Aug. 1999.

[63] J. Mo, H.-S.W. So, and J. Warland, "Comparison of multichannel MAC protocols," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 50-65, 2008.

[64] M. Nordborg, "Coalescent theory," *Handbook of statistical genetics*, John Wiley & Sons, 2001.

[65] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 4th edition, San Francisco, CA: Morgan Kaufmann Publishers, 2007.

[66] J. P. Sheu, C. W. Su, and G. Y. Chang, "Asynchronous quorum-based blind rendezvous schemes for cognitive radio networks," *IEEE Transactions on Communications*, vol. 64, no. 3, pp.918–930, 2016.

[67] C.-F. Shih, T. Y. Wu, and W. Liao, "DH-MAC: A dynamic channel hopping MAC protocol for cognitive radio networks," in *Proc. IEEE ICC'2010.*

[68] J. Shin, D. Yang, and C. Kim, "A channel rendezvous scheme for cognitive radio networks," *IEEE Communications Letter*, vol. 14, no. 10, pp. 954-956, 2010.

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

[70] R. R. Weber, "Optimal symmetric rendezvous search on three locations," *Mathematics of Operations Research*, vol. 37, no. 1, pp. 111–122, 2012.

[71] S. H. Wu, C. C. Wu, W. K. Hon, and K. G. Shin, "Rendezvous for heterogeneous spectrum-agile devices," In *Proc. IEEE INFOCOM*, 2014.

[72] T.-Y. Wu, W. Liao, and C.-S. Chang, "CACH: cycle-adjustable channel hopping for control channel establishment in cognitive radio networks," in *Proc. IEEE INFOCOM 2014.*

[73] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *System & Control Letters*, vol. 53, pp. 65–78, 2004.

[74] D. Yang, J. Shin, and C. Kim, "Deterministic rendezvous scheme in multichannel access networks," *Electronics Letters*, vol. 46, no. 20, pp. 1402-1404, 2010.

[75] L. Yu, H. Liu, Y.-W. Leung, X. Chu, and Z. Lin, "Multiple radios for effective rendezvous in cognitive radio networks," *IEEE International Conference on Communications (ICC)*, pp. 2857–2862, 2013.

[76] L. Yu, H. Liu, Y.-W. Leung, X. Chu, and Z. Lin, "Channel-hopping based on available channel set for rendezvous of cognitive radios." *IEEE International Conference on Communications (ICC)*, pp. 1573–1579, 2014.

[77] X. Zhang and H. Su, "CREAM-MAC: cognitive radio-enabled multi-channel MAC protocol over dynamic spectrum access networks," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 110-123, February 2011.

[78] Y. Zhang, Q. Li, G. Yu and B. Wang, "ETCH: efficient channel hopping for communication rendezvous in dynamic spectrum access networks," in *Proc. IEEE INFOCOM 2011.*

[79] J. Zhao, H. Zheng and G.-H. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *Proc. IEEE DySPAN'05.*