Channel-Hopping Based on Available Channel Set for Rendezvous of Cognitive Radios

Lu Yu[1], Hai Liu[1], Yiu-Wing Leung[1], Xiaowen Chu[1], Zhiyong Lin[2,1]
[1]Dept of Computer Science, Hong Kong Baptist University, Hong Kong
[2]Dept of Computer Science, GuangDong Polytechnic Normal University, GuangZhou, China
Email: {lyu, hliu, ywleung, chxw, zylin}@comp.hkbu.edu.hk

***Abstract*—Rendezvous is a necessary operation for cognitive users to establish communication links in cognitive radio networks (CRNs). To guarantee the rendezvous in finite time, all existing rendezvous algorithms generate CH (channel-hopping) sequences using the whole channel set and attempt rendezvous on each of the channels (i.e., both available channels and unavailable channels). In practice, the available channel set is usually a small portion of the whole channel set due to dynamics of channel availabilities and limited sensing capabilities of cognitive users. Thus, the CH sequences using the whole channel set may attempt unnecessary rendezvous in uncertain channels (e.g., unavailable channels or randomly-selected channels) which greatly degrades the performance. In this study, we propose a new rendezvous algorithm that generates *channel-hopping sequences based on available channel set (CSAC)* for more efficient rendezvous. We prove that CSAC gives guaranteed rendezvous and derive its upper-bound on maximum time-to-rendezvous (MTTR) which is an expression of the number of available channels instead of the number of all potential channels. To the best of our knowledge, CSAC is the first one in the literature that exploits the only available channels in designing CH sequences while providing guaranteed rendezvous. Experimental results show that CSAC can significantly improve the MTTR compared to state-of-the-art.**

***Index Terms*—cognitive radio; rendezvous; channel hopping.**

I. Introduction

In conventional wireless networks, a majority of the licensed spectrum is underutilized while the unlicensed spectrum is over-crowded due to exponential growth of various wireless devices [1]. Cognitive radio network (CRN) is emerging as a new communication paradigm to exploit the wireless spectrum in a more intelligent and flexible way. Cognitive radio is a device which can sense the spectrum holes and access them by adaptively adjusting its transmission parameters. With cognitive radios, unlicensed users (or cognitive users, or secondary users (SUs)) are able to identify and access the vacant portions of the licensed spectrum which are not used by any licensed users (or primary users (PUs)). Unless otherwise specified, the users mentioned hereafter in this paper refer to SUs by default.

In CRNs, SUs remain unacquainted with each other before completing "rendezvous" which is defined as the process that two or more SUs establish a communication link on a common channel [1]. Rendezvous is an elementary operation on which almost all communications in CRNs rely. Channel-hopping (CH) is a typical technique for rendezvous. With the CH technique, each user hops among its available channels for rendezvous with its potential neighbors and rendezvous is achieved if two users hop on the same available channel at the same time. A channel is said to be available to a user if the user can operate on the channel without causing any interference to the PUs.

Before starting CH, users should identify available channels in the licensed spectrum via a process named *spectrum sensing* [2]. A user usually identifies only a small portion of the whole channel set as its available channels, due to dynamics of channel availability, its limited sensing capability and limited sensing time. For example, work in [3] optimizes the detection (sensing) time for channel efficiency. It shows that when the SNR (signal-to-noise ratio) is -3 dB, the channel availability ratio with the optimal detection time is only about 15% [3]. That is, the available channel set is usually a small portion of the whole channel set in practice.

There are a number of CH algorithms which provide guaranteed rendezvous (i.e., two users definitely achieve rendezvous in finite time) and a survey about this topic can be found in [4]. These algorithms generate the CH sequences based on the whole channel set and attempt rendezvous on each of the channels including both available and unavailable channels. Since rendezvous is attempted on each channel, two users can achieve rendezvous in finite time as long as they share at least one commonly-available channel. However, attempting rendezvous on unavailable channels does not help at all but unavoidably increases the time the users take to achieve rendezvous (i.e., time-to-rendezvous, TTR). In this sense, these existing algorithms become inefficient when available channels are far less than unavailable channels. To address this problem, some algorithms (e.g., [5]) employ a *random replacement* operation in which unavailable channels are randomly replaced by available channels so as to increase chances of rendezvous. However, the random replacement operation does not change the very essence of these algorithms. In extreme cases (most channels are unavailable), an algorithm with the random replacement operation degrades to a pure random algorithm.

In this study, we propose a new rendezvous algorithm that generates CH sequences based on available channel set (CSAC for short). We assume that there are two roles (either a *sender* or a *receiver*) for the users involved in a rendezvous process. This assumption is reasonable in many applications of wireless networks. For example, in broadcasting and routing of an ad hoc network, the source/forwarder usually operates as a sender which attempts rendezvous with its neighbors operating

TABLE I
COMPARISON OF OUR ALGORITHM AND THE EXISTING RENDEZVOUS ALGORITHMS UNDER THE ASYMMETRIC MODEL

Algorithms	Upper-bound on MTTR
CSAC	$n^2(m_p-1)-(G-2)n$, when $n \neq km_p$ (nm_p-G+1), when $n=km_p$ ($k>1$ and $k \in Z$)
Enhanced Jump-Stay [7]	$4P(P+1-G)$
Jump-Stay [5]	$6QP(P-G)$
CRSEQ [8]	$P(3P-1)$
MMC [9]	Infinity
Random [10]	Infinity

Remarks: m, n are the numbers of available channels of two users, respectively; m_p is the smallest prime number which is not smaller than m; Q is the number of all potential channels; P is the smallest prime number which is not smaller than Q; G is the number of commonly-available channels.

as the receivers. Based on the different roles of the users, CSAC generates CH sequences for the sender and the receiver, respectively. The contribution of this work is three-fold.

1) We propose a new rendezvous algorithm named CSAC. To the best of our knowledge, CSAC is the first one in the literature that exploits the only available channels in designing CH sequences while providing guaranteed rendezvous.
2) We prove that CSAC provides guaranteed rendezvous and derive an upper-bound on the maximum TTR (MTTR) of CSAC which is an expression of the number of available channels instead of the number of all potential channels (shown in Table I).
3) We conduct extensive simulation to evaluate the performance of CSAC. We found that the MTTR of CSAC is significantly smaller than those of the state-of-the-art when the ratio of available channels to all potential channels is less than 30%.

Table I compares the MTTR of CSAC and those of existing rendezvous algorithms. Under the asymmetric model [5], different users might have different available channels.

The rest of this paper is organized as follows. Related work is reviewed in Section II. System model and problem formulation are presented in Section III. In Section IV, we propose the CSAC algorithm and analyze its theoretical performance. Simulation results are presented in section V. We conclude our work in Section VI.

II. RELATED WORK

We classify the existing CH algorithms into two categories: i) centralized systems where a central server is preselected to allocate the spectrum for all SUs, and ii) decentralized systems where there is no central server.

Centralized systems: DSAP [11] and DIMSUMNet [12] are two typical centralized systems in which a server operates over a common control channel (CCC) accessible to all users. The server is responsible for scheduling the data communications between users. The centralized systems are not widely adopted due to their poor scalability and low robustness.

Decentralized systems using CCC: In the decentralized systems in [13] and [14], a global CCC is preselected and is well known to all users. Some other rendezvous systems are based on local CCC rather than global CCC. In [15] and [16], a cluster-based control-channel method was proposed, in which a local CCC is selected for each group. However, the extra costs in establishing and maintaining the global/local CCCs are considerable.

Decentralized systems without using CCC: The decentralized system without using CCC is referred to as the *blind rendezvous* system [5]. In recent years, it has drawn ever increasing attentions of researchers. Jump-Stay [5] [6] generates CH sequences composed of the jump pattern and the stay pattern. It gives guaranteed rendezvous by three combinations of the patterns: jump-stay, jump-jump and stay-stay. The Jump-Stay was further improved in [7] where the upper-bounds on both the MTTR and expected TTR were lowered to $O(P^2)$ where P is the smallest prime number greater than the number of all potential channels. Theis et al. presented an efficient CH sequence generating mechanism called modular clock algorithm (MC) and its modified version MMC in [9]. The main idea of MC and MMC is that each user picks a proper prime number and randomly selects a rate less than the prime number. Based on the prime number and the rate, the user generates its CH sequence via predefined modulo operations. However, MC and MMC cannot guarantee rendezvous if the selected rates of two users are identical. Yang et al. proposed two rendezvous algorithms, namely deterministic rendezvous sequence (DRSEQ) [17] and channel rendezvous sequence (CRSEQ) [8], which provide guaranteed rendezvous for the symmetric model (i.e., all users have the same available channels) and the asymmetric model, respectively. In CRSEQ, the sequence is generated based on triangle numbers and modulo operations. Bian et al. [18] presented an asynchronous channel hopping (ACH) algorithm which aims to maximize rendezvous diversity. It assumes that each user has a unique ID and ACH sequences are designed based on the user ID. Though the length of user ID is a constant, it may result in a long TTR in practice given that a typical MAC address contains 48 bits. There are other algorithms in this category such as M-/L-QCH [19], synchronous QCH [10], AMRCC [20], SYNCETCH and ASYNC-ETCH [21], MtQS-DSrdv [22], Ring-Walk [23] [26], and C-MAC [24]. Due to limited space, these algorithms are not reviewed and readers may refer to the survey in [4] for details.

To the best of our knowledge, among all existing rendezvous algorithms, only MC/MMC generates CH sequences based on the only available channel set. Unfortunately, MC/MMC does not guarantee the rendezvous in finite time. The algorithms other than MC and MMC generate CH sequences based on the whole channel set.

III. SYSTEM MODEL AND PROBLEM FORMULATION

Time is divided into slots of equal duration in the paper. We do not require time-synchronization. The licensed spectrum is divided into Q ($Q \geq 1$) non-overlapping channels $C = \{c_1, c_2, ..., c_Q\}$, where c_i denotes the i-th channel and usually is called channel i for convenience. All users in the network know the indices of channels. Let $C_i \in C$ denote the available channel set of user i $(i = 1, 2, ..., K)$, where a channel is said to be available to a user if the user can communicate on the channel without causing interference to any PUs.

We consider the rendezvous of two users, say user i and user j. Both users i and j are equipped with single cognitive radio. There are two roles of the users: sender and receiver. Without loss of generality, user i is the sender and user j the receiver. User i has m available channels $C_i = \{C_1^i, C_2^i, ..., C_m^i\}$ and user j has n available channels $C_j = \{C_1^j, C_2^j, ..., C_n^j\}$. C_i is usually not identical to C_j in practice due to dynamics of channel availability. Let G denote the number of channels commonly available to users i and j. There is at least one commonly-available channel between them (i.e., $G \geq 1$). Otherwise, the rendezvous is not possible.

Rendezvous Problem: Consider any pair of users denoted by user i and user j. User i is the sender with m available channels $C_i = \{C_1^i, C_2^i, ..., C_m^i\}$, and user j is the receiver with n available channels $C_j = \{C_1^j, C_2^j, ..., C_n^j\}$. The problem is to design a rendezvous algorithm that generates CH sequences for user i and user j respectively, such that users i and j are guaranteed to hop on a commonly-available channel in the same time slot, regardless of different time when the users start their CH sequences.

Time-to-rendezvous (TTR) is an important metric to evaluate the performance of rendezvous algorithms. It is rigorously defined as the number of time slots to successful rendezvous after all users have started channel-hopping. Since we do not reqeire time-synchronization and availability of channels may change all the time, the TTR of a rendezvous algorithm is usually not constant. The maximum TTR (MTTR) and average TTR are used to evaluate the overall performance in the simulation (Section V).

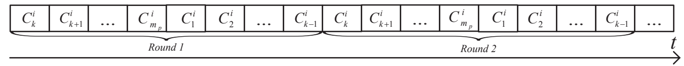
IV. CSAC ALGORITHM

A. Algorithm Description

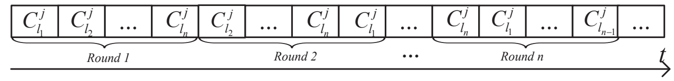
CSAC generates CH sequences for sender-role users and receiver-role users, respectively. Both sender-role sequence and receiver-role sequence are based on the only available channel set instead of the whole channel set.

Sender-role sequence (user i): Given $C_i = \{C_1^i, C_2^i, ..., C_m^i\}$, m_p is determined to be the smallest prime number which is not smaller than m. We first expand C_i and get m_p channels $\{C_1^i, C_2^i, ..., C_m^i, C_{m+1}^i, C_{m+2}^i, ..., C_{m_p}^i\}$ (still denoted by C_i), where C_h^i $(h = m+1, ..., m_p)$ are randomly selected from $\{C_1^i, C_2^i, ..., C_m^i\}$. The sender-role sequence is generated in rounds and each round contains m_p time slots. A starting channel is randomly selected from C_i. Then, user i (sender) moves to the next channel index in C_i and keeps hopping on m_p channels in the round-robin fashion. Fig. 1(a) shows the sender-role sequence, where the starting channel C_k^i is randomly selected from C_i.

Receiver-role sequence (user j): Given $C_j = \{C_1^j, C_2^j, ..., C_n^j\}$, the receiver-role sequence is generated in rounds and each round contains n time slots. The sequence



(a) Sender-role sequence



(b) Receiver-role sequence

Fig. 1. CH sequence of CSAC.

in the first round could be any permutation of C_j. In the next round, all channel indices are left-shifted by 1 to generate a new permutation of C_j. The subsequences in the remaining rounds are determined in the same way. Fig. 1(b) shows the receiver-role sequence, where $\{C_{l_1}^j, C_{l_2}^j, ..., C_{l_n}^j\}$, i.e., the subsequence in the first round, could be any permutation of C_j.

The CSAC algorithm is formally presented as follows.

Algorithm 1: CSAC algorithm (for sender)

Require: user i's available channels $C_i = \{C_1^i, C_2^i, ..., C_m^i\}$
$m_p =$ the smallest prime number not smaller than m
Randomly select $m_p - m$ channels from C_i and denote these channels by $\{C_{m+1}^i, C_{m+2}^i, ..., C_{m_p}^i\}$
Randomly select an integer k in $[1, m_p]$
t=0
while not rendezvous **do**
  $t = t + 1$
  $T = (t - 1 + k)\% m_p$
  Attempt rendezvous on channel C_T^i
end while

Algorithm 2: CSAC algorithm (for receiver)

Require: user j's available channels $C_j = \{C_1^j, C_2^j, ..., C_n^j\}$
Randomly generate a permutation of $C_j : \{C_{l_1}^j, C_{l_2}^j, ..., C_{l_n}^j\}$
t=0
while not rendezvous **do**
  $t = t + 1$
  $T = (\lfloor \frac{t}{n} \rfloor \% n + t\% n)\% n + 1$
  Attempt rendezvous on channel $C_{l_T}^j$
end while

An illustration example: Available channel sets of user i and user j are $C_i = \{1, 2\}$ and $C_j = \{1, 3, 4\}$, respectively. That is, $m = m_p = 2$ and $n = 3$. Fig. 2 shows the CH sequences of the two users, where user i selects channel 2 as its starting channel and user j's sequence in the first round is $\{3, 4, 1\}$ which is a permutation of C_j. As shown in Fig.2, the offset between the two users' sequences could be 0 or 1. Users i and j can achieve rendezvous regardless of the offset between their sequences (see the time slots in gray color in Fig. 2).

B. Algorithm Analysis

As for our proposed CSAC algorithm, a crucial question is: does it guarantee two users performing CSAC to achieve rendezvous in finite time? In this section, we prove the

User *i*	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
User *j*		3	4	1	4	1	3	1	3	4	3	4	1	4	1	3	1	3	4

(a) Offset=1

User *i*	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	1
User *j*	3	4	1	4	1	3	1	3	4	3	4	1	4	1	3	1	3	4	3

(b) Offset=0

Fig. 2. Rendezvous of two users performing CSAC.

correctness of CSAC and derive its upper-bound on MTTR. We first present the following two lemmas.

Lemma IV.1: Given two positive integers x and y, and y is a prime number. The statement "x is not evenly divisible by y" is equivalent to that "x and y are co-prime".

Proof: $\Longrightarrow$Suppose z is a common factor of x and y. Since y is a prime number, z can be only 1 or y itself. However, if z is identical to y, then x must be evenly divisible by y, leading to a contradiction. Thus, x and y have no common factor other than 1, i.e., they are co-prime.

$\Longleftarrow$Suppose that x is evenly divisible by y, i.e., there exists z such that $x = zy$. Since y is a prime number, y is not equal to 1. Thus, 1 and y are both common factors of x and y, i.e., x and y are not co-prime, leading to a contradiction. So, x cannot be evenly divided by y. ■

Lemma IV.2: Given two positive integers x and y, and y is a prime number. If x and y are co-prime, then x^2 and y are also co-prime.

Proof: Since y is a prime number, according to Lemma IV.1, we can equivalently prove that x^2 is not evenly divisible by y. Suppose this is not true, i.e., there exists z such that $x \times x = z \times y$. This implies that y must be a factor of x since y is a prime number. That is, x and y are not co-prime, leading to a contradiction. Thus, we should have that x^2 is not evenly divisible by y, or equivalently that x^2 and y are co-prime. ■

Based on Lemma IV.1 and Lemma IV.2, we prove the correctness of CSAC and derive its upper-bound on MTTR in two cases: 1) n is not divisible by m_p (Theorem IV.1); 2) n is divisible by m_p (Theorem IV.2).

Theorem IV.1: If n is not divisible by m_p, two users performing the CSAC algorithm can achieve rendezvous in at most $n^2 m_p - nG + 1$ time slots, where m and n are the numbers of channels of the two users, m_p is the smallest prime number which is not smaller than m, and G is the number of commonly-available channels of the two users.

Proof: We consider rendezvous of user i (sender) and user j (receiver). Based on Fig. 1 which shows the general sequences of CSAC, we denote user i's sequence in round 1 by $S^i = \{C^i_k, C^i_{k+1}, ..., C^i_{m_p}, C^i_1, C^i_2, ..., C^i_{k-1}\}$ (with m_p items) and user j's sequence in the first n rounds by $S^j = \{C^j_{l_1}, C^j_{l_2}, ..., C^j_{l_n}, C^j_{l_2}, ..., C^j_{l_n}, C^j_{l_1}, ..., C^j_{l_n}, C^j_{l_1}, ..., C^j_{l_{n-1}}\}$ (with n^2 items). Furthermore, we use S^i_h (S^j_h) to denote the h-th item of S^i (S^j) ($h = 1, 2, ...$). Then, according to CSAC, at time slot t user i and user j actually hop on channel $S^i_{(t-1 \ mod \ m_p)+1}$ and channel $S^j_{(t-1 \ mod \ n^2)+1}$, respectively.

Since time-synchronization is not available, without loss of generality, we investigate rendezvous of the two users at a starting point in which users i and j are at time slots t^i_0 and t^j_0, respectively. Next, we prove that, for any pair of channels $S^i_h \in S^i$ and $S^j_g \in S^j$, there must be a same time slot at which users i and j hop on S^i_h and S^j_g, respectively. Notice that this result implies the guaranteed rendezvous of the two users since they have commonly-available channels (i.e., S^i and S^j share common items). To achieve this, we should prove that, for any $1 \le h \le m_p$ and $1 \le g \le n^2$, there exists t to satisfy the following equation (i.e., user i hops on channel S^i_h at its time slot $t + t^i_0$ and user j hops on channel S^j_g at its time slot $t + t^j_0$)

$$\begin{aligned} h - 1 &\equiv t + t^1_0 - 1 \ mod \ (m_p) \\ g - 1 &\equiv t + t^2_0 - 1 \ mod \ (n^2) \end{aligned} \tag{1}$$

which can be equivalently rewritten as follows

$$\begin{aligned} t &\equiv (h - t^1_0) \ mod \ (m_p) \\ t &\equiv (g - t^2_0) \ mod \ (n^2) \end{aligned} \tag{2}$$

Since m_p is a prime number and n is not evenly divisible by m_p, from Lemma IV.2, we know m_p and n^2 are co-prime. Therefore, from the Chinese Remainder Theorem [9], there exists an integer t that solves Equation (2). As mentioned earlier, this result implies the guaranteed rendezvous of the two users. We further derive an upper-bound of TTR as follows. Notice that there are $m_p n^2$ combinations of h and g values (i.e., $m_p n^2$ channel pairs of S^i_h and S^j_g). Hence, t does not exceed $m_p n^2$ regardless of h and g values. On the other hand, since the two users have G commonly-available channels (i.e., S^i and S^j share G common items) and in S^j a same channel (say $C^j_{l_1}$) exactly appears n times, there are Gn pairs of commonly-available channels among all the $m_p n^2$ pairs of S^i_h and S^j_g. Therefore, rendezvous will occur in at most $m_p n^2 - nG + 1$ time slots. ■

We consider the next case that n is divisible by m_p.

Theorem IV.2: If n is divisible by m_p, two users performing the CSAC algorithm can achieve rendezvous in at most $(nm_p - G + 1)$ times lots, where m and n are the numbers of channels of the two users, m_p is the smallest prime number which is not smaller than m, and G is the number of commonly-available channels of the two users.

Proof: We consider rendezvous of user i (sender) and user j (receiver) from a starting point that both users have implemented their respective CH sequence. Since time-synchronization is not available, user i and user j may be at time slots t^i_0 and t^j_0, respectively. Then, we investigate any consecutive time-span of $n \times m_p$ time slots from this starting point.

We prove that, any pair of channels C^i_h in $\{C^i_1, C^i_2, ..., C^i_m, C^i_{m+1}, ..., C^i_{m_p}\}$ and C^j_g in $\{C^j_{l_1}, C^j_{l_2}, ..., C^j_{l_n}\}$ appears once and only once in the time-span. That is, there exists exactly one t ($0 < t \le n \times m_p$) such that user i hops on channel C^i_h at its time slot $t^i_0 + t$ and user j hops on channel C^j_g at its time slot $t^j_0 + t$. Next we prove it by contradiction.

Suppose that a pair of channels, say C_h^i and C_g^j appears more than one time. This implies that there exist two different t_1 and t_2 such that:

user i hops on channel C_h^i at time slot $t_0^i + t_1$,
user j hops on channel C_g^i at time slot $t_0^i + t_1$,

user i hops on channel C_h^i at time slot $t_0^i + t_2$,
user j hops on channel C_g^i at time slot $t_0^i + t_2$,

According to our CSAC algorithm, we must have (suppose that $t_2 > t_1$ for convenience):

$$1)\quad t_2 = t_1 + \mu \times m_p \tag{3}$$

Sender-role hops on the m_p channels in round-robin fashion. The channel will appear after each m_p time slots. Each channel of sender will appear for n times in any consecutive $n \times m_p$ time slots. $1 \leq \mu < n$.

$$2)\quad t_2 = t_1 + a \times (n-1) + b \times n \tag{4}$$

In each round of receiver, all channel indices are left-shifted by 1 from the previous round. The channel will appear after each $(n-1)$ time slots. However, when the channel is the first one in current round, it will appear after $(n-1)+n$ time slots (the third and fourth channel 1 in Fig. 1 (a)). Each channel of receiver will appear for m_p times in any consecutive $n \times m_p$ time slots. $1 \leq a < m_p$ and $b \in \{0, 1\}$.

Using Equations 1) and 2), and the fact that $n = \delta \times m_p$ (n is evenly divisible by m_p), we should have

$$\mu \times m_p = a \times (\delta \times m_p - 1) + b \times \delta \times m_p \tag{5}$$

$$a = ((a+b) \times \delta - \mu) \times m_p \tag{6}$$

That is, a should be divisible by m_p, which contradicts the requirement $1 \leq a < m_p$. So any pair of channels C_h^i and C_g^j in any consecutive time-span of $n \times m_p$ time slots appears once and only once.

This result implies the guaranteed rendezvous of the two users. Notice that there are $m_p \times n$ combinations of h and g values (i.e., $m_p n$ channel pairs of S_h^i and S_g^j) in at most $m_p n$ time slots. Hence, the combinations in $m_p n$ time slots are different to each other. We further derive an upper-bound of TTR as follows. If there are G commonly-available channels between two users, there are G pairs of commonly-available channels among all the $m_p n$ pairs of S_h^i and S_g^j. The worst case is that the whole G times of rendezvous occur in the last G time slots of channel hopping sequence. Therefore, rendezvous will occur in at most $m_p n - G + 1$ time slots. In other word, $TTR \leq m_p n - G + 1$. ■

V. Simulation

We built a simulator in Visual Studio 2010 to evaluate the performance of our proposed CSAC algorithm. We select random algorithm [10], Jump-Stay [5] and CRSEQ [8] as the baseline algorithms for comparison. We introduce a parameter θ ($0 < \theta < 1$) to control the ratio of the number of available channels to that of the all channels. Available channels are randomly selected from the whole channel set such that the average number of available channels is equal to θQ, where Q is the number of all potential channels. We let θ vary from 0.1 to 0.3 since the available channel set is usually a small portion of the whole channel set [3]. Besides θ, we are concerned with parameter G, i.e., the number of commonly-available channels of the two users involved in the rendezvous. For each θ, we let G properly vary in $[1, \theta Q]$. For each combination of parameter values, we perform 10,000,000 independent runs and compute average TTR and maximum TTR (MTTR) accordingly. We report the simulation results in two different scenarios: I) n is divisible by m_p and II) n is not divisible by m_p. n is the number of available channels of the receiver, and m_p is the smallest prime number which is not smaller than the number of available channels of the sender.

A. Scenario I: n is divisible by m_p

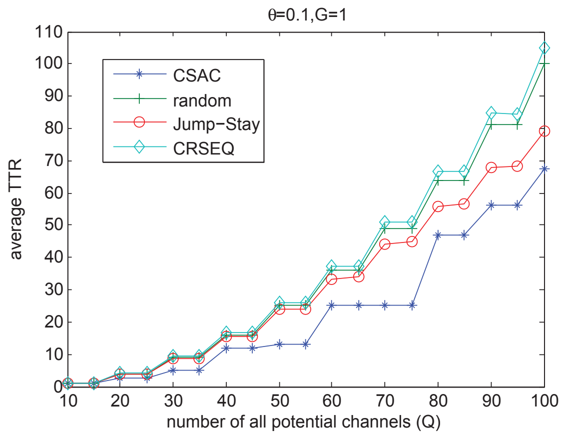
For the sake of convenience and space saving, we focus on $n = m_p$ in this scenario. Fig. 3 shows both the average TTRs and maximum TTRs of different algorithms against the number of all potential channels (i.e., Q) with $\theta = 0.1$ and $G = 1$. Since there is only one commonly-available channel ($G = 1$), such case is believed to be hard for quick rendezvous. According to Fig. 3, our CSAC algorithm significantly outperforms other algorithms in terms of both average TTR and maximum TTR. For example, when $Q = 60$, CSAC, random algorithm, Jump-Stay and CRSEQ respectively give average TTR of 25.29, 35.99, 33.42 and 37.44, and respectively give maximum TTR of 75, 643, 656 and 710. Fig. 4 shows the performance of different algorithms when $\theta = 0.3$ and $G = 1$.

Both Fig. 3 and Fig. 4 indicate that CSAC gives significantly lower maximum TTR than other algorithms. Furthermore, with increased number of all potential channels, CSAC shows slower increasing trend in terms of maximum TTR than other algorithms. However, in terms of average TTR, CSAC is not always the best, and it may be inferior to Jump-Stay, as shown in Fig. 4(a). Actually, in the simulation we observed that, CSAC gives its worst TTR (approaching MTTR) with a high probability, while the worst TTRs of other algorithms occur with relatively low probabilities. That is, CSAC has a small deviation of TTR and very often its TTR is close to its maximum TTR in the simulation. As a result, CSAC may have larger average TTR than other algorithms (e.g., Jump-Stay).

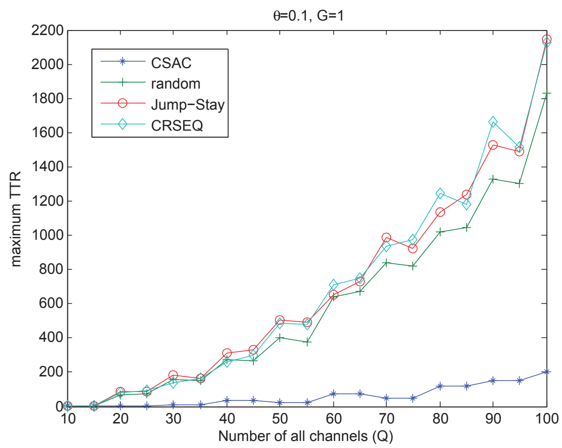
To further verify CSAC's attractive performance in terms of maximum TTR, we fix $Q = 60$ and $\theta = 0.3$ but vary G from 1 to $0.3Q$. Fig. 5 shows the corresponding simulation results. According to Fig. 5, we can again find that CSAC always gives smaller maximum TTR than other algorithms. Furthermore, as G increases (i.e., with more commonly-available channels), the maximum TTR of CSAC decreases, just like that of other algorithms. Such results are consistent with our previous theoretical analysis in Section IV.

B. Scenario II: n is not divisible by m_p

Fig. 6 shows the performance of different algorithms against the number of all potential channels with $\theta = 0.1$ and $G = 1$. It can be seen that CSAC performs best in terms of both

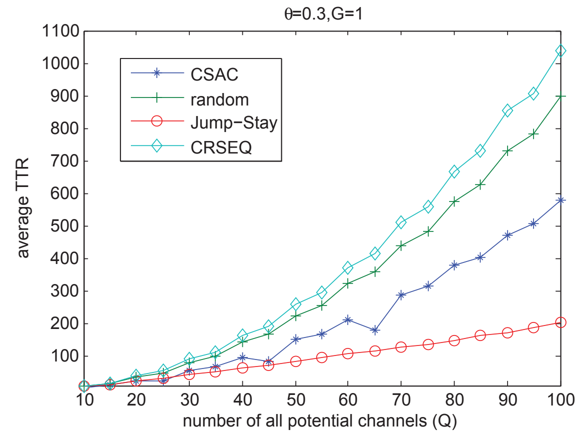


(a) Q versus Average TTR

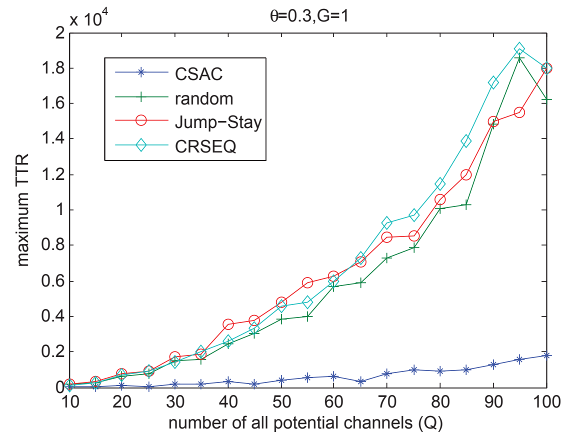


(b) Q versus Maximum TTR

Fig. 3. Performance of different algorithms when $\theta = 0.1$, $G = 1$.



(a) Q versus Average TTR



(b) Q versus Maximum TTR

Fig. 4. Performance of different algorithms when $\theta = 0.3$, $G = 1$.

average TTR and maximum TTR. For example, when $Q = 60$, the average TTRs of CSAC, random algorithm, Jump-Stay and CRSEQ are 11.18, 12.26, 11.97 and 12.86, respectively. When $Q = 100$, the maximum TTRs of CSAC, random algorithm, Jump-Stay and CRSEQ are 392, 1363, 1757 and 1529, respectively. We can see that CSAC shortens the MTTR up to 71.23%. Similar to Scenario I, in this scenario CSAC shows significant superiority to other algorithms in terms of maximum TTR. Such advantage of CSAC is further verified in Fig. 7 where we fix $Q = 60$ and $\theta = 0.1$ but vary G from 1 to $0.1Q$.

VI. Conclusion

We proposed a new algorithm named CSAC for the blind rendezvous in cognitive radio networks. CSAC is the first one in the literature that generates CH sequences based on the available channel set instead of the whole channel set while providing guaranteed rendezvous. We derived an upper-bound on the MTTR of CSAC, which is significantly smaller than those of the state-of-the-art when the available channel set is a small portion of the whole channel set. The efficiency of CSAC was further verified in the simulation. Compared with the existing solutions, CSAC can shorten the MTTR up to 71.23% when the total number of channels is 100 and the ratio of available channels is 10%. CSAC makes the first attempt towards a new direction in designing CH sequences based on the only available channel set. We believe that more rendezvous algorithms will be presented in this direction and the MTTR will be further improved.

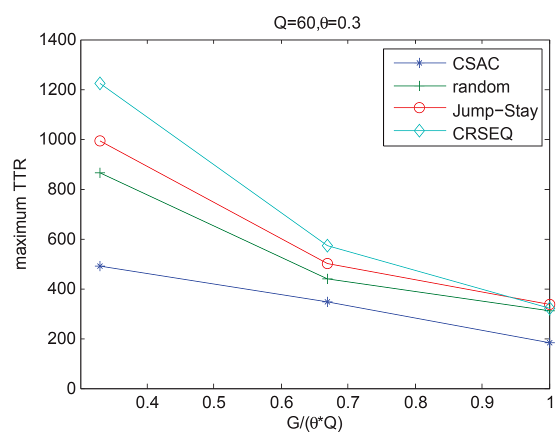
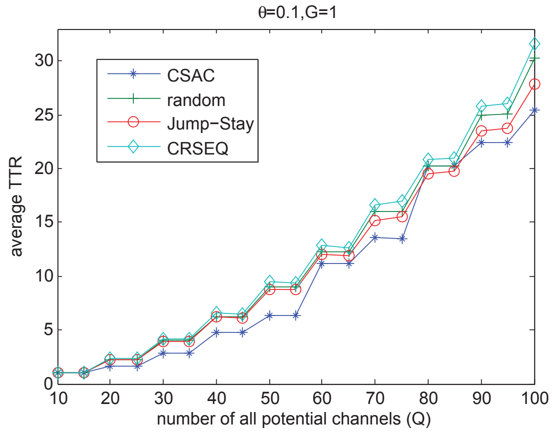


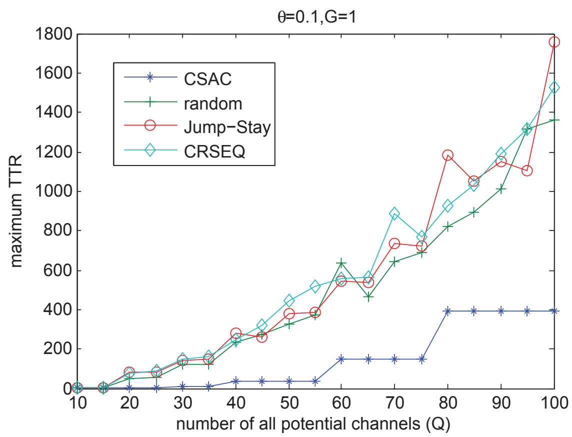
Fig. 5. Influence of G when $Q = 60$, $\theta = 0.3$.

Acknowledgment

This work is supported in part by Research Grant of Hong Kong [Project No. GRF211513] and Research Grant of Hong Kong Baptist University [Project No. FRG2/12-13/055]. Zhiyong Lin is supported by the National Natural Science Foundation of China (Grant No. 61202453) and the Natural Science Foundation of Guangdong Province in China (Grant



(a) Q versus Average TTR



(b) Q versus Maximum TTR

Fig. 6. Performance of different algorithms when $\theta = 0.1$, $G = 1$, $m_p \neq n$.

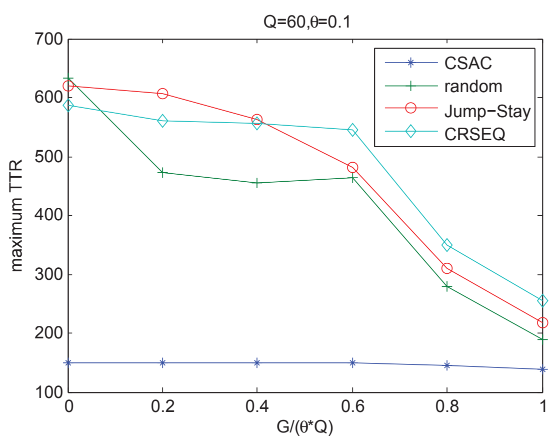


Fig. 7. Influence of G when $Q = 60$, $\theta = 0.1$.

No. S2011040002890).

REFERENCES

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty, "NeXt Generation Dynamic Spectrum Access Cognitive Radio Wireless Networks: A Survey," *Computer Networks*, vol. 50, no. 13, pp. 2127-2159, 2006.

[2] T. Ycek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 11617160, 2009.

[3] P. Wang, L. Xiao, S. Zhou, and J. Wang, "Optimization of detection time for channel efficiency in cognitive radio systems," *Proc. of IEEE WCNC*, pp.11117115, Mar. 2007.

[4] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Taxonomy and Challenges of Rendezvous Algorithms in Cognitive Radio Networks," invited position paper, *Proc. of IEEE ICNC*, pp. 645-649, Hawaii, 2012.

[5] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Jump-Stay Rendezvous Algorithm for Cognitive Radio Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1867-1881, 2012.

[6] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," *Proc. of IEEE INFOCOM*, pp. 2444-2452, Apr. 2011.

[7] 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.

[8] J. Shin, D. Yang, and C. Kim, "A Channel Rendezvous Scheme for Cognitive Radio Networks," *IEEE Communications Letters*, vol. 14, no. 10, pp. 954-956, 2010.

[9] 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, 2010.

[10] K. Bian, J.-M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 68917703, 2011.

[11] V. Brik, E. Rozner, S. Banerjee, and P. Bahl, "DSAP: A Protocol for Coordinated Spectrum Access," *Proc. of IEEE DySPAN*, pp. 611-614, Nov. 2005.

[12] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "DIMSUMnet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum," *Proc. of IEEE WoWMoM*, pp. 78-85, Jun. 2005.

[13] J. Jia, Q. Zhang, and X. Shen, "HC-MAC: A Hardware-constrained Cognitive MAC for Efficient Spectrum Management," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 106-117, 2008.

[14] L. Ma, X. Han, and C.-C. Shen, "Dynamic Open Spectrum Sharing for Wireless Ad Hoc Networks," *Proc. of IEEE DySPAN*, pp. 203-213, Nov. 2005.

[15] J. Zhao, H. Zheng and G.-H. Yang, "Distributed Coordination in Dynamic Spectrum Allocation Networks," *Proc. of IEEE DySPAN*, pp. 259-268, Nov. 2005.

[16] L. Lazos, S. Liu and M. Krunz, "Spectrum Opportunity-based Control Channel Assignment in Cognitive Radio Networks," *Proc. of IEEE SECON*, pp. 1-9, Jun. 2009.

[17] D. Yang, J. Shin, and C. Kim, "Deterministic Rendezvous Scheme in Multichannel Access Networks," *Electronics Letters*, vol. 46, no. 20, pp. 1402-1404, 2010.

[18] 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.

[19] K. Bian, J.-M. Park, and R. Chen, "A Quorum-based Framework for Establishing Control Channels in Dynamic Spectrum Access Networks," *Proc. of ACM MobiCom*, pp. 25-36, Sept. 2009.

[20] C. Cormio and K. R. Chowdhury, "Common Control Channel Design for Cognitive Radio Wireless Ad Hoc Networks Using Adaptive Frequency Hopping," *Ad Hoc Networks*, Vol. 8, pp. 430-438, 2010.

[21] Y. Zhang et al., "ETCH: Efficient Channel Hopping for Communication Rendezvous in Dynamic Spectrum Access Networks," *Proc. of IEEE INFOCOM*, pp. 2471-2479, Apr. 2011.

[22] S. Romaszko, "A Rendezvous Protocol with the Heterogeneous Spectrum Availability Analysis for Cognitive Radio Ad Hoc Networks," *Hindawi Publishing Corporation Journal of Electrical and Computer Engineering*, Article ID 715816. 2013.

[23] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Ring-Walk Based Channel-Hopping Algorithm with Guaranteed Rendezvous for Cognitive Radio Networks," *Proc. of IEEE/ACM International Conference on Green Computing and Communications*, pp. 755-760, Dec. 2010.

[24] C. Cordeiro and K. Challapali, "C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks," *Proc. of IEEE DySPAN*, pp. 147-157, Apr. 2007.

[25] L. A. DaSilva and I. Guerreiro, "Sequence-Based Rendezvous for Dynamic Spectrum Access," *Proc. of IEEE DySPAN*, pp. 1-7, Oct. 2008.

[26] Z. Lin, H. Liu, X. Chu, Y. W. Leung, "Ring-Walk Rendezvous Algorithms for Cognitive Radio Networks," *Ad Hoc & Sensor Wireless Networks*, vol. 16, no. 4, pp. 243-271, 2012.