Tight Lower Bounds for Channel Hopping Schemes in Cognitive Radio Networks

Cheng-Shang Chang, *Fellow, IEEE,* Wanjiun Liao, *Fellow, IEEE,* and Tsung-Ying Wu

***Abstract*—In this paper, we consider the two-user multi-channel rendezvous problem in a cognitive radio network (CRN) and derive tight lower bounds for maximum time-to-rendezvous (MTTR) and maximum conditional time-to-rendezvous (MCTTR) of various channel hopping (CH) schemes under a channel loading constraint. In the symmetric and synchronous setting, we propose a novel Cycle Adjustable Channel Hopping (CACH) scheme to achieve the MTTR lower bound (when the channel loading is bounded above by $1/u$ with u being a prime power). Thus, the MTTR lower bound is tight and the CACH scheme is optimal in minimizing MTTR among all the symmetric and synchronous CH schemes under the same channel loading constraint. In the asymmetric setting, we show that the classical wait-for-mommy strategy can be used to achieve the MCTTR lower bound and thus it is optimal. In the symmetric and asynchronous setting, we also show a hierarchical construction of an asynchronous CH sequence by using two smaller asynchronous CH sequences. To further understand the effect of channel loading to the other performance metrics in a CRN, we perform various computer simulations for various CH schemes. Our simulation results show that the average time-to-rendezvous of CACH is independent of the total number of channels and it is also robust to the disturbance of primary users.**

***Index Terms*—rendezvous search, channel hopping, Galois field, cognitive radio networks.**

I. Introduction

WIRELESS networks used today are regulated by a fixed spectrum policy. This policy leads to the problem of inefficient usage of radio spectrum [2]. To solve this problem, cognitive radio (CR) [3] was introduced to improve the spectrum efficiency. In a cognitive radio network (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.

C. S. Chang is with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C. email: cschang@ee.nthu.edu.tw.

W. Liao and T.-Y. Wu are with Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. email: {wjliao,d96921024}@ntu.edu.tw.

Part of this paper was presented in [1] at the IEEE 33rd Annual Conference on Computer Communications (INFOCOM'14), Toronto, Canada, April 27–May 2, 2014.

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.

The most typical approach for control channel establishment is to use a dedicated global control channel among all SUs [4]–[7]. 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., [13], [14]). 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.

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., [8]–[13], [15]–[22]). In a CH scheme, time is usually divided into consecutive time intervals and each SU hops to a channel in every time interval according to a specific CH sequence. Eventually, two SUs rendezvous when they both hop to a common unblocked channel. As discussed in [22], CH schemes can be classified into various categories depending on their assumptions. 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 (such as ACH in [21] and ARCH in [22]), the sender and the receiver can use different strategies to rendezvous. On the other hand, both SUs in a *symmetric* CH scheme (such as SSCH in [8], SYN-MAC in [11], QCH in [12] and DH-MAC in [13]) have to follow the same strategy. As such, the performance of asymmetric CH schemes is better than that

of symmetric CH schemes. We note that the definition of *symmetry* might be used differently in various papers in this area. For instance, there are some papers in the literature (see e.g., [23], [24]) that define *asymmetric* users as users who have different available channel sets. As the available channel set to each user is only a subset of all the channels, such information can then be used for speeding up the rendezvous process [14], [24], [25]. In this paper, we do not assume that the available channel set to each user is fixed and known to each user. Also, a CH scheme is *synchronous* if the indices of time intervals 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. There are also several novel symmetric and asynchronous CH schemes that have been proposed in the literature, e.g., SeqR [9], CRSEQ [15], DRSEQ [16], ASYNCH-ETCH [18] and JS [19]. A comparison of all these CH schemes can be found in Table 1 of [22].

As addressed in [12], [22], there are four common metrics for evaluating the performance of a CH scheme: (i) degree of overlapping: the number of distinct channels for two SUs to rendezvous in each operation period, (ii) maximum time-to-rendezvous (MTTR): the maximum time for two SUs to rendezvous when there are no blocked channels, (iii) maximum conditional time-to-rendezvous (MCTTR): the maximum time for two SUs to rendezvous when there is at least one fixed unblocked channel in a CRN with N channels, and (iv) channel loading: the maximum probability that an SU hops to a particular channel at a particular time interval. Clearly, for the PU long-time blocking problem, a CH scheme should have a large degree of overlapping, preferable the maximum degree of overlapping. On the other hand, to reduce packet delay, it is preferable to have a low MTTR. Finally, to mitigate the control channel saturation problem, a CH scheme should control its channel loading so that the average number of SUs that hop to the same channel at the same time interval do not saturate the channel. As pointed out in [12], there is a tradeoff between time-to-rendezvous (TTR) and channel loading. In general, one can increase channel loading to reduce TTR and such a tradeoff can then be used to optimize system performance. Unfortunately, most existing CH schemes [8], [11], [13], [15]–[22] in the literature were designed for a fixed environment and they cannot be easily adjusted to optimize system performance.

Despite all the efforts in finding novel CH schemes, it is still not clear whether the performance of these CH schemes can be further improved. In our recent work [26], we made a first attempt to prove the optimality of various CH schemes under the *uniform* (equal) channel loading constraint. For this, we formulated a new type of rendezvous search problem, called the *multichannel rendezvous problem*, that is different from the classical rendezvous search problem (see e.g., the book [27] and references therein) in the additional channel loading constraint that puts a limit on the probability for a user to search a certain channel (or location).

In the multichannel rendezvous problem, there are N parallel channels (with $N \geq 2$), indexed from 0 to $N-1$. Also, time is slotted into time intervals, indexed from $t = 0, 1, 2, \ldots$. Two users (SUs in a CRN) who would like to rendezvous on a common unblocked channel generate their own random CH sequences *independently* and hop over these N channels with respect to time according to their CH sequences. Denote by $c_1(t)$ (resp. $c_2(t)$) 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, denoted by T, is the first time that these two users select a common unblocked channel, i.e.,

$$T = \inf\{t \geq 0 : c_1(t) = c_2(t) \notin B(t)\}. \tag{1}$$

As each user generates its CH sequence independently, a CH scheme in the multichannel problem satisfies the following independence assumption:

(A1) (Independence assumption) The two CH sequences $\{c_i(t), t \geq 0\}$, $i = 1$ and 2 are *independent*. Moreover, they are also independent of the channel blocking process $\{B(t), t \geq 0\}$.

The independence assumption of the two CH sequences seems reasonable in a practical scenario as the two users are not able to communicate with each other before they meet each other. The independence assumption of the channel blocking process is also reasonable when the behavior of primary users is not predicable. However, when the behavior of primary users is predicable or known in advance, then one might exploit the information of the channel blocking process to improve the performance. We note that there are some recent papers (see e.g., [14], [24], [25]) that use the information of the *available channel set* to each user to speed up the rendezvous process.

In [26], we added the following uniform channel loading constraint in the multichannel rendezvous problem:

(A2u) (Uniform channel loading assumption) At any time t, each channel is selected with an *equal* probability, i.e., for all $i = 1$ and 2, $j = 0, 1, 2, \ldots, N-1$, and $t \geq 0$,

$$\mathsf{P}(c_i(t) = j) = 1/N. \tag{2}$$

In this paper, we take one step further by considering the tradeoff between channel loading and the other performance metrics, such as MTTR and MCTTR. We ask the question how much performance gain we can have for MTTR and MCTTR if we are allowed to increase channel loading. For this, we consider the *general* channel loading assumption below:

(A2) (Channel loading assumption) The maximum probability that a user hops to a particular channel at a particular interval is not greater than $1/u$, i.e., for all $i = 1$ and 2, $j = 0, 1, \ldots, N-1$ and $t \geq 0$,

$$\mathsf{P}(c_i(t) = j) \leq 1/u. \tag{3}$$

Note that the definitions of channel loading in [12], [22] are equivalent to ours if the CH sequences in the their CH systems are chosen according to a probability distribution. Thus, it is more general to consider *random* CH sequences and define channel loading in terms of the maximum probability that a user hops to a particular channel at a particular interval. By doing so, the channel loading of the blind rendezvous scheme

that uses independent coin-tossing random sequences (see e.g., [17]) can also be defined.

In this paper, we will extend the results in [26] by deriving tight lower bounds for the MTTR and MCTTR of various CH schemes in the synchronous setting under the channel loading assumption in (A2). In the symmetric and asynchronous setting, we will also address another unsolved question in [26] by deriving a tight lower bound for the period of an asynchronous CH sequence with maximum degree of overlapping. Our main *theoretical* contributions include the following results:
(i) Synchronous CH schemes with channel loading constraints: In Section II, we consider *synchronous* CH schemes with channel loading constraints. Under the constraint that channel loading cannot exceed $1/u$ for some integer $2 \leq u \leq N$, we show in Theorem 2 that the MTTR of a symmetric CH scheme cannot be smaller than $u+1$. In Algorithm 2, we then propose a Cycle Adjustable Channel Hopping (CACH) scheme and show that the CACH scheme achieves the lower bound when u is a prime power. Thus, such a lower bound is tight when u is a prime power, and the CACH scheme is also optimal in minimizing MTTR among all the symmetric and synchronous CH schemes under the same channel loading constraint. In Theorem 16, we further show that the MCTTR of a CH scheme cannot be smaller than uN. The MCTTR of CACH is $(u+1)N$ and it is slightly larger than the lower bound. In Algorithm 3, we further construct an *asymmetric* CH scheme, called the *synchronous wait-for-mommy* strategy, that achieves the lower bound for MCTTR in Theorem 16.
(ii) Asynchronous CH schemes: In Section III, we consider *asynchronous* CH sequences that guarantee rendezvous even when there are $N-1$ blocked channels. For the asymmetric setting, we construct an asynchronous CH scheme, called the *asynchronous wait-for-mommy* strategy in Algorithm 4, that achieves the lower bound for MCTTR in Theorem 16. The MTTR of such a CH scheme is N, which is smaller than the MTTR of ACH [21] and the MTTR of ARCH [22]. For the symmetric setting, we show in Theorem 18 that the period of an asynchronous CH sequence with maximum degree of overlapping cannot be smaller than N^2+N+1. Such a lower bound is not only better than the N^2 lower bound in [15], [21] but also tight for $N=2$ and $N=8$. In particular, the periodic CH sequence with 8 channels and 73 time slots in a period in [20] achieves the lower bound. We also show in Theorem 19 a hierarchical construction of an asynchronous CH sequence by using two smaller asynchronous CH sequences. For $N=64$, such a construction enables us to find an asynchronous CH sequence that has a smaller MCTTR than CRSEQ [15].

In Table I, we summarize the lower bounds for MTTR and MCTTR in the four settings, (Symmetric, Synchronous), (Asymmetric, Synchronous), (Asymmetric, Asynchronous), and (Symmetric, Asynchronous). In the last column of the table, we show the optimal CH schemes that achieve these lower bounds. In comparison with the state-of-the-art CH schemes in Table 1 of [22], we note that in the symmetric and synchronous setting CACH has a smaller MTTR than L-QCH [12] under the same channel loading. By choosing $u=N$, the channel loading of CACH is smaller than that of SSCH [8] under the same MTTR. In the asymmetric and synchronous setting, the synchronous wait-for-mommy strategy and RCCH in [22] have the same degree of overlapping, channel loading, MTTR and MCTTR when the channel loading is set to $2/N$. However, the synchronous wait-for-mommy strategy is more flexible than RCCH as the synchronous wait-for-mommy strategy can also be operated under any other channel loading $1/u$ with u being a positive integer. In the asymmetric and asynchronous setting, the asynchronous wait-for-mommy strategy, ACH in [21] and ARCH [22] all have the same degree of overlapping, channel loading and MCTTR. However, the MTTR of the asynchronous wait-for-mommy strategy is N, which is smaller than N^2-N+1 in ACH and $2N-1$ in ARCH. According to Table 1 of [22], the period of CRSEQ [15] is $N(3N-1)$ (with N being a prime), which is the best result in the literature for an asynchronous CH sequence with maximum degree of overlapping in the symmetric and asynchronous setting. The period of CRSEQ is still much larger than our lower bound N^2+N+1 and that suggests there might be still room for improvement in this setting.

TABLE I: Summary of the lower bounds and the optimal CH schemes that achieve the lower bounds

	channel loading	MTTR	MCTTR or period	Optimal CH scheme
Sym. Syn.	$1/u$	$u+1$ (Theorem 2)	n/a	CACH[(i)] (Algorithm 2)
Asym. Syn.	$1/u$	u	uN (Theorem 16)	Syn. w-f-m[(ii)] (Algorithm 3)
Asym. Asyn.	$1/N$	N	N^2 (Theorem 16)	Asyn. w-f-m (Algorithm 4)
Sym. Asyn.	n/a	n/a	N^2+N+1[(iii)] (Theorem 18)	$(8,73)$-MACH[(iv)] [20]

(i): u has to be a prime power
(ii): w-f-m stands for wait-for-mommy
(iii): the minimum period of an asynchronous CH sequence with maximum degree of overlapping
(iv): The lower bound is achieved for $N=8$

In addition to the theoretical analysis of the CH schemes, we also perform various computer simulations in Section IV to further understand the effect of channel loading to the other performance metrics in a CRN. For this, we compare SYNMAC [11], SSCH [8], L-QCH [12], RRICH [1], CACH [1] EJSCH [23], AHWCH [34] and ETCH [18]. Our simulation results show that the average time-to-rendezvous of CACH is independent of the total number of channels and it is also robust to the disturbance of primary users.

II. Synchronous channel hopping schemes with channel loading constraints

In this section, we consider *synchronous* channel hopping schemes in the multichannel rendezvous problem.

Definition 1: If a CH scheme satisfies the independence assumption in (A1) and the channel loading assumption in (A2), it is called a CH scheme with channel loading not greater than $1/u$. Furthermore, a CH scheme is *symmetric* if it also satisfies the following assumption:

(A3) (Symmetric assumption) The CH sequences $\{c_i(t), t \geq 0\}$, $i=1$ and 2, follow the same joint distribution.

One way for (A3) to hold is for both SUs to generate their CH sequences by using a common (random) algorithm independent of the channel blocking process $\{B(t), t \geq 0\}$.

A. A lower bound for MTTR

In this section, we show a tight lower bound for the MTTR of a symmetric CH scheme with channel loading not greater than $1/u$.

Theorem 2: For a *symmetric* CH scheme with channel loading not greater than $1/u$, if u is an integer and $2 \leq u \leq N$, then its MTTR cannot be smaller than $u+1$.

In Theorem 2 of [26], it was showed that the MTTR of a symmetric CH scheme with channel loading $1/N$ is not smaller than $N+1$. By taking $u = N$, it is easy to see that the lower bound in Theorem 2 is a generalization of that. We will show in Section II-C that the Cycle Adjustable Channel Hopping scheme (CACH) achieves the lower bound when u is a prime power. Thus, the lower bound is tight when u is a prime power.

The rest of this section is devoted to the proof of Theorem 2. Our proof relies heavily on Lemma 3 and Lemma 4 that were previously proved in [26], Lemma 1 and Lemma 3, respectively. The generalized union bound in Lemma 3 is tighter than the usual union bound as it subtracts a sum of additional nonzero probabilities.

Lemma 3: (Generalized union bound, [26], Lemma 1) For any $n \geq 2$ events, $E_1, E_2, \ldots, E_n$,

$$\mathsf{P}(\cup_{i=1}^{n} E_i) \leq \sum_{i=1}^{n} \mathsf{P}(E_i) - \sum_{i=1}^{n-1} \mathsf{P}(E_i \cap E_{i+1}). \tag{4}$$

Lemma 4 shows that the event for the two *symmetric* SUs to rendezvous at two different time slots has a nonzero probability. Note that such an event is undesirable as it wastes an addition time slot. This is also the key difference between the symmetric setting and the asymmetric setting. In the asymmetric setting, the probability of such an event could be zero, e.g., the wait-for-mommy strategy [29].

Lemma 4: ([26], Lemma 3) Consider two CH sequences $\{c_1(t), t \geq 0\}$ and $\{c_2(t), t \geq 0\}$ over N channels. If these two sequences satisfy (A1) and (A3), then

$$\mathsf{P}(c_1(s) = c_2(s),\ c_1(t) = c_2(t)) \geq \frac{1}{N^2}, \quad s \neq t. \tag{5}$$

In the following proposition, we show a result for a constrained optimization problem. The result of such an optimization problem will be used in Lemma 6 for bounding the probability that the two SUs rendezvous at a particular time slot under the channel loading constraint.

Proposition 5: Suppose that u is an integer and $1 \leq u \leq N$. Consider the following constrained optimization problem:

$$\begin{aligned} \max \quad & \sum_{j=0}^{N-1} p_j q_j \\ s.t. \quad & 0 \leq p_j \leq 1/u, \quad 0 \leq q_j \leq 1/u, \\ & \sum_{j=0}^{N-1} p_j = 1, \quad \sum_{j=0}^{N-1} q_j = 1. \end{aligned}$$

Then its maximum value is $1/u$.

Proof. Note that

$$p_j q_j \leq \frac{p_j^2 + q_j^2}{2}.$$

From the majorization theory [30], the maximum of $\sum_{j=0}^{N-1} p_j^2/2$ under the same constraint is achieved when $p_j = 1/u$, $j = 0, 1, \ldots, u-1$ and $p_j = 0$ for $j = u, u+1, \ldots, N-1$. Thus, $\sum_{j=0}^{N-1} p_j^2/2 \leq 1/(2u)$. Similarly, we also have $\sum_{j=0}^{N-1} q_j^2/2 \leq 1/(2u)$. Thus, the maximum value of the optimization problem is bounded above $1/u$.

On the other hand, if we choose $p_j = q_j = 1/u$, $j = 0, 1, \ldots, u-1$ and $p_j = q_j = 0$ for $j = u, u+1, \ldots, N-1$, then the objective value of the optimization problem is $1/u$. Such a choice achieves the maximum value of the optimization problem. ■

Lemma 6: Consider two CH sequences $\{c_1(t), t \geq 0\}$ and $\{c_2(t), t \geq 0\}$ over N channels. If these two sequences satisfy (A1) and (A2) with u being an integer and $1 \leq u \leq N$, then

$$\mathsf{P}(c_1(t) = c_2(t)) \leq 1/u. \tag{6}$$

Proof. Since we assume in (A1) that these two sequences are independent of each other, we have

$$\begin{aligned} & \mathsf{P}(c_1(t) = c_2(t)) \\ &= \sum_{j=0}^{N-1} \mathsf{P}(c_1(t) = j,\ c_2(t) = j) \\ &= \sum_{j=0}^{N-1} \mathsf{P}(c_1(t) = j) \cdot \mathsf{P}(c_2(t) = j). \end{aligned} \tag{7}$$

From (A2), we know that for $i = 1$ and 2,

$$0 \leq \mathsf{P}(c_i(t) = j) \leq 1/u.$$

As a direct consequence of Proposition 5 of (7), we then have

$$\mathsf{P}(c_1(t) = c_2(t)) \leq 1/u.$$

■

Proof. (Theorem 2) Consider the event $\{T \geq u\}$ that these two CH sequences do not rendezvous before time u when there are no blocked channels. We will show that $\mathsf{P}(T \geq u) > 0$ so that MTTR cannot be smaller than $u+1$. Note that

$$\begin{aligned} & \mathsf{P}(T \geq u) = \mathsf{P}(c_1(s) \neq c_2(s), 0 \leq s \leq u-1) \\ &= 1 - \mathsf{P}(\cup_{s=0}^{u-1} \{c_1(s) = c_2(s)\}). \end{aligned}$$

Using the generalized union bound in Lemma 3, we have for $u \geq 2$,

$$\begin{aligned} & \mathsf{P}(\cup_{s=0}^{u-1} \{c_1(s) = c_2(s)\}) \\ &\leq \sum_{s=0}^{u-1} \mathsf{P}(c_1(s) = c_2(s)) \\ &\quad - \sum_{s=0}^{u-2} \mathsf{P}(c_1(s) = c_2(s), c_1(s+1) = c_2(s+1)). \end{aligned}$$

It then follows from (5) in Lemma 4 and (6) in Lemma 6 that

$$\mathsf{P}(\cup_{s=0}^{u-1}\{c_1(s)=c_2(s)\}) \leq 1-\frac{u-1}{N^2}.$$

Thus, for $u \geq 2$

$$\mathsf{P}(T \geq u) = 1-\mathsf{P}(\cup_{s=0}^{u-1}\{c_1(s)=c_2(s)\}) \geq \frac{u-1}{N^2} > 0.$$

■

B. The round-robin indemnity channel hopping scheme

In this section, we introduce the construction of the Round-Robin Indemnity Channel Hopping (RRICH) sequence. Such a construction will be used in our simulation for performance comparison in Section IV. It will also be used in the next section to construct the Cycle Adjustable Channel Hopping (CACH) scheme that achieves the MTTR lower bound in Theorem 2.

Our construction for the RRICH sequence is based on the mathematical theory of Galois fields [28]. 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. 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.

In RRICH, 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 $GF(N)$.

For each SU i, its RRICH sequence is a periodic sequence with period $N(N+1)$. Each period of $N(N+1)$ time intervals is called a frame. The sequence for SU i in a frame is determined by using a set of two CH parameters: $\{x_i, h_i\}$ (called the RRICH parameter set). The parameter x_i is called the initial seed. It denotes the initial channel of the RRICH sequence and its value is an integer ranged over $[0, N-1]$. The other parameter h_i is called the hopping seed. It is used to determine which channel for SU i to switch to. In order for the SU to change its control channel over time, we will not select the zero element as a hopping seed (we note that such a constraint will be removed in CACH). Hence, the value of a hopping seed is an integer ranged over $[1, N-1]$. Specifically, let $c_i(t)$ be the control channel of SU i used at the t^{th} time interval for $0 \leq t \leq N(N+1)-1$. Suppose that $t=q(N+1)+r$, where q is the quotient of t divided by $N+1$ and r is its remainder. Then $c_i(t)$ is determined by:

$$c_i(t) = \begin{cases} h_i \oplus q, & \text{if } r = N \\ (x_i \oplus q) \oplus (h_i \otimes r), & \text{if } 0 \leq r \leq N-1 \end{cases}. \quad (8)$$

To see the intuition behind (8), suppose that $0 \leq t \leq N-1$. For this case, we have $q=0$. The above equation is simply a "line" in the field $GF(N)$ with h_i being its "slope." As h_i is a nonzero element, the line is not a constant. As such, every SU hops to different channels as time goes on. Moreover, for two SUs with different hopping seeds, they hop as two lines with different "slopes" and these two lines intersect each other at a unique point. For two SUs with identical hopping seeds, they hop as two "parallel lines." That is why we have to add indemnity time intervals (the time intervals with $r=N$) for them to rendezvous.

For the ease of our presentation, we partition each frame of $N(N+1)$ time intervals into N sub-frames, each with $N+1$ time intervals. Specifically, the q^{th} sub-frame, $q=0,1,\ldots,N-1$, contains the time intervals from $q(N+1)$ to $q(N+1)+N$. Call the channel that an SU uses at an indemnity interval the indemnity channel. Note that the initial seed and the indemnity channel of an SU are updated by "adding" q for every sub-frame while the hopping seed remains unchanged for every sub-frame. As such, the channel selections for every sub-frame behave exactly the same as those in the first sub-frame by re-indexing the channels through a "rotation" under the $\oplus$ operation. That is why we call such a sequence the round-robin indemnity channel hopping scheme.

In the following lemma, we show that an SU can scan all the channels and check the availability of each channel.

Lemma 7: An SU visits all the N channels within the first N time intervals in each sub-frame.

Proof. See the proof of Lemma 1 in [1]. ■

The following lemma shows when two SUs rendezvous in each sub-frame.

Lemma 8: . Consider two SUs with the parameter sets $\{x_1, h_1\}$ and $\{x_2, h_2\}$.

(i) If they are assigned with the same hopping seed and the same initial seed, i.e., $x_1 = x_2$ and $h_1 = h_2$, then they will rendezvous at each time interval.

(ii) If they are assigned with the same hopping seed ($h_1 = h_2$), but with different initial seeds x_1 and x_2 ($x_1 \neq x_2$), they will rendezvous at the indemnity time intervals (the last time interval in each sub-frame), i.e., $t=q(N+1)+N$, $q=0,1,2,\ldots,N-1$.

(iii) If they are assigned with different hopping seeds ($h_1 \neq h_2$), they will rendezvous at the r^{th} time interval in each sub-frame, i.e., $t = q(N+1)+r$, $q = 0, 1, 2, \ldots, N-1$, where

$$r = (h_2 \oplus (-h_1))^{-1} \otimes (x_1 \oplus (-x_2)). \quad (9)$$

Proof. See the proof of Lemma 2 in [1]. ■

In the following theorem, we show that RRICH achieves the maximum degree of overlapping if the two SUs do not have the same parameter set. Together with Lemma 7, RRICH achieves the maximum degree of overlapping, i.e., all the channels can be used as rendezvous channels for any two SUs in an operation period.

Theorem 9: In RRICH, any two SUs will rendezvous at least once in each sub-frame. Moreover, the channels they rendezvous in the N sub-frames are *distinct* if these two SUs do not have the same parameter set.

Proof. See the proof of Theorem 3 in [1]. ■

As a direct consequence of Theorem 9, we show that RRICH also solves the PU long-time blocking problem.

Corollary 10: . Suppose that there are only m ($m < N$) fixed channels that are used by PUs. Any two SUs will rendezvous within $(m+1)(N+1)$ time intervals.

Proof. See the proof of Corollary 4 in [1]. ■

Algorithm 1: (Round-robin indemnity channel hopping scheme) Each SU chooses its initial seed x independently and uniformly over $[0, N-1]$ and its hopping seed h independently and uniformly over $[1, N-1]$. Construct its CH sequence $\{c_{x,h}(t), t \geq 0\}$ according to (8).

Corollary 11: The RRICH scheme is a synchronous CH scheme with channel loading $1/(N-1)$. Its MTTR is $N+1$ and its MCTTR is $N(N+1)$.

Proof. Clearly the independent assumption in (A1) and the symmetric assumption in (A3) are satisfied trivially. Now we show that the channel loading of RRICH is $1/(N-1)$. If $t = q(N+1)+N$, then $c_{x,h}(t) = (h \oplus q)$ and an SU that selects such a channel must select the hopping seed h. As the hopping seed is selected uniformly over $[1, N-1]$, the probability that an SU selects such a channel is then $1/(N-1)$. On the other hand, if $t = (q+1)N + r$ for some $0 \leq r \leq N-1$, then for every fixed h and every fixed channel c there is a unique x such that $c = (x \oplus q) \oplus (h \otimes r)$. Thus, the probability that an SU will select channel c is the probability that the SU selects the exact x that solves the equation. Such a probability is $1/N$. For both cases, we conclude that the channel loading of RRICH is $1/(N-1)$.

From Corollary 10, we also know that the its MTTR is $N+1$ (when $m = 0$) and its MCTTR is $N(N+1)$ (when $m = N-1$). ■

We note that RRICH and SSCH (see in Table 1 of [22]) has the same channel loading and the same MTTR. In fact, RRICH is a generalization of SSCH in two folds: (i) RRICH uses the field operations which are much more general than the prime number modular arithmetic in SSCH, and (ii) RRICH implements "rotation" under the $\oplus$ operation and thus the degree of overlapping is N. We also note that the idea of using "rotation" was previous used in [13].

C. The cycle adjustable channel hopping scheme

One problem of the RRICH scheme is that the time-to-rendezvous (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, we introduce the Cycle-Adjustable Channel Hopping (CACH) scheme in this section. We will show that the CACH scheme achieves the MTTR lower bound in Theorem 2.

The key idea of CACH is to create another layer of logical channels and have SUs rendezvous on logical channels. By choosing a modulo operation between logical channels and physical channels, CACH still achieves the maximum degree of overlapping as RRICH and thus it can still be used for solving the PU long-time blocking problem.

To reduce the TTR, we choose a much smaller prime power u for the construction of the first sub-frame in RRICH and have two SUs rendezvous on one of the u logical channels in the first $u+1$ time intervals. As in the construction of RRICH, we find a Galois field $GF(u)$ with the two operations $\oplus$ and $\otimes$. Then SU i chooses its parameter set $\{x_i, h_i\}$, where x_i is the initial seed and h_i is the hopping seed. However, unlike RRICH, both the initial seed and the hopping seed are chosen in $[0, u-1]$. In other words, the hopping seed can be the zero element in the $GF(u)$ used here. Each CACH sequence is a periodic sequence with period $(u+1)N$. For $0 \leq t \leq (u+1)N-1$, we further partition this period of $(u+1)N$ time intervals into N sub-frames, each with $u+1$ time intervals. The last interval in a sub-frame is called the indemnity time interval. Let $\ell_i(t)$ and $c_i(t)$ be the logical channel and the physical channel used by SU i at the t^{th} time interval. Suppose that $t = q(u+1)+r$, where q is the quotient of t divided by $(u+1)$ and r is its remainder. Then $\ell_i(t)$ and $c_i(t)$ are determined by the following equation:

$$\ell_i(t) = \begin{cases} h_i, & \text{if } r = u \\ x_i \oplus (h_i \otimes r), & \text{if } 0 \leq r \leq u-1 \end{cases}, \quad (10)$$

$$c_i(t) = (\ell_i(t) + q) \bmod N. \quad (11)$$

The construction of the sequence $\{\ell_i(t), t \geq 0\}$ is the same as that in (8) except we remove the effect of q. Thus, the sequence $\{\ell_i(t), t \geq 0\}$ is a periodic sequence with period $u+1$ and it repeats itself in every sub-frame. The index q is used in the mapping from a logical channel to a physical channel through the modulo operation in (11). As such, the physical channels used in each sub-frame are different.

In Figure 1, we give an example for the construction of CACH sequences for N=5 and u=3. In this example, the addition in $GF(3)$ is the usual addition with the MOD 3 operation and the multiplication in $GF(3)$ is the usual multiplication with the MOD 3 operation. Since u=3, each

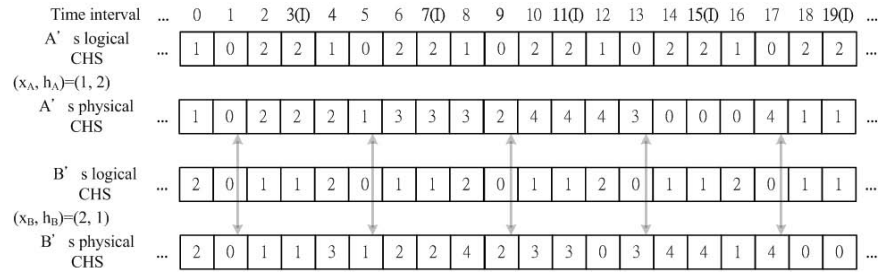


Fig. 1: CACH sequences for two SUs with 5 physical channels and 3 logical channels (i.e., N =5 and u=3).

sub-frame contains four time intervals with the last time interval in each sub-frame being the indemnity interval. For SU A with parameter set $(x_A, h_A) = (1, 2)$, we have $\ell_A(0) = x_A = 1$, $\ell_A(1) = (x_A + h_A) \bmod 3 = 0$, and $\ell_A(2) = (x_A + 2h_A) \bmod 3 = 2$. As the last time interval in each sub-frame is the indemnity interval, $\ell_A(3) = h_A = 2$. Note that the logical channel hopping sequence $\ell_A(t)$ repeats itself in each sub-frame with the sequence 1, 0, 2, 2. The physical channel hopping sequence $c_A(t)$ then adds 1 with the MOD 5 operation to 1, 0, 2, 2 in each sub-frame and that leads to 1, 0, 2, 2 for the 0^{th} sub-frame, 2, 1, 3, 3 for the 1^{st} sub-frame, 3, 2, 4, 4 for the 2^{nd} sub-frame, 4, 3, 0, 0 for the 3^{rd} sub-frame and 0, 4, 1, 1 for the 4^{th} sub-frame. Both the logical channel hopping sequence and the physical channel hopping sequence for SU B with the parameter set $(x_B, h_B) = (2, 1)$ are also shown in Figure 1.

Following the same argument as in the proof of Lemma 8, we have the following lemma for CACH. Note that this lemma still holds even though we allow the hopping seed to be the zero element.

Lemma 12: Consider two SUs with the parameter sets $\{x_1, h_1\}$ and $\{x_2, h_2\}$.

(i) If they are assigned with the same hopping seed and the same initial seed, i.e., $x_1 = x_2$ and $h_1 = h_2$, then they will rendezvous at each time interval.

(ii) If they are assigned with the same hopping seed ($h_1 = h_2$), but with different initial seeds x_1 and x_2 ($x_1 \neq x_2$), they will rendezvous at the indemnity time intervals (the last time interval in each sub-frame), i.e., , $t = q(u+1)+u$, $q = 0, 1, 2, \ldots, N-1$.

(iii) If they are assigned with different hopping seeds ($h_1 \neq h_2$), they will rendezvous at the r^{th} time interval in each sub-frame, i.e., $t = q(u + 1) + r$, $q = 0, 1, 2, \ldots, N - 1$, where

$$r = (h_2 \oplus (-h_1))^{-1} \otimes (x_1 \oplus (-x_2)). \qquad (12)$$

In Theorem 13, we show that CACH also achieves the maximum degree of overlapping as RRICH.

Theorem 13: Any two SUs will rendezvous at least once in each sub-frame. Moreover, the physical channels they rendezvous in the first m sub-frames contain at least m *distinct* channels, $m = 1, 2, \ldots, N$.

Proof. See the proof of Theorem 6 in [1]. ■

Analogous to proof for Corollary 10, one can use the results in Theorem 13 to show that CACH also solves the PU long-time blocking problem.

Corollary 14: Suppose that there are m ($m < N$) fixed channels that are used by PUs. Any two SUs will rendezvous within $(m + 1)(u + 1)$ time intervals.

Algorithm 2: (Cycle adjustable channel hopping scheme) Each SU chooses its initial seed x independently and uniformly over $[0, u-1]$ and its hopping seed h independently and uniformly over $[0, u - 1]$. Construct its CH sequence $\{c_{x,h}(t), t \geq 0\}$ according to (11).

Corollary 15: The CACH scheme is a synchronous CH scheme with channel loading $1/u$. Its MTTR is $u + 1$ and its MCTTR is $N(u + 1)$.

Proof. Clearly the independent assumption in (A1) and the symmetric assumption in (A3) are satisfied trivially. To see that the channel loading of CACH is $1/u$, note that the probability that an SU is distributed in a logical channel (and the corresponding physical channel) is simply $1/u$.

From Corollary 14, we also know that the its MTTR is $u+1$ (when $m = 0$) and its MCTTR is $N(u+1)$ (when $m = N-1$). ■

In particular, if we choose $u = 2$, the MTTR for CACH is 3 and its channel loading is only 1/2, which is lower than 2/3 of M-QCH [12]. For this case, CACH is better than M-QCH as CACH has a lower channel loading while keeping the same degree of overlapping and the same MTTR. Now we compare CACH with L-QCH [12]. If the MTTR is τ, it is shown in Theorem 2 of [12] that the channel loading of any QCH system is at least $1/\sqrt{\tau}$. L-QCH is the QCH system with the channel loading $1/\sqrt{\tau}$. Taking $u = \tau - 1$, we then derive that the channel loading of the CACH scheme is only $1/(\tau - 1)$, which is significantly lower than $1/\sqrt{\tau}$ in L-QCH. In view of these, we conclude that CACH is in general much better than QCH in terms of reducing channel loading while keeping the same degree of overlapping and the same MTTR. In fact, in view of Corollary 15, CACH achieves the MTTR lower bound in Theorem 2 and thus is optimal in minimizing MTTR among all the symmetric and synchronous CH schemes with the same channel loading constraint.

Certainly, choosing the number of logical channels u that optimizes the system performance, e.g., throughput, depends on various system characteristics, e.g., the number of SUs, the number of channels, and the characteristics of PUs. For this, we will perform various computer simulations in Section

IV. One of our findings from the simulations results is to set the number of logical channels u close to the average number of neighbors for a reasonably good throughput. As the load of CACH is $1/u$, the average number of SUs that hop to a rendezvous channel is close to 1 if u is close to the average number of neighbors.

D. A lower bound for MCTTR

In this section, we show a tight lower bound for the MCTTR of a CH scheme with channel loading not greater than $1/u$.

Theorem 16: For a CH scheme with channel loading not greater than $1/u$, if u is an integer and $1 \leq u \leq N$, then its MCTTR cannot be smaller than uN.

Proof. Recall that MCTTR is defined as the maximum time for two SUs to rendezvous when there are $N-1$ blocked channels. Consider the scenario that every channel has an equal probability of being the only unblocked channel, i.e., with an equal probability $1/N$, channel z is the only unblocked channel for $z = 0, 1, 2, \ldots, N-1$. Consider the event $\{T \geq uN-1\}$ that these two CH sequences do not rendezvous before time $uN-1$. We will show that $\mathsf{P}(T \geq uN-1) > 0$ so that MCTTR cannot be smaller than uN. Note that $\{T < uN-1\}$ is the event that these two CH sequences rendezvous before time $uN-1$. Thus,

$$\begin{aligned}&\mathsf{P}(T < uN-1)\\ &= \frac{1}{N}\sum_{z=0}^{N-1} \mathsf{P}(\cup_{s=0}^{uN-2}\{c_1(s) = c_2(s) = z\}).\end{aligned}$$

Using the union bound yields

$$\begin{aligned}&\mathsf{P}(\cup_{s=0}^{uN-2}\{c_1(s) = c_2(s) = z\})\\ &\leq \sum_{s=0}^{uN-2} \mathsf{P}(c_1(s) = c_2(s) = z).\end{aligned}$$

Since these two sequences are independent, we then have

$$\mathsf{P}(T < uN-1) = \frac{1}{N}\sum_{s=0}^{uN-2}\sum_{z=0}^{N-1} \mathsf{P}(c_1(s) = z) \cdot \mathsf{P}(c_2(s) = z).$$

From the channel loading assumption in (A2) and Proposition 5, it then follows that

$$\mathsf{P}(T < uN-1) \leq \frac{1}{N}\sum_{s=0}^{uN-2}\frac{1}{u} = 1 - \frac{1}{uN} < 1.$$

Thus, $\mathsf{P}(T \geq uN-1) > 0$. ∎

From Corollary 15, the MCTTR of CACH is $(u+1)N$. This is slightly larger than the lower bound for MCTTR in Theorem 16. The reason is that CACH is a symmetric CH scheme while the derivation of the lower bound for MCTTR in Theorem 16 does not require the symmetric assumption in (A3).

Now we show how to construct an *asymmetric* CH scheme with channel loading not greater than $1/u$ to achieve the lower bound for MCTTR in Theorem 16. The idea is to use the wait-for-mommy strategy [29] and the concept of logical channels in CACH.

Algorithm 3: (Synchronous wait-for-mommy strategy) For any t, let $q(t) = \lfloor t/u \rfloor$ and $r(t) = (t \bmod u)$.

(Receiver/mommy) SU 1 generates independently a uniformly distributed random variable U_1 over $[0, u-1]$. Then construct its CH sequence as follows:

$$\ell_1(t) = (r(t) + U_1) \bmod u, \tag{13}$$

$$c_1(t) = (\ell_1(t) + q(t)) \bmod N. \tag{14}$$

(Sender/child) SU 2 also generates independently a uniformly distributed random variable U_2 over $[0, u-1]$. Then construct its CH sequence as follows:

$$\ell_2(t) = U_2, \tag{15}$$

$$c_2(t) = (\ell_2(t) + q(t)) \bmod N. \tag{16}$$

Clearly, both CH sequences are periodic with period uN. As in CACH, we may view every u time intervals as a sub-frame. Since U_1 and U_2 are uniformly distributed over $[0, u-1]$, it is easy to see that the channel loading for both SUs is $1/u$. Moreover, according to the wait-for-mommy strategy in (13) and (15), the receiver (the mommy) cycles through the u (logical) channels periodically and the sender (the child) stays at the same (logical) channel for u time slots. Thus, these two SUs rendezvous for every sub-frame. From the rotation operations in (14) and (16), the rendezvous channels for the first N frames are *distinct* and thus the MCTTR for such a CH scheme is uN.

III. Asynchronous channel hopping schemes

In this section, we consider asynchronous channel hopping schemes. In Section III-A, we consider the asymmetric setting. We then consider the symmetric setting in Section III-B and show a hierarchical construction in Section III-C.

A. The asymmetric setting

The asymmetric setting is relatively easy. Observe that the smallest channel loading is $1/N$. As a corollary of Theorem 16 (with $u = N$), we know that the MCTTR of an asynchronous CH scheme cannot be smaller than N^2. There are many known CH schemes that achieve such a lower bound, e.g., ACH [21] and ARCH [22]. In Algorithm 4 below, we propose the asynchronous wait-for-mommy strategy that also achieves such a lower bound.

Algorithm 4: (Asynchronous wait-for-mommy strategy) For any t, let $q(t) = \lfloor t/N \rfloor$ and $r(t) = (t \bmod N)$.

(Receiver/mommy) SU 1 generates independently a uniformly distributed random variable U_1 over $[0, N-1]$. Then construct its CH sequence as follows:

$$c_1(t) = (r(t) + U_1) \bmod N. \tag{17}$$

(Sender/child) SU 2 also generates independently a uniformly distributed random variable U_2 over $[0, N-1]$. Then construct its CH sequence as follows:

$$c_2(t) = (q(t) + U_2) \bmod N. \tag{18}$$

For such a CH scheme, the receiver (the mommy) cycles through the N channels periodically with period N and the

sender (the child) stays at the same channel for N time slots and then repeatedly hops to another unvisited channel for another N time slots. Clearly, the channel loading for such a CH scheme is $1/N$. Moreover, even in the asynchronous setting, it is easy to see that the MTTR and the MCTTR for such a CH scheme is N and N^2 respectively when the sender starts to look for its receiver. As indicated in Table 1 of [22], the MTTR of ACH [21] is $N^2 - N + 1$ and that of ARCH [22] is $2N - 1$. In comparison with these two CH schemes, the MTTR of the asynchronous wait-for-mommy strategy is much smaller.

B. A lower bound for the period of an asynchronous channel hopping sequence in the symmetric setting

Now we focus on the *symmetric* setting. Recall that a periodic CH sequence is said to achieve the maximum degree of overlapping for a system with N channels if two asynchronous SUs 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 degree of overlapping (MACH).

Definition 17: 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 degree of overlapping) 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 degree of overlapping property in (ii), we know if there is a time shift d between two SUs, then SU 1 with $c_1(t) = c(t)$ and SU 2 with $c_2(t) = c(t + d)$ 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., [15], [20], [21]). Clearly, for an (N, p)-MACH sequence, its MCTTR is bounded above by its period p.

For example, the periodic sequence 0010111... with period 7 in [20] is a $(2, 7)$-MACH sequence. By packing eight difference sets into 73 time slots, it was further shown in [20] that there is a $(8, 73)$-MACH sequence. If N is a prime, the CRSEQ scheme in [15] is an $(N, N(3N - 1))$-MACH sequence. Such a result is the best known result for MCTTR in the symmetric and asynchronous setting when the number of channels is large.

It was argued in [15] (by using Theorem 2 in [31]) and Theorem 1 of [21] that the period p of an (N, p)-MACH sequence cannot be smaller than N^2. In this section, we tighten the lower bound to $N^2 + N + 1$ in Theorem 18. Our lower bound is not only better but also *tight* for $N = 2$ and $N = 8$. In particular, the $(2, 7)$-MACH sequence in [20] and the $(8, 73)$-MACH sequence in [20] achieve the lower bound in Theorem 18. Thus, the $(2, 7)$-MACH sequence in [20] and the $(8, 73)$-MACH sequence in [20] are optimal in the sense of minimizing the period among all the MACH sequences with the same number of channels.

Theorem 18: For any (N, p)-MACH sequence with $N \geq 2$, its period p cannot be smaller than $N^2 + N + 1$.

Proof. Let n_i be the number of times in a period that channel i is selected. Clearly, we have

$$\sum_{i=0}^{N-1} n_i = p. \tag{19}$$

Suppose that channel i is selected at $0 \leq t_{i,1} < t_{i,2} < \ldots < t_{i,n_i} \leq p - 1$. Let

$$S_i = \{(t_{i,j}, t_{i,k}), j \neq k, j, k = 1, 2, \ldots, n_i\}.$$

Clearly, the size of the set S_i, i.e., the number of un-ordered pairs, is $n_i * (n_i - 1)$. For an (N, p)-MACH sequence, we know that for any time shift $0 \leq d \leq p - 1$, there is $\tau(i, d)$ such that

$$c(\tau(i, d)) = c((\tau(i, d) + d) \bmod p) = i. \tag{20}$$

For each $1 \leq d \leq p - 1$, the ordered pair

$$(\tau(i, d), (\tau(i, d) + d) \bmod p)$$

is in S_i. Thus, for all $i = 0, 1, \ldots, N - 1$,

$$p - 1 \leq |S_i| = n_i(n_i - 1). \tag{21}$$

Let $i^* = \text{argmin}[n_i]$ be the channel that are selected the smallest number of times in a period. From (19), we know that

$$Nn_{i^*} \leq \sum_{i=0}^{N-1} n_i = p. \tag{22}$$

Using this in (21) yields

$$Nn_{i^*} \leq n_{i^*}(n_{i^*} - 1) + 1. \tag{23}$$

This shows that for $N \geq 2$,

$$n_{i^*} \geq N + 1. \tag{24}$$

From (22), it then follows that

$$p \geq N(N + 1).$$

It remains to show that $p \neq N(N + 1)$. We will prove this by contradiction. Suppose that $p = N(N + 1)$. In this case, we know from (24) and (19) that $n_i = N + 1$ for all $i = 0, 1, \ldots, N - 1$. Let

$$\begin{aligned} \tilde{S}_i \quad = \quad & \{d : d = (t_{i,j} - t_{i,k}) \bmod p, \\ & \text{for some } j \neq k, j, k = 1, 2, \ldots, N + 1\}. \end{aligned} \tag{25}$$

From the modulo operation and $j \neq k$, we know that $\tilde{S}_i \subset \{1, \ldots, p-1\}$ and thus $|\tilde{S}_i| \leq p-1$. On the other hand, from the property of maximum degree of overlapping, we also know that $d \in \tilde{S}_i$ for each $1 \leq d \leq p-1$. Thus, $\tilde{S}_i = \{1, \ldots, p-1\}$ and $|\tilde{S}_i| = p - 1$. Since $p = N(N + 1)$, it then follows that

$$|\tilde{S}_i| = N(N + 1) - 1. \tag{26}$$

As there are $N(N+1)$ ordered pairs $(t_{i,j}, t_{i,k})$ for $j \neq k$, there must exist some $1 \leq d' \leq N(N + 1) - 1$ and two ordered

pairs (t_{i,j_1}, t_{i,k_1}) and (t_{i,j_2}, t_{i,k_2}) with $j_1 \neq j_2$, $k_1 \neq k_2$ such that

$$\begin{aligned} d' &= (t_{i,j_1} - t_{i,k_1}) \bmod N(N+1) \\ &= (t_{i,j_2} - t_{i,k_2}) \bmod N(N+1). \end{aligned} \tag{27}$$

This then implies that

$$\begin{aligned} &(t_{i,j_1} - t_{i,k_2}) \bmod N(N+1) \\ &= (t_{i,j_2} - t_{i,k_1}) \bmod N(N+1) = d'', \end{aligned} \tag{28}$$

for some $1 \leq d'' \leq N(N+1)-1$. Thus, $|\tilde{S}_i| \leq N(N+1)-2$ and this contradicts to (26). Therefore, p cannot be $N(N+1)$. ■

C. Hierarchical construction of asynchronous CH sequences

For $N = 2$ and $N = 8$, 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 beat the best known MACH sequence in the literature. Our idea for this is the hierarchical construction that constructs a MACH sequence by using two smaller MACH sequences as shown in the following theorem.

Theorem 19: 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.,

$$q(t) = \lfloor t/(2p_1 - 1) \rfloor, \tag{29}$$

and

$$r(t) = (t \mod (2p_1 - 1)). \tag{30}$$

Let

$$c(t) = c_1(r(t)) + c_2(q(t)) * N_1. \tag{31}$$

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

Proof. (i) (Periodicity) To see that $c(t)$ is periodic with period $(2p_1 - 1) * p_2$, let $t' = t + (2p_1 - 1) * p_2$. Then $r(t') = r(t)$ and $q(t') = q(t) + p_2$. Thus, we have $c_1(r(t)) = c_1(r(t'))$ and $c_2(q(t)) = c_2(q(t) + p_2) = c_2(q(t'))$ from the periodicity of $c_2(t)$. In view of the construction of the sequence of $c(t)$ in (31), we then have $c(t) = c(t') = c(t + (2p_1 - 1) * p_2)$.

(ii) (Maximum degree of overlapping) We need to show for any $0 \leq d \leq (2p_1 - 1) * p_2 - 1$ and $0 \leq i \leq N - 1$, there exists $\tau(i, d)$ such that

$$0 \leq \tau(i, d) \leq (2p_1 - 1) * p_2 - 1,$$

and

$$c(\tau(i, d)) = c(\tau(i, d) + d) = i.$$

For any $0 \leq i \leq N_1 * N_2 - 1$, we let $i_1 = (i \mod N_1)$ and $i_2 = \lfloor i/N_1 \rfloor$. Then $i = i_1 + i_2 * N_1$. Now for $0 \leq d \leq (2p_1 - 1) * p_2 - 1$, we consider the following two cases:

Case 1. $0 \leq r(d) \leq p_1 - 1$:

Let $d_1 = r(d)$ and $d_2 = q(d)$. Clearly,

$$d = d_1 + d_2 * (2p_1 - 1). \tag{32}$$

Since $0 \leq r(d) \leq p_1 - 1$ and $0 \leq d \leq (2p_1 - 1) * p_2 - 1$, we know that $0 \leq d_1 \leq p_1 - 1$ and $0 \leq d_2 \leq p_2 - 1$. Thus, it follows from the maximum degree of overlapping property of $\{c_1(t), t \geq 0\}$ and $\{c_2(t), t \geq 0\}$ that for $0 \leq i_1 \leq N_1 - 1$ and $0 \leq i_2 \leq N_2 - 1$ there exist $\tau_1(i_1, d_1)$ and $\tau_2(i_2, d_2)$ such that

$$0 \leq \tau_1(i_1, d_1) \leq p_1 - 1, \tag{33}$$

$$c_1(\tau_1(i_1, d_1)) = c_1(\tau_1(i_1, d_1) + d_1) = i_1, \tag{34}$$

$$0 \leq \tau_2(i_2, d_2) \leq p_2 - 1 \tag{35}$$

$$c_2(\tau_2(i_2, d_2)) = c_2(\tau_2(i_2, d_2) + d_2) = i_2. \tag{36}$$

Now let

$$\tau(i, d) = \tau_1(i_1, d_1) + \tau_2(i_2, d_2) * (2p_1 - 1). \tag{37}$$

In view of (33) and (35), we know from (37) that

$$0 \leq \tau(i, d) \leq (2p_1 - 1) * p_2 - 1. \tag{38}$$

Observe from (37) that

$$r(\tau(i, d)) = \tau_1(i_1, d_1),$$

and

$$q(\tau(i, d)) = \tau_2(i_2, d_2).$$

Thus, it follows from (31), (34) and (36) that

$$\begin{aligned} &c(\tau(i, d)) = c_1(\tau_1(i_1, d_1)) + c_2(\tau_2(i_2, d_2)) * N_1 \\ &= i_1 + i_2 * N_1 = i. \end{aligned} \tag{39}$$

On the other hand, it follows from (37) and (32) that

$$\tau(i, d) + d = \tau_1(i_1, d_1) + d_1 + (\tau_2(i_2, d_2) + d_2) * (2p_1 - 1). \tag{40}$$

Since $0 \leq \tau_1(i_1, d_1) \leq p_1 - 1$ and $0 \leq d_1 \leq p_1 - 1$, we have

$$0 \leq \tau_1(i_1, d_1) + d_1 < 2p_1 - 1.$$

Thus, we know that

$$r(\tau(i, d) + d) = \tau_1(i_1, d_1) + d_1$$

and that

$$q(\tau(i, d) + d) = \tau_2(i_2, d_2) + d_2.$$

It then follows from (31), (34) and (36) that

$$\begin{aligned} &c(\tau(i, d) + d) \\ &= c_1(\tau_1(i_1, d_1) + d_1) + c_2(\tau_2(i_2, d_2) + d_2) * N_1 \\ &= i_1 + i_2 * N_1 = i. \end{aligned} \tag{41}$$

In view of (39) and (41), we conclude that

$$c(\tau(i, d)) = c(\tau(i, d) + d) = i.$$

Case 2. $p_1 \leq r(d) < 2p_1 - 1$:

In this case, let

$$\tilde{d} = (2p_1 - 1) * p_2 - d. \tag{42}$$

Since $p_1 \leq r(d) < 2p_1 - 1$, we know that $d \neq 0$ and thus $0 < d < (2p_1 - 1) * p_2$. From (42), we have

$$0 < \tilde{d} < (2p_1 - 1) * p_2$$

and

$$0 < r(\tilde{d}) = 2p_1 - 1 - r(d) \leq p_1 - 1. \quad (43)$$

It then follows from Case 1 that there exists $\tau(i, \tilde{d})$ such that

$$0 \leq \tau(i, \tilde{d}) \leq (2p_1 - 1) * p_2 - 1, \quad (44)$$
$$c(\tau(i, \tilde{d})) = c(\tau(i, \tilde{d}) + \tilde{d}) = i. \quad (45)$$

Now let

$$\tau(i, d) = \Big((\tau(i, \tilde{d}) + \tilde{d}) \bmod (2p_1 - 1) * p_2)\Big). \quad (46)$$

Clearly, we have

$$0 \leq \tau(i, d) \leq (2p_1 - 1) * p_2 - 1.$$

In conjunction with (42), we also have

$$\begin{aligned}&(\tau(i, d) + d) \bmod ((2p_1 - 1) * p_2)\\ &= ((\tau(i, \tilde{d}) + \tilde{d} + d) \bmod (2p_1 - 1) * p_2)\\ &= \tau(i, \tilde{d}). \end{aligned} \quad (47)$$

Using (46), (45), (47) and the periodicity of $c(t)$ yields

$$\begin{aligned}&c(\tau(i, d)) = c(\tau(i, \tilde{d}) + \tilde{d})\\ &= c(\tau(i, \tilde{d})) = c(\tau(i, d) + d) = i.\end{aligned}$$

■

As an illustrating example, we can use the $(8, 73)$-MACH sequence in [20] as the two MACH sequences in Theorem 19 to construct a $(64, 10585)$-MACH sequence. On the other hand, one can also use the CRSEQ scheme in [15] to construct a $(67, 13400)$-MACH sequence for a system of 64 channels. For a system of 64 channels, our 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 our hierarchical construction to construct a MACH sequence for systems with a large number of channels.

IV. Simulation results

In this section, we compare the performance of various channel hopping algorithms by computer simulations, including SYN-MAC [11], SSCH [8], L-QCH [12], RRICH [1], CACH [1] EJSCH [23], AHWCH [34] and ETCH [18]. We do this by implementing an event-driven C++ simulator for an IEEE 802.11a ad hoc network. Both SUs and PUs are distributed randomly in a 380mx380m region. Each simulation result is obtained by averaging 30 randomly generated topologies. In our simulations, we only consider disjoint flows, where each source SU and each destination SU cannot have multiple flows. To avoid the scenario of having many SUs choosing the same hopping seed, we also allow SUs to change their hopping seeds. Specifically, we let each SU change its hopping seed with probability 1/1000 at the beginning of each time interval. The transmission range (the range within which a message is successfully received if there is no interference from other SUs) and the interference range (the range within which SUs in receive mode will be interfered with by an unrelated transmitting SU) are set to 250m and 550m, respectively. The behavior of a PU is modeled by a (discrete-time) Gilbert-Elliott model [32], [33] (see Figure 2). The Gilbert-Elliott model is an Markov chain with only two states: the BUSY state and the IDLE state. The transition probability from the BUSY state to the IDLE state in the next time interval is $1-\alpha$ and the transition probability from the IDLE state to the BUSY state in the next time interval is $1 - \beta$. The simulation time is set to 6s. The length of each time interval is set to 6 ms. In the first 2 ms, each pair of SUs will exchange the control messages. If both SUs correctly receive the control messages from each other, then we have a successful rendezvous. We note that our simulations are implemented from the MAC layer (without using any detailed wireless channel model) and all SUs have to contend the right of accessing the channel through the IEEE 802.11 Distributed Coordination Function. Also, in order to see the effects (and the insights) of these rendezvous algorithms, we do not simulate data traffic for the remaining 4ms after a successful rendezvous in a time interval of 6ms.

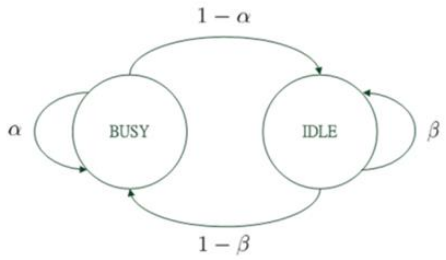
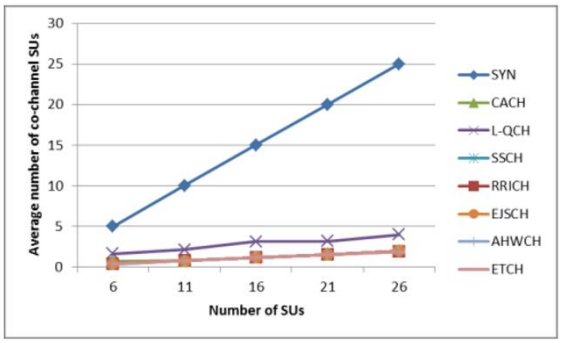
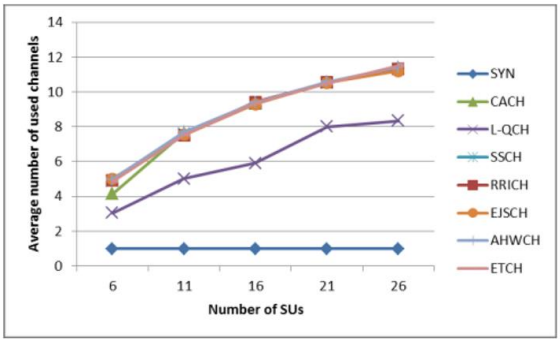


Fig. 2: **The Gilbert-Eliott model for a PU.**

In our first experiment, there are 13 channels and each channel is associated with a PU. The parameter β of the Gilbert-Elliott model is set to 0.5 for each PU. On the other hand, the parameter α is set to 0.5 with probability $1/5$ and 0.7 with probability $4/5$. As each SU is within the interference range of another SU, each SU is a neighbor of another SU and thus the number of neighboring SUs (including itself) of an SU is simply the total number of SUs. As suggested in [1], the number of logical channels u in CACH is set to be the minimum of the number of SUs and the number of channels in our simulations. Also, to simulate AHWCH, the length of the ID of each SU is set to 6 bits. In Figure 3(a), we show the effect of the number of SUs on the average number of co-channel SUs (for a particular SU). A co-channel SU for a particular SU in a particular time interval is an SU that operates on the same channel as that SU in the time interval. To measure this, we choose an arbitrary SU and take the average of the number of co-channel SUs over time. Clearly, the larger the average number of co-channel SUs is, the larger the co-channel interference is. As such, a large average number of co-channel SUs might suffer from the problem of control channel saturation. It can be seen from Figure 3(a) that the average numbers of co-channel SUs for these algorithms are quite close except SYN and L-QCH. When the number of SUs is larger than 11, the number of logical channels is upper bounded by the number of channels. In this case, we set the number of logical channels of CACH to the number of channels and

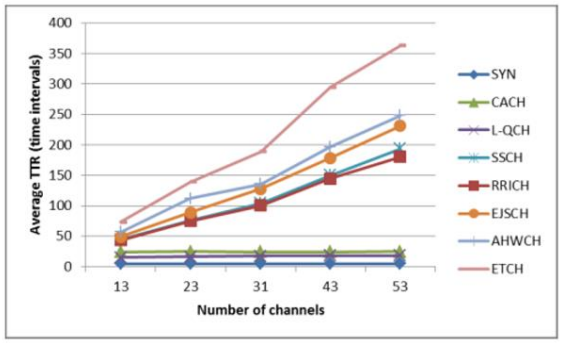


(a) Effect of the number of SUs on the average number of co-channel SUs.

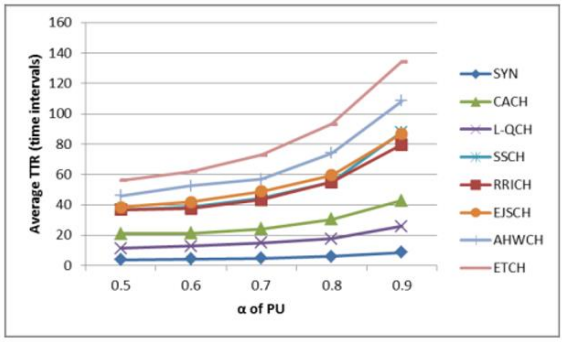


(b) Effect of the number of SUs on the average number of used channels.

Fig. 3: Effects of the number of SUs on the average number of co-channel SUs and the average number of used channels.



(a) Effect of the number of channels on average TTR.



(b) Effect of the PU behavior on the average TTR.

Fig. 4: Effects of the number of channels and the PU behavior on the average TTR.

CACH is reduced to RRICH. Also, the average number of co-channel SUs for L-QCH is larger than those of RRICH, CACH, SSCH, EJSCH, AHWCH and ETCH. Such a result is expected as the channel loading of L-QCH is larger than the channel loading of RRICH, CACH, SSCH, EJSCH, AHWCH and ETCH. Since the channel loading of SYN-MAC is 1, the average number of co-channel SUs for SYN-MAC increases linearly with respect to the number of SUs. On the other hand, the average numbers of co-channel SUs for RRICH, CACH, SSCH, EJSCH, AHWCH, ETCH and L-QCH only increase slowly with respect to the increase of the number of SUs. In Figure 3(b), we further show the effect of the number of SUs on the average number of used channels in a time interval. A channel is said to be used in a time interval if there is (at least) one SU that operates that channel as the control channel in that time interval. To measure this, we count the number of used channels in every time interval and then take its average. Intuitively, a CH scheme that has a large average number of used channels tends to distribute its traffic evenly over the channels. It is observed that RRICH, CACH, SSCH, EJSCH, AHWCH and ETCH have the same average number of used channels when the number of SUs is larger than 11. When the number of SUs is only 6, the average number of used channels for RRICH is larger than that of CACH because the channel loading in CACH is larger than that of RRICH. Also, the average number of used channels for CACH is better than L-QCH, even when the number of SUs is 6. This is because L-QCH only distributes the control traffic over the time (but not over the channels).

In our second experiment, we consider six disjoint flows. The transmitters (source SUs) of these flows are first distributed randomly in the region. Each receiver (destination SU) is then distributed randomly and repeatedly in the region until it is within the range of its transmitter. In Figure 4(a), we show the effect of the number of channel on the average TTR. Since SYN-MAC allows all SUs to hop to the same channel, it has the lowest average TTR. Since the MTTRs of CACH and L-QCH are independent of the number of channels, their average TTRs are also not influenced by the number of channels. However, the MTTRs of RRICH, SSCH, EJSCH, AHWCH, and ETCH depend on the number of channels, and the average TTRs of RRICH, SSCH, EJSCH, AHWCH, and ETCH increases when the number of channels increases. Moreover, since the degree of overlapping of RRICH is N, RRICH has a lower average TTR than SSCH (as SSCH suffers from the PU long-time blocking problem). We also note that the slopes of the average TTRs of AHWCH and EJSCH are higher than those of RRICH and SSCH in Figure 4(a). This is because both SSCH and RRICH have the advantage of time synchronization. On the other hand, both AHWCH and EJSCH are designed without the need of time synchronization. In Figure 4(b), we measure the effect of α (of a PU) on the average TTR. The larger α is, the longer the time for a PU to be on the BUSY state. When α of each PU is between 0.5 and 0.8, RRICH and SSCH almost have the same average TTR. However, when α is set 0.9, the average

TTR of SSCH is increased rapidly due to the long-time PU blocking problem. Also, in AHWCH and EJSCH there is a "stay" mode that requires an SU to stay on a channel for a long period of time. If that channel is blocked by a PU, then that SU will not meet its counterpart during its "stay" mode and thus AHWCH and EJSCH might suffer from the long-time PU blocking problem. It seems that ETCH also suffers from the long-time PU blocking problem as an SU might hop to a certain channel more often than the other channels in an ETCH sequence. The three CH schemes, SYN, CACH and LQCH, do not increase their average TTRs quickly because their degree of overlapping is equal to N and then they are immune to the long-time PU blocking problem. In view of Figure 4(a) and Figure 4(b), we note that L-QCH has a lower average TTR than that of CACH. This is because the channel loading of L-QCH is larger than that of CACH. On the other hand, CACH have a smaller number of co-channel SUs and a larger number of used channels than those of L-QCH. We also note that CACH requires to optimize the number of logical channels u. If such an optimization is not possible and the number of channels is not too large, then RRICH could be a more practical choice than CACH.

V. Discussions

In this section, we discuss two additional methods/assumptions to enhance rendezvous algorithms: (i) the unique ID assumption in Section V-A and (ii) the assumption of the available channel set in Section V-B.

A. Unique ID

In view of the theoretical results in Table I, it is clear that the rendezvous problem is relative easy in the *asymmetric* setting. In order for the two SUs to play different roles (as a sender/child or a receiver/mommy), one commonly used assumption in the literature is to assume there is a unique ID for each SU, for instance the 48-bit universal MAC address (see e.g., [21], [34]). Now suppose that each user is assigned with a unique n-bit ID $(b_0, b_1, \ldots, b_{n-1})$. In order to use the unique ID assumption in the asynchronous setting, the trick is then to transform the unique ID into another m-bit codeword $(c_0, c_1, \ldots, c_{m-1})$ so that it is *cyclically unique*, i.e., for any cyclic shift d, there does not exist another codeword $(c'_0, c'_1, \ldots, c'_{m-1})$ that is the same as $(c_d, c_{d+1}, \ldots, c_{(m-1+d) \bmod m})$. Then each user can use the $(t \bmod m)^{th}$ bit of its codeword to determine the role it should play in the t^{th} time period. The cyclic uniqueness property ensures that there is some period among the first m time period that the two SUs will play different roles.

The method used in [21] to transform the unique ID $(b_0, b_1, \ldots, b_{n-1})$ into another m-bit cyclically unique codeword $(c_0, c_1, \ldots, c_{m-1})$ is to append $(b_0, b_1, \ldots, b_{n-1})$ with n consecutive 1's and n consecutive 0's. This results in a $3n$-bit codeword. On the other hand, the method used in [34] is to add a new symbol "2" in front of the n-bit ID. This creates a problem of dealing with the additional symbol "2."

Here we discuss a simple method of transforming the unique ID $(b_0, b_1, \ldots, b_{n-1})$ into another m-bit cyclically unique codeword $(c_0, c_1, \ldots, c_{m-1})$. Note that the concatenation of n bits of consecutive 1's and n bits of consecutive 0's in [21] in fact acts a *delimiter*. So is the new symbol "2" in [34]. This is similar to adding a delimiter in front of a packet. One widely used delimiter for framing packets is to use the 0111110 bit sequence and then use the bit-stuffing algorithm to ensure that there are no six consecutive 1's in the body of the message. The problem of using the bit-stuffing algorithm is that it generates a variable length code. Though it is fine for a variable length packet, it is not suitable for fixed length codewords here. In Lemma 3 of [35], it was shown that the C-transform can be used as a one-to-one mapping that maps an n-bit ID to another m-bit codeword that does not have ℓ consecutive 1's for any $\ell \geq 2$.

B. Available channel set

In some recent papers (se e.g., [14], [17], [23]–[25], the available channel set to each user is assumed to be fixed and known to each user. Such information can be used for speeding up the rendezvous process. In particular, Chang and Huang [14] proposed a fast rendezvous channel hopping algorithm (FRCH) that re-maps all the unavailable channels in a round of $2N+1$ time slots in DRSEQ [16] alternatively to an available channel. By doing so, FRCH still has the same MTTR $(2N+1)$ as DRSEQ [16] and its MCTTR can be reduced to $N(2N+1)$ (for some N specified in [14]), which is better than $N(3N-1)$ in CRSEQ [15]. Such an algorithm performs well when the number of available channels is $O(N)$. However, when the number of available channels to each user is much smaller than N, re-mapping all the unavailable channels in a round of $2N+1$ time slots to the same available channel might result in an average TTR of order $O(N)$. When the number of available channels to each user is much smaller than N, the modular clock algorithm [17] might be a better choice for minimizing the average TTR as it only needs to cycle through the channels that are available to each user. However, the modular clock algorithm does not have a bounded MCTTR (Proposition 5 of [17]). A compromise might be for each user to use a good MACH (e.g., the (8,73)-MACH in [20]) and simply remaps every unavailable channel at random to one of its available channel. By doing so, the MCTTR is still bounded. Also, when the the number of available channels to each user is much smaller than N, it behaves as the random algorithm (Algorithm 1 in [17]) that has an average TTR independent of N.

Finally, we note that the rendezvous problem with the information of available channel set is also relative easy in the *asymmetric* setting. Suppose that user 1 (resp. user 2) is the *sender* (resp. *receiver*) and it has n_1 (resp. n_2) available channels. Then user 1 can choose a prime p_1 not less than n_1 and user 2 can choose an even number p_2 not less than n_2. At time t, if $(t \bmod p_i) < n_i$, user i selects its $(t \bmod p_i)^{th}$ available channel from its available channel set. Otherwise, it selects at random a channel from its channel available set. As p_1 and p_2 are relatively prime, all the $n_1 n_2$ available channel pairs will be visited at least once in a period of $p_1 p_2$ time slots. Thus, the MCTTR is $p_1 p_2$. To extend such an approach to the symmetric setting, one might use the unique ID method

described in the previous section to determine the role of each user. We note that the unique ID method in [34] requires determining three different roles in order to have a bounded MCTTR.

VI. Conclusion

In this paper, we derived various lower bounds for MTTR and MCTTR of various CH schemes in the multichannel rendezvous problem. We also showed there are CH schemes that achieve these lower bounds, and thus these lower bounds are tight for some choices of u and N. However, there are still some theoretical gaps:
(i) In the symmetric and asynchronous setting, the gap between our lower bound $N^2 + N + 1$ in Theorem 18 and the best achievable result $N(3N - 1)$ in [15] is still very large. Also, the lower bound in Theorem 18 is for the period of a MACH sequence. It is not a lower bound for MCTTR.
(ii) In the symmetric and synchronous setting, the MCTTR of CACH is $(u+1)N$, which is still larger than the lower bound uN in Theorem 16. The main difficulty is to incorporate the symmetric assumption in (A3) in the proof of Theorem 16.
(iii) If the primary users are often on from time to time, the average-time-to-rendezvous is a better performance metric than MTTR. One possible extension is to consider the average time-to-rendezvous when there are randomly blocked channels as in [26]. It is possible to derive a lower bound for such a performance metric under a channel loading constraint. However, it is not clear whether such a lower bound would be tight.

Acknowledgement

This work was supported in part by the Excellent Research Projects of National Taiwan University, under Grant Number AE00-00-04, and in part by National Science Council (NSC), Taiwan, under Grant Numbers NSC102-2221-E-002-014-MY2 and 102-2221-E-007-006-MY3.

References

[1] 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*.
[2] Federal Comm. Commission, "Spectrum policy task force report," Washington, DC, *FCC 02-155*, 2002.
[3] J. Mitola III and G. Q. Maguire Jr.,"Cognitive radio: making software radios more personal," *IEEE Personal Communications*, Aug. 1999.
[4] J. Zhao, H. Zheng and G.-H. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *Proc. IEEE DySPAN'05*.
[5] L. Le and E. Hossain, "OSA-MAC: a MAC protocol for opportunistic spectrum access in cognitive radio networks," in *Proc. IEEE WCNC'08*.
[6] T. Chen, H. Zhang, G. M. Maggio, and I. Chlamtac, "CogMesh: a cluster-based cognitive radio network," in *Proc. IEEE DySPAN'07*.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] Y. R. Kondareddy and P. Agrawal, "Synchronized MAC protocol for multi-hop cognitive radio networks," in *Proc. IEEE ICC'08*.
[12] 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.
[13] 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*.
[14] G.-Y. Chang and Jen-Feng Huang, "A fast rendezvous channel-hopping algorithm for cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 7, pp. 1475–1478, 2013.
[15] 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.
[16] D. Yang, J. Shin, and C. Kim, "Deterministic rendezvous scheme in multichannel access networks," *Electronics Letters*, vol. 46, no. 20, pp. 1402-1404, 2010.
[17] 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.
[18] 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*.
[19] 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*.
[20] 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.
[21] 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.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] 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.
[27] S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous*. Dordrecht: Kluwer Academic Publishers, 2003.
[28] R. P. Grimaldi. *Discrete and Combinational Mathematics: An Applied Introduction*. Addison Wesley 2004.
[29] E. J. Anderson and R. Weber, "The rendezvous problem on discrete locations," *Journal of Applied Probability,* vol. 28, pp. 839–851, 1990.
[30] A.W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*. New York: Academic Press, 1979.
[31] 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.
[32] E. N. Gilbert, "Capacity of a burst noise channel," *Bell system technical journal*, vol. 39, no. 5 pp. 1253–1265, 1960.
[33] 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.
[34] 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.
[35] 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.

Cheng-Shang Chang (S'85-M'86-M'89-SM'93-F'04) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1983, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, USA, in 1986 and 1989, respectively, all in Electrical Engineering. From 1989 to 1993, he was employed as a Research Staff Member at the IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y. Since 1993, he has been with the Department of Electrical Engineering at National Tsing Hua University, Taiwan, R.O.C., where he is a Tsing Hua Chair Professor. His current research interests are concerned with network science, high speed switching, communication network theory, and mathematical modeling of the Internet. Dr. Chang received an IBM Outstanding Innovation Award in 1992, an IBM Faculty Partnership Award in 2001, and Outstanding Research Awards from the National Science Council, Taiwan, in 1998, 2000 and 2002, respectively. He also received Outstanding Teaching Awards from both the college of EECS and the university itself in 2003. He was appointed as the first Y. Z. Hsu Scientific Chair Professor in 2002 and elected to an IEEE Fellow in 2004. Dr. Chang received the Academic Award from the Ministry of Education and the Merit NSC Research Fellow Award from the National Science Council in 2011. He is the author of the book "Performance Guarantees in Communication Networks" and the coauthor of the book "Principles, Architectures and Mathematical Theory of High Performance Packet Switches." He served as an editor for Operations Research from 1992 to 1999 and an editor for IEEE/ACM Transactions on Networking from 2007 to 2009. He is currently serving as an editor-at-large for IEEE/ACM Transactions on Networking and an editor for IEEE Transactions on Network Science and Engineering. Dr. Chang is a member of IFIP Working Group 7.3.

Wanjiun Liao received the PhD degree in electrical engineering from the University of Southern California in 1997. She is a distinguished professor of electrical engineering in National Taiwan University (NTU), Taipei, and an adjunct research fellow of the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. Her research interests include the design and analysis of wireless multimedia networking, cloud data center networking, and green communications. She was on the editorial boards of the *IEEE Transactions on Wireless Communications* and the *IEEE Transactions on Multimedia*. She has also served on the organizing committees of many international conferences, including serving as the TPC vice chair of the IEEE GLOBECOM 2005 Symposium on Autonomous Networks, TPC cochair of the IEEE GLOBECOM 2007 General Symposium, TPC cochair of the IEEE VTC 2010 Spring, and the TPC cochair of the IEEE ICC 2010 Next Generation Networking and Internet Symposium. The papers she coauthored with her students won the Best Paper Awards of the IEEE ICME 2000 and the IEEE GLOBECOM 2011, and IEEE ComSoc 2011 Multimedia Communications Best Paper Award. She received the Republic of China (R.O.C.) Distinguished Women Medal in 2000 and was elected as a distinguished lecturer of the IEEE Communications Society (2011-2012). She is a fellow of the IEEE.

Tsung-Ying Wu received the B.S. degree in Department of Information Management from National Central University and the M.S. degree in Department of Computer Science from National Tsing Hua University in 2003 and 2007, respectively. He is currently a Ph.D. student in Electrical Engineering Department of National Taiwan University. His current research area is cognitive radio network, focusing on the problems of control channel establishment, cooperative transmission and energy harvesting issues. Mr. Wu received a scholarship, called the Sandwich program, in Taiwan and he was a visiting student at Erlangen-Nuernberg University in Germany from 2014 to 2015.