Dynamic Rendezvous Algorithms For Cognitive Radio Networks

Haosen Pu*, Zhaoquan Gu*, Xiao Lin*, Qiang-Sheng Hua†, Hai Jin†
*Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing, P.R. China
†Services Computing Technology and System Lab/Cluster and Grid Computing Lab,
School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, P.R. China

Abstract*—Rendezvous is a fundamental process in constructing cognitive radio networks (CRNs), in which two users find a common channel for communication. The licensed spectrum is assumed to be divided into n non-overlapping channels and the users can sense the spectrum by equipping with cognitive radios. Most of previous works assume that the user can find a set of available channels (the channels not occupied by the licensed users) after spectrum sensing stage and the status of all channels are stable all the time. However, this assumption may not be true in reality and we focus on designing efficient algorithms when the status of the channels varies dynamically. In this paper, we introduce two models to describe the dynamic rendezvous problem. Denote p_j^i as the probability that channel j is available for user i. In the Independent model, assuming all p_j^i variables are independently distributed and we propose efficient algorithms for both synchronous and asynchronous users, which guarantee rendezvous in $O(\log^2 n)$ and $O(\log^3 n)$ time slots with high probability respectively. In the Dependent model, two nearby users have relevant available probabilities and we introduce a sensing phase and an attempting phase to guarantee rendezvous in $O(\varepsilon \log^3 n \log\log\log n)$ time slots with high probability, where ε is a small constant. We also present an algorithm to increase rendezvous load in the long run, which guarantee rendezvous for at least $\frac{1}{4}$ of all time slots.

I. INTRODUCTION

Wireless spectrum has been a scarce resource due to the exponential growth of wireless devices and applications. However, with the traditional spectrum management, a majority of the licensed spectrum is underutilized (according to the FCC, the utilization rate varies from 15% to 85% [1]) while the unlicensed spectrum is overcrowded [3]. Dynamic spectrum access (DSA) technique has been proposed to construct cognitive radio networks (CRNs) to alleviate the spectrum scarcity problem in wireless communications [11]. A CRN consists of primary users (PUs) that own the licensed spectrum and secondary users (SUs) that are allowed to share the licensed spectrum with PUs. Each SU is equipped with one or more cognitive radios to sense the free licensed spectrum [9], [14] and it can access the available channels without causing interference to PUs. Unless otherwise specified, "users" hereafter refer to SUs.

The users in CRN need to communicate with each other for information exchange and so on. Rendezvous is the fundamental process in constructing the CRN, through which the users can establish a link on some common channel for communication. Generally speaking, the users firstly sense the licensed spectrum to find available channels which means the channels are not occupied by nearby PUs, and then they can access these channels for communication attempt. Once two users access the same channel at the same time slot, a link can be constructed for communication and we say they *rendezvous* on the common channel. *Time to rendezvous* (TTR) is used to measure the time cost to choose a common channel and *maximum time to rendezvous* $(MTTR)$ denotes the rendezvous time of the worst case, where two users may start the rendezvous process at different time slots and they may have different available channels.

To the best of our knowledge, all existing rendezvous algorithms implicitly assume the available channels for the users are fixed after the spectrum sensing stage [2], [6], [8], [10], [13]. But in practical situations, the condition of licensed spectrum may vary at any time due to the activities of PUs. For example, the PUs can watch TV freely and change to different TV channels to reclaim the different licensed spectrum in the TV white space. Since the PUs have priority for the licensed spectrum, once some PU occupies a licensed channel, the SUs have to vacate the channel immediately even it is used for information exchange. Therefore, the available channels sensed by the SUs may vary over time.

In this paper, we focus on designing channel selection strategies when the available channels for the users are not stable. Technically, the licensed spectrum is assumed to be divided into n non-overlapping channels as $U = \{1, 2, \cdots, n\}$, where the indices are known to all users. Supposing time is divided into slots of equal length $2t_0$, where t_0 is the sufficient time to establish a link if two users access the same channel at the same time. In each time slot, a licensed channel $j \in U$ is available for user a (or b) with probability p_j^a (or p_j^b). In this paper, we propose two theoretical models to describe the dynamic setting:

1) *Independent Model*: $p_j^a, p_j^b (1 \le j \le n)$ are independently distributed ;
2) *Dependent Model*: p_j^a $(1 \le j \le n)$ is independently distributed. But as two users can communicate with each other, they are dispersed in nearby residences and

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

we assume p_j^b has some dependency on p_j^a such that $|p_j^a - p_j^b| \leq \varepsilon_1$ where ε_1 is a small constant.

To the best of our knowledge, this is the first work designing rendezvous algorithms when the channels' status varies dynamically. The main contributions of this paper are summarized as follows.

1) For the Independent model , we propose randomized algorithms for both synchronous and asynchronous settings such that when $p_j^i, i \in \{a, b\}, j \in [1, n]$ obey the uniform distribution, two synchronous users can achieve rendezvous in $O(\log^2 n)$ time slots with high probability (w.h.p), and two asynchronous users can achieve rendezvous in $O(\log^3 n)$ time slots w.h.p.;
2) For the Dependent model, we propose a distributed algorithm which guarantees rendezvous in $O(\varepsilon \log^3 n \log\log\log n)$ time slots w.h.p. when p_j^a ($j \in [1, n]$) obey the uniform distribution and ε is a small constant;
3) We propose an algorithm to increase rendezvous load such that the users can communicate in at least $\frac{T}{4}$ of the time slots in a long run where T is the length of time since the users have already begun the algorithm. Due to the page limit, we put this part in the full version [12].
4) We conduct extensive simulations to evaluate our algorithms under different circumstances and the results corroborate our analyses.

Due to the page limit, all the proofs of theorems and corollaries will be put in the full version [12].

II. Related Work

There are mainly two types of rendezvous algorithms: centralized algorithms and distributed algorithms. In extant centralized rendezvous algorithms, the users either utilize a central controller or employ a dedicated common control channel(CCC). Although centralized algorithms can simplify the rendezvous process, they have several disadvantages: the difficulty of establishing CCC, the low scalability and inflexibility, and the vulnerability to jamming attack.

Therefore, the distributed rendezvous algorithms are more preferable in practice. The main technique is channel hopping(CH), which means each user can generate a specific sequence and hops among the available channels according to it. Generally speaking, there are mainly two types of the constructed sequences: global sequence (GS) [6], [10], [13] is constructed based on all channels' information and local sequence (LS) [4], [5], [7] is generated on the basis of the users' local information, such as the available channels and the distinct identifer.

However, all the existing algorithms implicitly assume the available channels for each user is fixed after the spectrum sensing stage, which is unrealistic. In our paper, we initiate the study of rendezvous schemes when the channels' status varies dynamically.

III. Preliminaries

A. System Model

The licensed spectrum is divided into n non-overlapping channels as $U = \{1, 2, \ldots, n\}$(we assume that n is known for all users) and the users are equipped with cognitive radios to sense and access these channels. After the spectrum sensing stage, each user i finds out a set of available channels $C_i \subseteq U$, where a channel $j \in C_i$ is *available* if it is not occupied by any nearby PUs. Assuming time is divided into time slots with equal length $2t_0$, where t_0 is the sufficient time to construct a link for communication if they choose the same channel at the same time. In each slot, the user selects an available channel for rendezvous attempt. If two users choose the same channel in the same time slot, a communication link between them can be established and we call this *rendezvous*. Generally speaking, there are three different settings:

1) **Anonymous** & **Non-anonymous**: In an anonymous network, the users do not have unique identifiers (IDs) and they cannot be distinguished from each other. Therefore, all users have to run the same algorithm. However, different users in a non-anonymous network have distinct IDs and they may choose different strategies to rendezvous.

2) **Synchronous** & **Asynchronous**: If two users start the rendezvous algorithm at the same time, we call them synchronous users. Otherwise, they are asynchronous users. Note that, the time slots for two users may be not aligned and we can transform the situation into slot-aligned scenario by setting the length $2t_0$ as described above [7].

3) **Static** & **Dynamic**: All previous rendezvous algorithms assume the available channels for the users are fixed after they begin the rendezvous process, which we call static setting. In reality, the channel status may vary over time and the available channel set could be different in different time slots. We call this dynamic setting.

In this paper, we consider the rendezvous problem under **anonymous, asynchronous and dynamic** settings. For two users a and b, supposing channel $j \in U$ is available for user $i \in \{a, b\}$ at time slot t with probability $p_j^i(t)$. For simplicity, we assume that the PUs' usage on a licensed channel is stable and thus the probability channel j is available for user i in different time slots is fixed (i.e. $\forall t_1, t_2, i \in \{a, b\}$, $p_j^i(t_1) = p_j^i(t_2) = p_j^i$). According to different relationships between p_j^a and p_j^b, we propose two models to describe the dynamic setting:

1) **Independent Model:** All p_j^i ($i \in \{a, b\}, 1 \leq j \leq n$) are independently distributed;

2) **Dependent Model:** For channel $1 \leq j \leq n$, $|p_j^a - p_j^b| \leq \varepsilon_1$ where ε_1 is a small constant. Moreover, we assume the probabilities p_j^a ($1 \leq j \leq n$) are independently distributed.

Dependent model is more practical to depict the reality. Two users can communicate with each other, implying they are close in a geographic area, and thus a PU who influences one user is likely to influence the other. So we use $|p_j^a - p_j^b| \leq \varepsilon_1$ ($1 \leq j \leq n$) to depict the dependence.

Different from all previous rendezvous algorithms, the spectrum sensing stage is not necessary in our design and we do not need to know the available channel set before starting the rendezvous algorithm. In each time slot, our algorithms choose a licensed channel and try to access it if it is available. Otherwise, the user finds out that the channel is occupied by some PU and this time slot is just wasted. Time to rendezvous (TTR) denotes the time slots cost to achieve rendezvous and we use maximum time to rendezvous ($MTTR$) to evaluate the worst situation when the users are asynchronous. Except for TTR, we also use *rendezvous load* as another metric which represents the proportion of time that the users have achieved during a long time T. In dynamic setting, the users cannot always keep communication after rendezvous due to the dynamic occupation of the licensed channels, and rendezvous load is also an important metric.

B. *Probability Techniques*

We also use other probability techniques such as Cumulative function, Bayes Theorem and Chernoff Bound in our work, the details can be found in the full version [12].

IV. Rendezvous Algorithms for the Independent Model

In this section, we propose efficient randomized algorithms for both synchronous and asynchronous settings under the Independent model. In our analysis, we assume that all p_j^i ($i \in \{a, b\}, 1 \le j \le n$) are i.i.d. and they obey the uniform distribution in $[0, 1]$.

A. *Rendezvous Algorithm for Synchronous Users*

Algorithm 1 Synchronized algorithm for the Independent Model

1: $j = 1$, $t = L = 6\lceil \log n \rceil$;
2: **while** Not rendezvous **do**
3:    Try to access channel j;
4:    $t = t - 1$;
5:    **if** $t = 0$ **then**
6:       $t = L$;
7:       $j = j\%L + 1$;
8:    **end if**
9: **end while**

We describe the synchronous algorithm in Alg. 1. First, we divide time into rounds where each round contains $L = 6\lceil \log n \rceil$ time slots. In the k-th round, the user tries to access channel $j = (k - 1)\%L + 1$ as Line 3. If two users have a common available channel in the first L rounds, rendezvous can be guaranteed.

Theorem 1: Alg. 1 guarantees rendezvous for two synchronous users under the Independent model in $MTTR = O(\log^2 n)$ time slots w.h.p..

Algorithm 2 Asynchronous Algorithm for the Independent Model

1: $t = 1$, $M = 3\lceil \log n \rceil$;
2: Choose first M channels as $V = \{1, 2, \ldots, M\} \subseteq U$;
3: Construct a sequence $S = \{c_1, c_2, \ldots, c_L\}$ based on V using the DRDS method [6], where $L = 3P^2$ (P is the smallest prime number such that $P \ge M$);
4: **while** Not rendezvous **do**
5:    $k = (t - 1)\%L + 1$;
6:    Try to access channel $c_k \in S$;
7:    $t = t + 1$;
8: **end while**

B. *Rendezvous Algorithm for Asynchronous Users*

Asynchronous users start the rendezvous algorithm in different time slots and we propose an efficient randomized algorithm based on the DRDS method [6].

As described in Alg. 2, we pick the first $M = 3\lceil \log n \rceil$ channels to compose set $V = \{1, 2, \ldots, M\} \subseteq U$ and then constructs a rendezvous sequence S of length $L = 3P^2$ (P is the smallest prime number no less than M) based on the DRDS method [6] as Line 3. After the construction, the user tries to access the channels by repeating the sequence every L time slots until rendezvous. There is one important property of S based on the DRDS method [6].

Property 4.1: $\forall \delta \ge 0$, $\forall j \in [1, M]$, there exists $1 \le T \le L$ such that $c_{(T-1)\%L+1} = c_{(T+\delta-1)\%L+1} = j$.

We use the property to derive the time complexity to rendezvous.

Theorem 2: Alg. 2 guarantees rendezvous for two asynchronous users under the Independent model in $MTTR = O(\log^3 n)$ time slots w.h.p..

V. Rendezvous Algorithm for the Dependent Model

In the dependent model, we assume the probability that channel $j \in [1, n]$ is available for user a (p_j^a) is independently distributed, but since two users are close to each other, the probability that channel $j \in [1, n]$ is available for user b (p_j^b) satisfying $|p_j^a - p_j^b| \le \varepsilon_1$(this assumption is reasonable because nearby SUs may be interfered by same PUs with high probability), where ε_1 is a small constant. In this section, we propose a randomized algorithm consisting of a sensing phase and an attempting phase to guarantee rendezvous in a short time with high probability. In the section, we also assume that p_j^a obeys uniform distribution from $[0, 1]$.

A. *Sensing Phase for the Dependent Model*

For user i, we are only aware of the cumulative distribution function of the channels' available probability and we do not have the exact value for practical rendezvous attempt. Hence we try to find the channels that are available with high probability. As shown in Alg. 3, we consider the first M channels and the user tries to access each channel P times. According to the response (access successfully or not), we can

Algorithm 3 Sensing Phase for User i

```
1:  M = 2⌈log n⌉, P = ⌈(9/ε_2^2) log n⌉, t = 1;
2:  ∀j ∈ [1, M], P_j = 0;
3:  while t ≤ MP do
4:      Try to access channel j = (t − 1)%M + 1;
5:      if ACCESS SUCCESSFULLY then
6:          P_j = P_j + 1;
7:      end if
8:      t = t + 1;
9:  end while
10: ∀1 ≤ j ≤ M, p̃^i_j = P_j/P;
```

figure out an estimation of the probabilities that these channels are available as Line 10. The values of the parameters (M,P) are designed to guarantee the accuracy of our estimation of the probabilities, which play an important role in the attempting phase.

B. Attempting Phase for the Dependent Model

After the sensing phase, we have the estimation values of the channels' available probabilities, and our algorithm works on the basis of them. Before introducing the attempting phase, we present a rendezvous sequence construction for a special situation where each user has exactly two available channels and there is at least one common channel.

Supposing two available channels are $v_1, v_2 \in U$ ($v_1 < v_2$), we describe the sequence construction in Alg. 4.

Algorithm 4 Rendezvous Sequence Construction

```
1:  l_1 = ⌈log n⌉, l_2 = ⌈log l_1⌉;
2:  Find c ∈ [1, l_1] such that v_1's c-th bit is 0 and v_2's c-th
    bit is 1;
3:  D⃗ = {∗, c_{l_2}, c_{l_2−1}, ..., c_1} where (c_{l_2}, c_{l_2−1}, ..., c_1) is
    the binary representation of c;
4:  Denote the rendezvous sequence S = ∅;
5:  for r = 1 : l_2 + 1 do
6:      If D⃗(r) = ∗, add s_∗ = (v_1, v_1, v_1, v_2, v_2, v_2) twice to
        S;
7:      If D⃗(r) = 0, add s_0 = (v_1, v_1, v_2, v_2, v_1, v_2) twice to
        S;
8:      If D⃗(r) = 1, add s_1 = (v_1, v_1, v_2, v_1, v_2, v_2) twice to
        S;
9:  end for
10: Repeat S twice to get the rendezvous sequence S;
```

Alg. 4 first finds a number $c \in [1, l_1]$ such that the c-th bit of v_2 is 1 and the corresponding bit of v_1 is 0 (since $v_1 < v_2$, such number exists). Then Alg. 4 constructs a vector $\overrightarrow{D}$ by adding the symbol $*$ to the binary representation of c as Line 3. S is then generated in l_2+1 round where $l_2 = \lceil \log\log n\rceil$. In each round, 12 numbers are added to the sequence according to the r-th value of $\overrightarrow{D}$. After the construction, the rendezvous sequence S is repeated twice. Obviously, the length of the constructed sequence is $|S| = 24(\lceil \log\log n\rceil + 1)$. If $v_1 = v_2$,

Algorithm 5 Attempting Phase for User i

```
1:  P̃^i_max = max{p̃^i_1, p̃^i_2, ..., p̃^i_M}, S_i = ∅;
2:  for 1 ≤ j ≤ M do
3:      if p̃^i_j ≥ P̃^i_Max − ε_1 − 2ε_2 then
4:          S_i = S_i ⋃{j};
5:      end if
6:  end for
7:  Order the channels in S_i by increasing order as S_i =
    {c_{i,1}, c_{i,2}, ..., c_{i,m}} where m = |S_i|;
8:  Let p, q be the smallest two prime numbers larger than m;
9:  L = 0, r = 1, t = 1;
10: while Not rendezvous do
11:     if L = 0 then
12:         s_{i,r} = (r − 1)%p + 1, v_{i,r} = (r − 1)%q + 1;
13:         Invoke Alg. 4 to construct sequence seq_r based on
            two channels s_{i,r}, v_{i,r};
14:         Let L be the length of seq_r;
15:         r = r + 1;
16:     end if
17:     Try to access the t-th channel of seq_r as seq_{r,t};
18:     L = L − 1; t = t + 1;
19: end while
```

we construct the sequence by repeating channel v_1 for $|S|$ time slots.

Lemma 5.1: For users a and b with two available channels $V_1 = \{a_1, a_2\}$ and $V_2 = \{b_1, b_2\}$, if $V_1 \cap V_2 \neq \emptyset$ and the overlapping length of their sequences is at least $\frac{|S|}{2}$, rendezvous can be guaranteed in $\frac{|S|}{2} = 12(\lceil \log\log n\rceil + 1)$ time slots.

After the sensing phase, we present the attempting phase in Alg. 5. In the attempting phase, we first construct the set (S_i) of the channels which are more likely to be available based on the estimation values of the sensing phase as Line 2-6 (ε_2 is a constant and we determine the value in our analysis part). Then two prime numbers are picked as p, q and two channels are chosen based on the primes as Line 12. The rendezvous sequence seq_r is constructed by invoking Alg. 4 and the length of seq_r is $L = 24(\lceil \log\log M\rceil + 1)$ (since two channels are both from $[1, M]$). After every L time slots, another two channels are chosen and a new sequence is constructed until rendezvous.

For two users with chosen sets S_1 and S_2 in the attempting phase, rendezvous is guaranteed if the channel sets intersect. We derive the time to rendezvous as Lemma 5.2.

Lemma 5.2: For any channel $j \in S_1 \cap S_2$, two users can rendezvous on channel j in $O(|S_1||S_2| \log\log M)$ time slots.

C. Algorithm Performance Analysis

In order to show the correctness of our algorithm, we assume $|\widetilde{p}^i_j - p^i_j| \leq \varepsilon_2$ holds for $i \in \{a, b\}, 1 \leq j \leq M$ after the sensing phase, and then our algorithm can guarantee rendezvous under this assumption.

Lemma 5.3: If $|\widetilde{p}^i_j - p^i_j| \leq \varepsilon_2$ for $i \in \{a, b\}, 1 \leq j \leq M$, our algorithm can guarantee rendezvous between the users.

We show that this assumption holds with high probability:

Lemma 5.4: $|\widetilde{p}_j^i - p_j^i| \le \varepsilon_2$ holds for all $i \in \{a, b\}, 1 \le j \le M$ with probability at least $1 - \frac{1}{n^2}$,

Combining Lemma 5.3-5.4, our algorithm can guarantee rendezvous w.h.p.. From Lemma 5.2, the rendezvous time is bounded in $O(|S_1||S_2|\log\log M)$ time slots. In order to derive the detailed time complexity by estimating the size of S_1 and S_2, we introduce Lemma 5.5.

Lemma 5.5: Denote the following variables as:

$$j_1 = argmax_j\{\widetilde{p}_j^a | 1 \le j \le M\}, j_2 = argmax_j\{\widetilde{p}_j^b | 1 \le j \le M\}$$
$$j_1' = argmax_j\{p_j^a | 1 \le j \le M\}, j_2' = argmax_j\{p_j^b | 1 \le j \le M\}$$

and

$$S_1' = \{j | 1 \le j \le M, p_j^a \ge p_{j_1}^a - \varepsilon_1 - 4\varepsilon_2\}$$
$$S_2' = \{j | 1 \le j \le M, p_j^a \ge p_{j_1}^a - 3\varepsilon_1 - 4\varepsilon_2\}$$

then $S_1 \subseteq S_1'$ and $S_2 \subseteq S_2'$.

Then, we only need to estimate the size of S_1' and S_2' to derive the time complexity. Assume $F(x)$ is the cumulative probability function of p_j^i (for uniform distribution $F(x) = x$) and $G(x)$ is the cumulative probability function for $P_{max}^i = \max\{p_1^i, p_2^i, \ldots, p_M^i\}$, then

$$G(x) = P(P_{Max}^i \le x) = \prod_{j=1}^{M} P(p_j^i \le x) = F^M(x)$$

$$g(x) = \frac{d}{dx}G(x) = MF^{M-1}(x)$$

Let $Y = P_{max}^i$,

$$F(x|y) = \frac{\frac{\partial}{\partial y}F(x,y)}{f(y)} = \begin{cases} 1 & x \ge y \\ \frac{(M-1)F(x)}{MF(y)} & x < y \end{cases}$$

Thus, we can derive Lemma 5.6-5.7 and Corollary 1.

Lemma 5.6: Supposing random variables $\{x_1, x_2, \ldots, x_n\}$ are independent and identically distributed, and the cumulative probability function is $F(x)$, denote $Y = x_{max} = \max\{x_1, x_2, \ldots, x_n\}$, $S(\varepsilon) = \{j | x_j \ge x_{max} - \varepsilon\}$ and $\mu = \frac{F(y) - F(y-\varepsilon)}{F(y)}(n-1)$, then

$$P(|S(\varepsilon)| \ge 1 + (1 + \sqrt{\frac{3}{F(y) - F(y-\varepsilon)}})\mu | Y = y) \le e^{1-n}$$

Lemma 5.7: $\max_y(F(y) - F(y-\varepsilon)) \le \max_y(f(y))\varepsilon$

Since ε_1 is a small constant, we choose ε_2 such that $\varepsilon = 3\varepsilon_1 + 4\varepsilon_2 \le \frac{1}{4}$. Denote E_1 as the event that $P_{max}^i \le \frac{1}{2}$, then $P(E_1) \le F^M(\frac{1}{2}) \le \frac{1}{n^2}$. Denote E_2 as the event that two users do not rendezvous in a round, then $P(E_2|\overline{E_1}) \le \frac{1}{2} - (3\varepsilon_1 + 4\varepsilon_2) = \frac{1}{4}$. Denote E_3 as the event that $|S_1'|$ or $|S_2'|$ is larger than $\frac{\sqrt{3\max(F(y)-F(y-\varepsilon)}}{F(1/2)} \le \frac{\sqrt{3\max(f(y)\varepsilon}}{F(1/2)}$. Combining Lemma 5.2-5.7, we conclude the theorem as:

Corollary 1: If the i.i.d. variables p_j^a $(1 \le j \le n)$ obey the uniform distribution, our algorithm guarantees rendezvous for two users in $O(\varepsilon \log^3 n \log\log\log n)$ time slots w.h.p., where $\varepsilon = 3\varepsilon_1 + 4\varepsilon_2$.

TABLE I
$ETTR$ AND $MTTR$ COMPARISON OF ALG. 1

n	$ETTR$	$MTTR$
1000	4.01	28
2000	3.82	29
4000	4.09	30
8000	3.923	29
10000	4.094	32
20000	4.052	30

TABLE II
$ETTR$ AND $MTTR$ COMPARISON OF ALG. 2

n	$ETTR$	$MTTR$	$MTTR/\log^3 n$
1000	276.54	2411	2.436
2000	290.615	3185	2.415
4000	347.382	4042	2.359
8000	355.48	4968	2.279
10000	429.791	5463	2.329
20000	435.635	6548	2.245

From the above analysis, when ε_1 is small (for example $\varepsilon_1 = \frac{1}{\log n}$), the algorithm is much more efficient than algorithms for the Independent model. Moreover, when the user tries to rendezvous with multiple users, the sensing phase is only carried out once with time complexity $O(\frac{\log^2 n}{\varepsilon_2^2})$ and thus this algorithm's average time for rendezvous is dominated by the attempting phase's time.

VI. Simulation Results

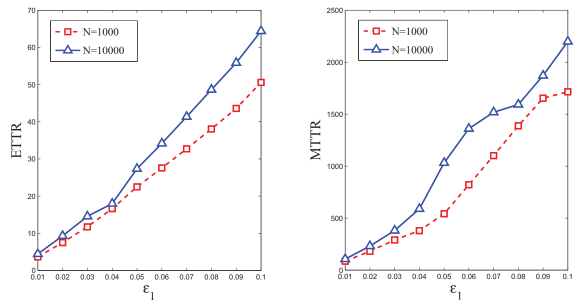
A. Evaluations for the Independent Model

In our simulation, all probabilities $p_j^i, i \in \{a, b\}, j \in [1, n]$ obey the uniform distribution in $[0, 1]$ in the Independent model. For two synchronous users, the simulation results of Alg. 1 is listed in Table I when the number of channels n increases from 1000 to 20000. As a surprising result, both the $ETTR$ and $MTTR$ values don't change evidently when n increases and it is extraordinarily small. The results show that our proposed algorithm works even better in practical case than that in theoretical analysis.

For two asynchronous users under the Independent model, Table II shows the $ETTR$ and $MTTR$ values when the number of channels n increases from 1000 to 20000. From the table, both $ETTR$ and $MTTR$ values increase when n gets larger, and the $MTTR$ values are linearly increasing with $\log^3 n$, which corroborates our theoretical analysis (from the table, the ratio of $MTTR$ and $\log^3 n$ is bounded by 2.5).

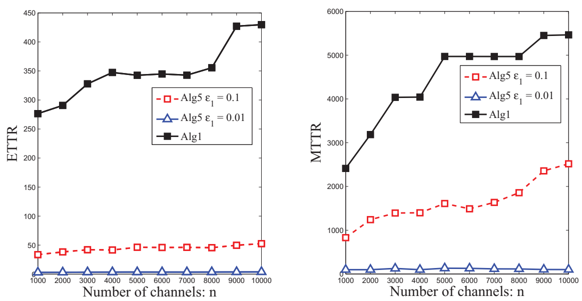
B. Evaluation for the Dependent Model

In order to evaluate the efficiency of our algorithm for the rendezvous problem under the Dependent model, we first evaluate the performance of Alg. 5 (attempting phase) based on uniform distribution. Moreover, we compare the algorithm under the Dependent model with the asynchronous algorithm (Alg. 2) under the Independent model for the uniform distribution to demonstrate the efficiency.



(a) $ETTR$ vs. ε_1 (b) $MTTR$ vs. ε_1

Fig. 1. $ETTR$ and $MTTR$ comparison for Alg. 5 when $n = 1000$ and $n = 10000$ under uniform distribution



(a) $ETTR$ vs. n (b) $MTTR$ vs. n

Fig. 2. $ETTR$ and $MTTR$ comparison for Alg. 2 and Alg. 5 when $\varepsilon_1 = 0.1$ and 0.01 under the uniform distribution

In the Dependent model, $|p_j^a - p_j^b| \leq \varepsilon_1$ for any channel $j \in [1, n]$ and ε_1 is an important parameter to choose. In Fig. 1, we show the performance of Alg. 5 (the attempting phase) under the uniform distribution (for user a as defined above) when ε_1 ranges from 0.01 to 0.1 for two scenarios $n = 1000$ and $n = 10000$ respectively. As depicted in the figure, the $ETTR$ values and $MTTR$ values for both situations are linearly increasing with ε_1, which corroborates our theoretical analysis in the time complexity to rendezvous. Additionally, the difference of TTR between $n = 1000$ and $n = 10000$ is not large since the value increases logarithmically with n.

We compare the performance of Alg. 5 with Alg. 2 if p_j^a $(1 \leq j \leq n)$ obeys the uniform distribution. Since different ε_1 values may result in different rendezvous time, we consider two situations for Alg. 5: $\varepsilon_1 = 0.1$ and $\varepsilon_1 = 0.01$. As shown in Fig. 2, Alg. 5 works much better than Alg. 2 from both $ETTR$ and $MTTR$ comparisons when n increases from 1000 to 10000. When ε_1 is smaller, the $ETTR$ and $MTTR$ values are also smaller. When $\varepsilon_1 = 0.01$, it only takes constant time to rendezvous and the $MTTR$ value is just 1% of Alg. 2.

Through the simulation results, our algorithms for both Independent model and Dependent model have good performance. For the Independent model, the algorithms for both synchronous and asynchronous settings perform well when the number of channels increases. For the Dependent model, the algorithm consisting of the sensing phase and the attempting phase outperforms the algorithms for the Independent model under the uniform distribution.

VII. Conclusion

In this paper, we propose the dynamic rendezvous problem in Cognitive Radio Networks (CRNs) where the status of the licensed channels varies over time. In order to describe the dynamic setting, we introduce the Independent model and the Dependent model; the probabilities that each channel is available are independently distributed in the Independent model, while two nearby users have dependent probabilities in the Dependent model. For the Independent model, we propose two randomized algorithms for two synchronous users and asynchronous users respectively, which guarantee rendezvous in $O(\log^2 n)$ and $O(\log^3 n)$ time slots with high probability. For the Dependent model, we present an efficient distributed algorithm consisting of a sensing phase and an attempting phase, which achieves rendezvous in $O(\varepsilon \log^3 n \log\log\log n)$ time slots with high probability. An algorithm to guarantee high rendezvous load is also proposed, which enables the users can communicate in $1/4$ of all time slots in a long run.

In the future, we aim to propose more realistic models to describe the dynamic setting and to design efficient algorithms to achieve rendezvous in certain harsh environment even with adversaries.

References

[1] FCC, ET Docket No 03-222 Notice of proposed rule making and order, December 2003.

[2] M. Ahmadi, Y. Zhuang, and J. Pan. Distributed Robust Channel Assignment for Multi-Radio Cognitive Radio Networks. In *Vehicular Technology Conference (VTC)*, 2012.

[3] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. NeXt Generation//Dynamic Spectrum Access//Cognitive Radio Wireless Networks: A Survey. *Computer Networks*, 50(13): 2127-2159, 2006.

[4] S. Chen, A. Russell, A. Samanta, and R. Sundaram. Deterministic Blind Rendezvous in Cognitive Radio networks. In *ICDCS*, 2014.

[5] I. Chuang, H.-Y. Wu, K.-R. Lee. and Y.-H. Kuo. Alternate Hop-and-Wait Channel Rendezvous Method for Cognitive Radio Networks. In *INFOCOM*, 2013.

[6] Z. Gu, Q.-S. Hua, Y. Wang, and F.C.M. Lau. Nearly Optimal Asynchronous Blind Rendezvous Algorithm for Cognitive Radio Networks. In *SECON*, 2013.

[7] Z. Gu, Q.-S. Hua, and W. Dai. Local Sequence Based Rendezvous Algorithms for Cognitive Radio Networks. In *SECON*, 2014.

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

[9] G. Li, Z. Gu, X. Lin, H. Pu, and Q.-S. Hua. Deterministic Distributed Rendezvous Algorithms for Multi-Radio Cognitive Radio Networks. In *MSWiM*, 2014.

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

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

[12] H. Pu, Z. Gu, X. Lin, Q.-S. Hua and H. Jin. Dynamic Rendezvous Algorithms For Cognitive Radio Networks. http://grid.hust.edu.cn/qshua/dynamic_rendezvous_full.pdf.

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

[14] N. Tadayon, and S. Aissa. Multi-Channel Cognitive Radio Networks: Modeling, Analysis and Synthesis. *IEEE Journal on Selected Area in Communications*, 2014.